CONFIGURATION GUIDE

Extreme NetIron Layer 2 Switching Configuration Guide, 6.3.00a

Supporting NetIron OS 6.3.00a

9035620-01
November 2018

Legal Notice

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

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

Trademarks

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

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

For additional information on Extreme Networks trademarks, please see: www.extremenetworks.com/company/legal/trademarks

Software Licensing

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

Support

For product support, phone the Global Technical Assistance Center (GTAC) at 1-800-998-2408 (toll-free in U.S. and Canada) or +1-408-579-2826. For the support phone number in other countries, visit: http://www.extremenetworks.com/support/contact/

For product documentation online, visit: https://www.extremenetworks.com/documentation/

Contents

Preface

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

Conventions

This section discusses the conventions used in this guide.

Notes, cautions, and warnings

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

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

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

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

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

Text formatting conventions

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

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

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

Command syntax conventions

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

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

Documentation and Training

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

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

Open Source Declarations

Some software files have been licensed under certain open source licenses. More information is available at: www.extremenetworks.com/support/policies/open-source-declaration/.

Training

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

Getting Help

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

Extreme Portal	Search the GTAC (Global Technical Assistance Center) knowledge base, manage support cases and service contracts, download software, and obtain product licensing, training, and certifications.
The Hub	A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
Call GTAC	For immediate support: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact

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

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

Subscribing to Service Notifications

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

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

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

4. Click **Submit**.

Providing Feedback to Us

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

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

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

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

- Email us at documentation@extremenetworks.com.

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

About This Document

Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

You should be familiar with the following protocols if applicable to your network - IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, MPLS, and VRRP.

Supported hardware and software

End of Support for ExtremeSwitching CES 2000 Series devices

Beginning with NetIron OS 6.3.00a and later, the ExtremeSwitching CES 2000 Series devices are not supported. Refer to the End of Sale and End of Support page for additional information.

The hardware platforms in the following table are supported by this release of this guide.

TABLE 1 Supported devices

ExtremeRouting XMR Series	ExtremeRouting MLX Series	ExtremeRouting CER 2000 Series
XMR 4000	MLX-4	CER 2024C
XMR 8000	MLX-8	CER-RT 2024C
XMR 16000	MLX-16	CER 2024F
XMR 32000	MLX-32	CER-RT 2024F
	MLXe-4	CER 2048C
	MLXe-8	CER-RT 2048C
	MLXe-16	CER 2048CX
	MLXe-32	CER-RT 2048CX
		CER 2048F
		CER-RT 2048F
		CER 2048FX
		CER-RT 2048FX

What's new in this document

NOTE
The NetIron 6.3.00 release (the image files and the documentation) is no longer available from the Extreme Portal. New software features introduced in release 6.3.00 are included in release 6.3.00a.

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

The following table includes descriptions of changes in functionality for the current release.

TABLE 2 Changes for the current release

Feature	Description	Described in
TVF LAG FID group size supports 32.	Supported values are now: 2, 4, 8, 16, or 32.	Configuring TVF FID group size on page 121

For the complete list of supported features and the summary of enhancements and configuration notes for this release, refer to the *Extreme NetIron OS Release Notes*.

Notice to the reader

This document may contain references to the trademarks of the following corporations. These trademarks are the properties of their respective companies and corporations.

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
Microsoft Corporation	Internet Explorer
Mozilla Corporation	Mozilla Firefox
Sun Microsystems	Java Runtime Environment

How command information is presented in this guide

Starting with Extreme NetIron 5.6.00, command syntax and parameter descriptions are removed from commands that are referenced in configuration tasks. To find the full description of a specific command, including all required and optional keywords and variables, refer to the *Extreme NetIron Command Reference* for your software release.

XMR Series and MLX Series Link Aggregation

XMR Series and MLX Series Link Aggregation

NOTE
This chapter is applicable only to the XMR Series and MLX Series devices.

This chapter describes how to configure Link Aggregation Groups (LAG) for the XMR Series and MLX Series. You can use a single interface to configure any of the following LAG types:

- **Static LAGs** - These LAG groups are manually-configured aggregate links containing multiple ports.
- **Dynamic LAGs** - This LAG type uses the Link Aggregation Control Protocol (LACP), to maintain aggregate links over multiple port. LACP PDUs are exchanged between ports on each device to determine if the connection is still active. The LAG then shuts down ports whose connection is no longer active.
- **Keep Alive LAGs** - In a Keep Alive LAG a single connection between a single port on 2 Extreme devices is established. In a keep alive LAG, LACP PDUs are exchanged between the 2 ports to determine if the connection between the devices is still active. If it is determined that the connection is no longer active, the ports are blocked.

NOTE
The new LAG configuration procedures supersede the previous configurations procedures for LAGs and Dynamic Link Aggregation.

LAG formation rules

The LAG formation rules are mentioned below:

- The 10Gx24-DM module ports can only be part of LAGs exclusively consisting of 24x10G ports. A LAG cannot have a mix of 24x10G module ports and any other 10G module ports.
- A port can only be a member of one LAG, and that LAG must be static, dynamic or a keep-alive LAG.
- The maximum number of port members that may be assigned to a LAG is dependent on the number of trunks specified by the *system-max trunk-num* value.
- The system supports up to 64 port IDs for CES 2000 Series and MLX Series devices when **snmp-server max-ifindex-per-module 64** is configured.
- All ports configured in a LAG must be of equal bandwidth. For example all 10 G ports.
- All ports configured in a LAG must be configured with the same port attributes.
- LAG formation rules are checked when a static or dynamic LAG is deployed.

- A LAG must have its primary port selected before it can be deployed.
- All ports configured in a LAG must be configured in the same VLAN.
- All ports must have the same PBR configuration before deployment. During deployment, the configuration on the primary port is replicated to all ports. On undeployment, each port inherits the same PBR configuration.
- All static LAG ports must have the same LACP BPDU forwarding configuration.
- A LAG member and an individual port cannot use the same name.
- VLAN and inner-VLAN translation

The LAG is rejected if any LAG port has VLAN or inner-VLAN translation configured

- Layer 2 requirements:

The LAG is rejected if the LAG ports:

- – Do not have the same untagged VLAN component.
  – Do not share the same SuperSpan customer ID (CID).
  – Do not share the same VLAN membership or do not share the same uplink VLAN membership
  – Do not share the same protocol-VLAN configuration
  – Are configured as mainly primary and secondary interfaces
  – Static LAG deployment will fail if the if LACP BPDU forwarding is disabled on the primary port and enabled on one or more of the secondary ports.
- Layer 3 requirements:

The LAG is rejected if any of the secondary LAG port has any Layer 3 configurations, such as IPv4 or IPv6 address, OSPF, RIP, RIPNG, IS-IS, and so on.

- Layer 4 (ACL) requirements:
  - All LAG ports must have the same ACL configurations; otherwise, the LAG is rejected.
  - A LAG cannot be deployed if any of the member ports has ACL-based mirroring configured on it.
  - A port with ACL-based mirroring configured on it cannot be added to a LAG.
- The router can support up to 256 LAGs, and each LAG can contain up to 64 member ports.
  - If the router is configured to support 32 LAGs by using the **system-max trunk-num** command, the maximum number of LAG ports is 64.
  - If the router is configured to support 64 LAGs by using the **system-max trunk-num** command, the maximum number of LAG ports is 32.
  - If the *system-max trunk-num* value is set to 256, the maximum number of LAG ports supported is 8.
  - The default system-max trunk-num value is set to 128, and each LAG can have up to 16 member ports
  - For 40G and 100G ports, the number of LAG FIDs is 128. The configurable ranges are from 2 to 64 LAGs.
- When configuring a static or dynamic LAG, if trunk load sharing type is set to "per-packet" the maximum number of "per-packet" trunks is set to 4.
- Ports can be in only one LAG group. All the ports in a LAG group must be connected to the same device at the other end. For example, if port 1/4 and 1/5 in Device 1 are in the same LAG group, both ports must be connected to ports in Device 2 or in Device 3. You cannot have one port connected to Device 2 and another port connected to Device 3.
- All LAG member properties must match the primary port of the LAG with respect to the following parameters:
  - Port tag type (untagged or tagged port)
  - Port speed and duplex
  - TOS-based Configuration - All ports in the LAG must have the same TOS-based QoS configuration before LAG deployment, During deployment the configuration on the primary port is replicated to all ports and on undeployment, each port inherits the same TOS-based QoS configuration.

To change port parameters, you must change them on the primary port. The software automatically applies the changes to the other ports in the LAG.

- Using the **system-max trunk-num** command, the device can support the following LAG/member port configurations:
  - 256 LAGs with each containing 8 member ports.
  - 128 LAGs with each containing 16 member ports.
  - 64 LAGs with each containing 32 member ports.
  - 32 LAGs with each containing 64 member ports.
- Using the **system-max trunk-num-100g** command, the device can support the following 40 GbE and 100 GbE LAG scalability configurations.
  - 64 LAGs with each containing 2 member ports.
  - 32 LAGs with each containing 4 member ports.
  - 16 LAGs with each containing 8 member ports.
  - 8 LAGs with each containing 16 member ports.
  - 4 LAGs with each containing 32 member ports.
  - 2 LAGs with each containing 64 member ports.
- The total number of ports in a trunk is controlled by the **system-max trunk-num** command for both non-100G and 100G trunks.
- Make sure the device on the other end of the LAG link can support the same number of ports in the link.

Mixed port LAG support for 2x100GbE module:

LAGs can be formed between ports that have the same speed. For example, the default speed for 10G or 1G port is auto for 4x10GbE card, similar to 2x100GbE card. LAG is supported between 10G port and 1G port, if the port speed is in AUTO configuration.

- If a 2x100GbE port is added to a LAG, then the port should be configured to operate in AUTO mode.
- The 2x100GbE port supports LAG formation with any other 10G or 1G port.
- If 2x100GbE port is speed configured and is added to LAG, then the LAG deployment fails with error message.
- If a 2x100GbE port is added to a LAG and if the ports of a LAG are of different operating speed, packet loss is expected.
- For 2x100GbE ports “confirm-port-up” value check is not performed at LAG deployment.

TABLE 3 The LAG/member configuration

Maximum number of 1/10G LAGs	Maximum number of 1/10G LAG ports	Maximum number of 100/40G LAG ports	Maximum number of 100/40G LAGs possible	Maximum number of LAGs in the system
256	8	2 to 8	16 for 8-ports 64 for 2-ports	256
128	16	2 to 16	8 for 16-ports 64 for 2-ports	128
64	32	2 to 32	4 for 32-ports 64 for 2-ports	64
32	64	4 to 64	2 for 64-ports 32 for 4-ports	32

Figure 1 displays an example of a valid, Keep ALIVE LAG link between two devices. This configuration does not aggregate ports but uses the LACP PDUs to maintain the connection status between the two ports.

FIGURE 1 Example of a 1-port keep alive LAG

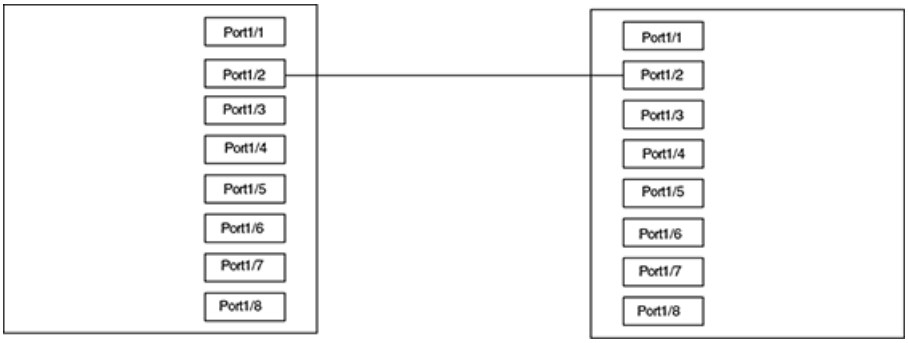


Figure 2 shows an example of a valid 2-port LAG link between devices where the ports on each end are on the same interface module. Ports in a valid 2-port LAG on one device are connected to two ports in a valid 2-port LAG on another device.

FIGURE 2 Example of 2-port LAG

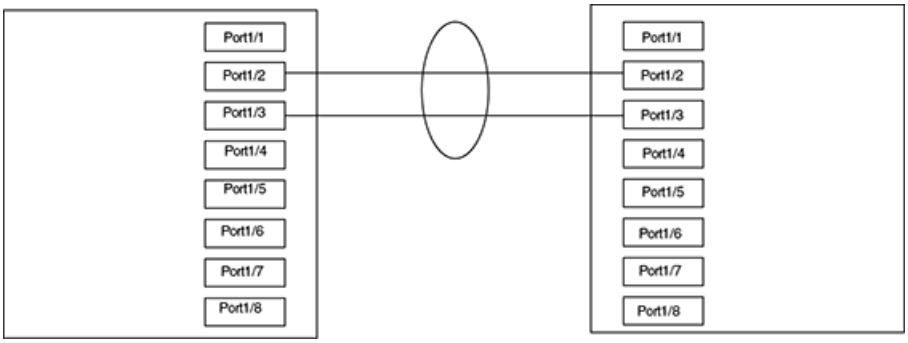
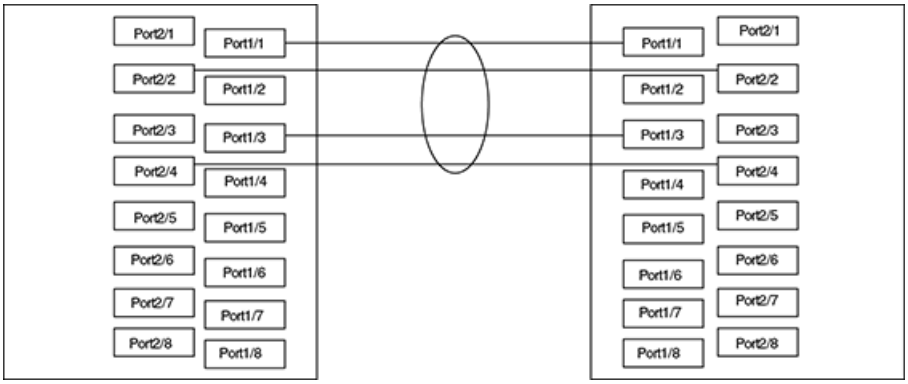


Figure 3 shows an example of two devices connected over a 4 port LAG where the ports on each end of the LAG are on different interface modules.

FIGURE 3 Examples of multi-slot, multi-port LAG



LAG load sharing

Extreme devices can be configured for load sharing over a LAG by using the following:

- Hash-based load sharing
- Per-packet load sharing

Each of these methods, that are described in the following sections, are configured per LAG using the trunk-type command as described in the Configuring load sharing type section.

Hash-based load sharing

The Extreme device shares the traffic load evenly across the ports in LAG group, while ensuring that packets in the flow are not reordered. Individual flows are assigned a LAG index to identify them. An improved hash based load sharing algorithm has the following enhancements:

- Better distribution
- Support for 32-port LAGs
- An increased number of fields in the packet header that can be used for load balancing
- Enhanced load sharing in configurations of ECMP with LAGs.

Traffic from each flow is then distributed across the ports in the LAG group using a hash index as follows:

- For Layer 2 switching, the hash index is based on the following:
  - IPv4 or IPv6 traffic: source MAC address and destination MAC address, IPv4v6 source and destination address, VLAN-ID, IPv4 protocol number or IPv6 next header and TCP or UDP source port and TCP or UDP destination port.

    NOTE
    TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or the **load-balance force-l4-hashing** command is configured.

- Layer-2 packets with an MPLS payload: source MAC address and destination MAC address, VLAN ID, Inner VLAN ID (for double-tagged packets), Ethertype, and up to 3 MPLS Labels.

  NOTE
  For double-tagged packets, Ethertype is not used and the TPID of the inner TAG must be 0x8100 to be considered a double-tagged packet.

- GRE encapsulated IPv4 traffic: source MAC address and destination MAC address, IPv4 source and destination address, IPv4 protocol number. VLAN-ID, and TCP or UDP source port and TCP or UDP destination port. Also, inner IPv4 source address and inner destination IPv4 address are used if there is at least one GRE or IPv6 tunnel configured.

  NOTE
  TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured.

- IPv6 tunnel encapsulated IPv6 traffic: source MAC address and destination MAC address, IPv6 source and destination address, TCP or UDP source port and TCP or UDP destination port, IPv6 next header, and VLAN ID.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:- Packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured. For 6over4 encapsulated packets, inner IPv6 destination address and inner IPv6 source address are used if there is at least one GRE or IPv6 tunnel configured.

- Layer-2, non-IPv4, IPv6 or non-MPLS packets: source MAC address, destination MAC address, VLAN ID, and Ether type.
- For Layer 3-Routing, the hash index is based on the following:
  - IPv4 or IPv6 traffic: source MAC address and destination MAC address, IPv4v6 source and destination address, VLAN-ID, IPv4 protocol number or IPv6 next header and TCP or UDP source port and TCP or UDP destination port.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured.

- GRE encapsulated IPv4 traffic: source MAC address and destination MAC address, IPv4 source and destination address, IPv4 protocol number. VLAN-ID, and TCP or UDP source port and TCP or UDP destination port. Also, inner IPv4 source address and inner destination IPv4 address are used if there is at least one GRE or IPv6 tunnel configured.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured.

- IPv6 tunnel encapsulated IPv6 traffic: source MAC address and destination MAC address, IPv6 source and destination address, TCP or UDP source port and TCP or UDP destination port, and IPv6 next header. and VLAN-ID.

NOTE
TCP or UDP Destination and Source port is used under the following conditions:- Packet is a TCP or UDP packet, or, the **load-balance force-l4-hashing** command is configured. For 6over4 encapsulated packets, inner IPv6 destination address and inner IPv6 source address are used if there is at least one GRE or IPv6 tunnel configured.

- For MPLS switching, the hash index is based on the following:
  - L2VPN traffic: outer source MAC address and outer destination MAC address, up to two MPLS Labels, VLAN ID, inner source MAC address and inner destination MAC address, If packet payload is an IPv4 or v6 packet: IPv4v6 source and destination address, IPv4 Protocol Number or IPv6 Next Header ID of the payload are used.
  - L3VPN traffic or IP shortcut traffic: outer source MAC address and outer destination MAC address, VLAN ID, inner source IPv4v6 address and inner destination IPv4v6 address, IPv4 Protocol Number or IPv6 Next Header ID, TCP source port and TCP destination port, UDP source port and UDP destination port, and up to two MPLS Labels.
  - MPLS packets with 3 labels: outer source MAC address and outer destination MAC address, VLAN ID, and all 3 MPLS Labels.

NOTE
For transit LSRs please note the following: The **load-balance speculate-mpls-ip** command must be active. It is on by default. If the **load-balance speculate-mpls-ip** command has been configured to be inactive, and the **load-balance speculate-mpls-enet** command is active, the packet will be processed like an L2VPN packet. If both commands are configured to be inactive, no inner layer 2 or layer 3 headers are considered but up to 3 MPLS labels are used for hashing.

Options for hash based load sharing

The following options can be used to refine the hash calculations used for LAGs:

- Speculate UDP or TCP Headers

- Mask Layer-4 Source and Destination Port Information
- Hash Diversification

Each of these options when configured apply to both IP Load Sharing and LAG Load sharing. They are described in detail in Configuring IP Chapter.

Load sharing for MPLS LAGs

Load sharing on MPLS LAG involves traffic flows that include the MPLS Inner and Outer Labels. These can be used exclusively or in combination with the IP and MAC source and destination addresses to determine the LAG index for a traffic flow.

Using IP source and destination addresses for load sharing

You can use the **load-balance speculate-mpls-ip** command to include the IP source and destination addresses in the calculation of the LAG index for a traffic flow within MPLS LAGs, as shown in the following.

```
device(config)# load-balance speculate-mpls-ip all
```

Syntax: [no] load-balance speculate-mpls-ip [**all** | **slot-number** | **slot-number np-id**]

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
The **load-balance speculate-mpls-ip** command will hash only on the IP portion.

Using MAC source and destination addresses for load sharing

You can use the **load-balance speculate-mpls-enet** command to include the MAC source and destination addresses in the calculation of the LAG index for a traffic flow within MPLS LAGs, as shown in the following.

```
device(config)# load-balance speculate-mpls-enet all
```

Syntax: [no] load-balance speculate-mpls-enet [**all** | **slot-number** | **slot-number np-id**]

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
The **load-balance speculate-mpls--enet** command will hash only on the Ethernet header portion.

Per-packet server LAG load sharing

Per-packet LAG load balancing is a type of LAG that load balances traffic on a per-packet basis, as compared to traditional server LAG load-balancing which balances traffic based on packet content such as source or destination addresses. In per packet server LAG load balancing, the packet processor (PPCR) on each module selects a port in the per packet server LAG to forward traffic in a round-robin fashion. For example, if the first port of the per packet server LAG is currently selected, the second port of the per-packet server LAG will be used next, and so on. Consequently, traffic is evenly distributed among all of the ports that are configured in a per packet server LAG.

Traffic that can be forwarded out of a per-packet LAG includes Layer 2 switching traffic, Layer 3 routing traffic, L3VPN (2547) traffic, VLL and VPLS traffic.

Configuring a LAG

The following configuration procedures are used to configure a LAG. Depending upon whether you are configuring a static, dynamic or keep-alive LAG, the configuration procedures may or may not apply as described:

- **Creating a Link Aggregation Group** - Required for all static, dynamic or keep alive LAGs.
- **Adding Ports to a LAG** - Required for all static, dynamic, or keep alive LAGs. A keep alive LAG contains only one port while static and dynamic LAGs can have 2 to 32 ports.
- **Configuring the Primary Port for a LAG** - Required for all static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Configuring the Load Sharing Type** - Optional for all static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Specifying the LAG Threshold for a LAG Group** - Optional for static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Configuring LACP Port Priority** - Optional for dynamic and keep alive LAGs.
- **Configuring an LACP Timeout** - Optional for dynamic and keep alive LAGs.
- **Configuring LACP BPDU Forwarding** - Optional for static LAGS only since LACP BDUs are discarded (dropped) on ports in which a static LAG has been configured as the default setting.

Creating a LAG using the LAG ID option

Before setting-up ports or configuring any other aspects of a Link Aggregation Group (LAG), you must create it first.

You can either assign a LAG ID explicitly or it will be automatically generated by the system. The LAG ID stays the same across system reload and hitless upgrade.

The command to configure LAGs allows explicit configuration of the LAG ID for static and dynamic LAGs.

To create a LAG with the LAG ID option, enter a command such as the following.

```
device(config)# lag blue static
device(config-lag-blue)#
```

Syntax: [no] lag name [**static** | **dynamic**] [**id number**]

The **ID** parameter is optional. The value of the **ID** parameter that you can enter is from 1 to 256. If you do not enter a LAG **ID** , the system will generate one automatically. Once the LAG **ID** is generated the system will save it in the configuration file along with the LAG name, therefore the value will stay the same across system reload.

NOTE
The LAG **ID** parameter is for static and dynamic LAGs only. No explicit configuration of a LAG **ID** is allowed on keepalive LAGs.

The **static** parameter specifies that the LAG with the name specified by the *lag-name* variable will be configured as a static LAG.

The **dynamic** option specifies that the LAG with the name specified by the *lag-name* variable will be configured as a dynamic LAG.

Configuration considerations

LAG IDs are unique for each LAG in the system. The same LAG IDs cannot be assigned to two or more different LAGs. If a LAG ID is already used, the CLI will reject the new LAG configuration and display an error message that suggests the next available LAG ID that can be used.

```
device(config)#lag lag3 static id 123
Error: LAG id 123 is already used. The next available LAG id is 2
.
```

LAG configured with LAG ID 124.

```
!
lag "lag1" static id 124
 ports ethernet 1/2 to 1/3
 primary-port 1/3
 deploy
!
```

The **show lag** command and the output.

```
device(config)# show lag
Total number of LAGs:          1
Total number of deployed LAGs: 1
Total number of s created:1 (127 available)
LACP System Priority / ID:     1 / 0000.0001.c000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "lag1" ID 124 (static Deployed) ===
LAG Configuration:
   Ports:         ethe 1/2 to 1/3
   Port Count:    2
   Primary Port:  1/3
    Type:    hash-based
Deployment:   ID 124, Active Primary 1/2
Port  Link L2 State  Dupl Speed  Tag Priori MAC Name
1/2   Up   Forward   Full 10G   124   No  level0 0000.0001.c002
1/3   Up   Forward   Full 10G   124   No  level0 0000.0001.c002
```

Creating a keepalive LAG

To create a **keep-alive** LAG, enter the following.

```
device(config)# lag lag1 keep-alive
```

Syntax: [no] lag name keep-alive

The **keep-alive** option specifies that the LAG with the name specified by the *lag-name* variable will be configured a keep-alive LAG. The keep-alive LAG option allows you to configure a LAG for use in keep alive applications similar to the UDLD feature.

Disabling the Detection of Remote LACP Configuration Removal

The **lacp-cfg-det-dis** command is a global command and is used to disable detecting remote end LACP configuration removal. By default, this feature this is enabled. To disable this feature, enter a command such as the following:

```
device(config)# lacp-cfg-det-dis
```

Syntax: lacp-cfg-det-dis

> NOTE
> When you have interoperability with another vendor's device, you will need to disable this feature as other vendors devices will remove the fiber.

Modifying an existing LAG name

The device supports changing an existing LAG name without deleting and recreating the LAG.

Use the **update-lag-name** command to modify an existing LAG name. This command works for all LAG types, such as static, dynamic, and keepalive LAGs.

```
device(config)# lag blue
device(config-lag-blue)# update-lag-name extreme
```

Syntax: update-lag-name *new-name*

> NOTE
> The modified LAG name should be unique across all the LAG names that are available.

Adding Ports to a LAG or Deleting Ports from a LAG

A static or dynamic LAG can consist of from 2 to 32 ports of the same type and speed that are on any interface module within the Extreme chassis. A keep alive LAG consists of only one port.

To configure the static LAG named "blue" with two ports, use the following command:

```
device(config)# lag blue static
device(config-lag-blue)# ports ethernet 3/1 ethernet 7/2
```

Syntax: [no] ports ethernet *slot/port* [**to** *slot/port*] [**ethernet** *slot/port*]

The ports added to a LAG can be of type **ethernet** as specified for the **slot/port** where they reside. The ports can be added to the LAG sequentially as shown in the following example:

```
device(config-lag-blue)# ports ethernet 3/1 ethernet 7/2 ethernet 4/3 ethernet 3/4
```

A range of ports from a single interface module can be specified. In the following example, Ethernet ports 1, 2, 3 and 4 on the interface module in slot 3 are configured in a single LAG:

```
device(config-lag-blue)# ports ethernet 3/1 to 3/4
```

Additionally, you can mix a range of ports from one interface module with individual ports from other interface modules to form a LAG as shown in the following:

```
device(config-lag-blue)# ports ethernet 3/1 to 3/4 ethernet 10/2
```

Using the **no** option allows you to remove ports from a LAG. For example, you can remove port 3/4 from the LAG created above, as shown in the following:

```
device(config-lag-blue)# no ports ethernet 3/4
```

Ports can be added to an undeployed LAG or to currently deployed LAG using the commands described. For special considerations when adding ports to or deleting ports from a currently deployed LAG, refer to the following sections:

Adding a port to a currently deployed LAG on page 39

Deleting a port from a currently deployed LAG on page 39

NOTE
The command supports dynamic selection of primary port. The next primary port that is selected is the next available least port ID.

Configuring the primary port for a LAG

The primary port must be explicitly assigned using the **primary-port** command.

To designate the primary port for the static LAG "blue", use the following command.

```
device(config)# lag blue static
device(config-lag-blue)# primary-port 3/2
```

Syntax: [no] primary-port slot/port

This configuration is applicable for configuration of a static or dynamic LAG in MLX Series devices only. Once a primary port has been configured for a LAG, all configurations that apply to the primary port are applied to the other ports in the LAG.

NOTE
The command supports dynamic selection of primary port. The next primary port that is selected is the next available least port ID. For more information related to dynamic selection, refer to Dynamic primary port selection on page 40 topic in the guide.

Configuring load sharing type

Individual LAGs can be configured to perform load sharing over the ports in the LAG using either a hash based or per packet method, as shown in the following .

```
device(config)# lag blue static
device(config-lag-blue)# trunk-type hash-based
```

Syntax: [no] trunk-type hash-based | **per-packet**

NOTE
This configuration is only applicable for configuration of a static or dynamic LAGs.

Specifying the LAG threshold

Trunk threshold brings the LAG down if the trunk threshold condition is not met by the LAG.

Syntax: [no] trunk-threshold *number*

You can specify a threshold from 1 (the default) up to the number of ports in the LAG.

NOTE
This configuration is only applicable for configuration of a static or dynamic LAG.

Below are the different behavioral patterns for static and dynamic LAG.

Static LAG behavior

You can configure the Extreme device to disable all of the ports in a LAG when the number of active member ports drops below a specified threshold value. For example, if a LAG has 8 ports, and the threshold for the LAG is 5, then the LAG is disabled if the number of available ports in the LAG drops below 5. If the LAG is disabled, then traffic is forwarded over a different link or LAG.

For example, the following commands establish a LAG consisting of four ports, and then establish a threshold of three ports for this LAG.

```
device(config)# lag blue static
device(config-lag-blue)# ports ethernet 3/1 to 3/4
device(config-lag-blue)# trunk-threshold 3
```

In this example, if the number of active ports drops below three, then all the ports are disabled in the LAG.

When a LAG is down because of not meeting the LAG condition, the LAG is kept intact and it is re-enabled if enough ports become active to reach the threshold.

NOTE
The **trunk-threshold** command should be configured only at one end of the trunk. If it is set on both sides, link failures result in race conditions and change in functionality.

Dynamic LAG behavior

Unlike in static LAGs, if the number of active ports in a dynamic LAG falls below the threshold value, the ports are logically blocked and out-of-sync is signaled on all the active member ports in that LAG until it satisfies the trunk threshold condition again.

For example, the following commands establish a LAG consisting of four ports, and then sets a threshold of three ports.

```
device(config)# lag red dynamic
device(config-lag-blue)# ports ethernet 3/1 to 3/4
device(config-lag-blue)# trunk-threshold 3
```

In this example, if the number of active ports drops below 3, then all the ports in the LAG are logically blocked. An out-of-sync is signaled on all the active member ports in that LAG until it satisfies the trunk threshold condition again.

NOTE
Configure the same LACP Trunk threshold value on both sides of the LAG for better performance or for connecting to the third-party devices.

NOTE
Configuring the LACP trunk threshold on a third-party device may occasionally cause performance or incorrect traffic flow issues. To avoid this issue, configure the trunk threshold on either the MLX Series device only or on both the MLX Series device and the third-party device.

Configuring an LACP port priority

In a dynamic or keep-alive LAG, a port priority can be configured at the global level.

```
device(config)# lag blue dynamic
device(config-lag-blue)# lacp-port-priority 100000
```

Syntax: [no] lacp-port-priority slot/port *number*

> NOTE
> This configuration is only applicable for configuration of a dynamic or keep-alive LAGs.

Configuring an LACP system priority

In a dynamic or keep-alive LAG, a system priority can be configured at global level.

```
device(config)# lacp system-priority 4
```

Syntax: [no] lacp system-priority *number*

The number value specifies the value of the LACP system priority. This can be a value from 1 to 65535. The default system-priority value is 1.

> NOTE
> This configuration is only applicable for configuration of a dynamic or keep-alive LAGs.

> NOTE
> In a system configuration with multiple MCT peers, the LACP system priority on both the MCT nodes should be the same.

Configuring an LACP timeout

In a dynamic or keep-alive LAG, a port's timeout can be configured as short (3 seconds) or long (90 seconds). After you configure a port timeout, the port remains in that timeout mode whether it is up or down and whether or not it is part of a LAG.

> NOTE
> For CES 2000 Series and CER 2000 Series devices, Extreme recommends that you use the default setting of the long timeout mode for the LACP timers.

All the ports in a LAG should have the same timeout mode. This requirement is checked when the LAG is enabled on the ports. For example, to configure a port for a short LACP timeout, use the following command.

```
device(config)# lag blue dynamic
device(config-lag-blue)# lacp-timeout short
```

Syntax: [no] lacp-timeout [**long** | **short**]

To delete the configuration, use the **no** form of this command.

The **long** keyword configures the port for the long timeout mode-90 seconds. With the long timeout, an LACPDU is sent every 30 seconds. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transition to the appropriate state based on its current state.

The **short** keyword configures the port for the short timeout mode--3 seconds. In the short timeout configuration, an LACPDU is sent every second. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transitions to the appropriate state based on its current state.

If you specify neither **long** nor **short**, the state machine operates based on the standard IEEE specification as its default behavior. The original IEEE specification says that the state machine starts with short the timeout and moves to the long timeout after the LAG is established. However, sometimes a vendor's implementation always uses either the short timeout or the long timeout without changing the timeout. Extreme provides this command so that you can configure Extreme devices to interoperate with other vendor's devices.

NOTE
This configuration is applicable to the configuration of dynamic or keep-alive LAGs only.

Configuring LACP BPDU Forwarding

Enabling and Disabling LACP BPDU Forwarding on a Port

For scenarios in which static LAG ports require LACP BPDU packet forwarding, you can issue the **forward-lacp** command in the interface configuration mode. Once LACP Forwarding has been enabled on a static LAG, all the LACP BPDUs will follow regular packet forwarding actions.

When LACP forwarding is enabled, the link OAM packets received on the LACP forwarding enabled interface will be processed and flooded on the VLAN. If the LACP forwarding is not enabled, the link OAM packets will be processed and then dropped.

To enable LACP BPDU forwarding, enter the LACP Forwarding command as follows.

```
device(config-if-e1000-3/5)# forward-lacp
```

Syntax: forward-lacp

To disable LACP BPDU forwarding, enter the lacp-forwarding command as follows. When a static LAG is undeployed the LACP BPDU forwarding state of the LAG will be retained on the individual ports.

```
device(config-if-e1000-3/5)# [no]
forward-lacp
```

Syntax: [no] forward-lacp

The **forward-lacp** option specifies that the port of the specified static LAG will be configured for LACP-BPDU forwarding. If the specified port is a dynamic or keep alive LAG, an error message will be displayed.

Enabling and Disabling LACP BPDU Forwarding on a LAG

NOTE
The forward-lacp command must be issued on the physical port configuration, not in LAG configuration.

When the LACP forwarding is enabled on the primary port of the static LAG, the LACP BPDU forwarding is enabled on all ports of the LAG when the LAG is deployed. When the static LAG is undeployed the BPDU forwarding state is retained.

NOTE
LACP BPDU forwarding is not supported for any port of dynamic or keep alive LAGs.

- If LACP BPDU forwarding is enabled on the primary and secondary ports, the static LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is enabled on the primary port and disabled on the secondary ports, the static LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is disabled on the primary and secondary ports, the static LAG deployment will be successful and LACP BPDU forwarding will be disabled on the LAG ports.

Deploying a LAG

After configuring a LAG, you must explicitly enable it before it begins aggregating traffic. This task is accomplished by executing the **deploy** command within the LAG configuration. After the **deploy** command runs, the LAG is in the aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG. The running configuration will no longer display deployed LAG ports other than the primary port.

To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keep-alive LAGs. After a non keep-alive LAG is deployed, a LAG is formed. If there is only one port in the LAG, a single port LAG is formed. For a dynamic LAG, LACP is started for each LAG port. For a keep-alive LAG, no LAG is formed and LACP is started on the LAG port.

You can deploy a LAG as shown in the following for the "blue" LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
```

Syntax: [no] deploy [**forced** | **passive**]

When the **deploy** command is executed:

For a static and dynamic LAGs, the current LAG veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a LAG is formed with all the ports in the LAG.

For dynamic LAGs, by default all LAG ports will be on the **active** mode and LACP is activated on all LAG ports. If you specify **passive** mode, the LACP ports do not initiate the aggregation aggressively and ports will respond to LACP packets only when it receives LACP PDUs.

For a keep-alive LAGs, no LAG is formed, and LACP is started on the LAG port.

Once the **deploy** command is issued, all LAG ports will behave like a single port.

If the **no deploy** command is executed, the LAG is removed. For dynamic LAGs, LACP is de-activated on all of the LAG ports.

If the **no deploy** command is issued and more than 1 LAG port is not disabled the command is aborted and the following error message is displayed: "Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted." Using the **forced** keyword with the **no deploy** command in the previous situation, the un-deployment of the LAG is executed.

Operations available under LAG once it is deployed

Once a LAG has been deployed, the following configurations can be performed on the deployed LAG:

- Configuring ACL-based Mirroring
- Disabling Ports within a LAG
- Enabling Ports within a LAG

- Monitoring and Individual LAG Port
- Dynamic primary port selection
- Assigning a name to a port within a LAG
- Enabling sFlow Forwarding on a port within a LAG
- Setting the sFlow Sampling Rate for a port within a LAG
- Configuring LACP BPDU Forwarding

Configuring ACL-based mirroring

To configure ACL-based mirroring for all ports in a LAG, configure it on the primary port of the LAG at the interface configuration level (see Configuring IP Chapter). ACL-based mirroring can be configured for an individual member port within a LAG by using the **acl-mirror-port** command, as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# acl-mirror-port ethe-port-monitored 3/1 ethernet 3/2
```

In this example, traffic on Ethernet port 3/1 (a member port of LAG "blue") will be mirrored to Ethernet port 3/2.

Syntax: [no] acl-mirror-port { **ethe-port-monitored slot/port** | **named-port-monitored name** } **ethernet slot/port**

Use the **ethe-port-monitored** option with the appropriate *slot/port* variable to specify an Ethernet port for which you want to provide ACL mirroring.

Use the **named-port-monitored** option with the appropriate *slot/port* variable to specify a named port for which you want to provide ACL mirroring.

The **ethernet** keyword precedes the *slot/port* variable identifying the port which will receive the mirrored packets.

> NOTE
> A port with ACL-based mirroring already configured on it cannot be added to a LAG, and a LAG cannot be deployed if any of its member ports has ACL-based mirroring. To use ACL-based mirroring on a LAG member port, deploy the LAG, then configure mirroring on the member port. If a port is removed from a LAG, ACL-based mirroring will be removed from that port, and if a LAG is deleted mirroring will be removed from all member ports.

Disabling ports within a LAG

You can disable an individual port within a LAG using the disable command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# disable ethernet 3/1
```

Syntax: [no] disable ethernet [**slot/port**] | **named** [**name**]

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to disable.

Use the **named** option with the appropriate *[slot/port]* variable to specify a named port within the LAG that you want to disable.

Enabling ports within a LAG

You can enable an individual port within a LAG using the enable command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# enable ethernet 3/1
```

Syntax: [no] enable ethernet [**slot/port**] | **named** [**name**]

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to enable.

Use the **named** option with the appropriate *[slot/port]* variable to specify a named port within the LAG that you want to enable.

Adding a port to a currently deployed LAG

Ports can be added to a currently deployed LAG. Adding a port to a deployed LAG uses the same procedures as described in Adding Ports to a LAG or Deleting Ports from a LAG on page 32. When you add ports to a deployed LAG, the MAC address of the port being added is changed to that of the primary port of the LAG to which it is being added.

When adding a port to a currently deployed static LAG the LACP BPDU forwarding configuration must be the same as the LAG. Follow the procedure on Enabling and Disabling LACP BPDU Forwarding on a Port on page 36.

Deleting a port from a currently deployed LAG

Ports can be deleted from a currently deployed LAG. Deleting a port in a currently deployed LAG uses the same procedures as described in Adding Ports to a LAG or Deleting Ports from a LAG on page 32. However, when deleting ports from a currently deployed LAG you must consider the following:

- The primary port cannot be removed.
- If removal of a port will result in the trunk threshold value becoming greater than the number of ports in the LAG, the port deletion will be rejected.
- The port being deleted must be in the "disabled" state or you must use the forced option (as described in the following command syntax) when deleting it from a currently deployed LAG. Otherwise, the deletion request will be denied and the following error message will be displayed: "Error: ports to be deleted from the deployed LAG are not disabled, deleting these ports from the LAG may form a loop - aborted."
- When a port is deleted from a deployed static LAG, the LACP BDPU forwarding state of the LAG will be retained on the deleted port.

To delete port 3/1 which is in the "enabled" state from a currently deployed LAG named "blue", use the following command:

```
device(config)# lag blue static
device(config-lag-blue)# no ports ethernet 3/1 forced
```

Syntax: no ports ethernet slot/port [**to slot/port**] [**ethernet slot/port**] [**forced**]

This command operates as described in Adding Ports to a LAG or Deleting Ports from a LAG on page 32 except for the **forced** option which is described in the following:

The **forced** option to the **no ports** command deletes a port from a currently deployed LAG even if it is currently in the "enabled state". Because deleting an enabled port from a currently deployed LAG can cause a loop to be formed, we recommend that you disable any port being removed from a LAG before removing it. Only use the **forced** option when you are confident that a loop will not be created in your network topology.

NOTE
When a port is deleted from a currently deployed LAG, the MAC address of the port is changed back to its original value.

Monitoring an individual LAG port

By default, when you monitor the primary port in a LAG group, aggregated traffic for all the ports in the LAG is copied to the mirror port. You can configure the device to monitor individual ports in a LAG including Ethernet, or named ports. You can monitor the primary port or another member port individually.

NOTE
You can use only one mirror port for each monitored LAG port. To monitor traffic on an individual port in a LAG group, enter commands such as the following.

This command enables monitoring of an individual port within a LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# monitor ethe-port-monitored 3/1 ethernet 10/3 both
```

Syntax: [no] monitor ethe-port-monitored [**slot/port**] | **named-port-monitored** [**name**] | **ethernet** [**slot/port**] [**input** | **output** | **both**]

Use the **ethe-port-monitored** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to monitor.

Use the **named-port-monitored** option with the appropriate *[slot/port]* variable to specify a named port within the LAG that you want monitor.

The **ethernet** *slot/port* parameter specifies the port to which the traffic analyzer is attached.

The **input, output, and both** parameters specify the traffic direction to be monitored.

Dynamic primary port selection

Dynamic primary port selection enables changing the current primary port of a deployed LAG without un-deploying the LAG and without causing traffic disruption in the network. Use the **lag port-primary-dynamic** command to enable dynamic primary port selection at the global configuration level.

Dynamic primary port selection occurs when:

- The primary port is not specified during LAG creation using the **deploy** command.
- The user tries to delete the primary port when the LAG is already in the deployed state.

Dynamic primary port selection is designed to support instances where the LAG acts as a member of virtual interface VE and run protocols on top of the VE. This is applicable for Layer 3 protocol configuration running on VE where LAG is member of VE. The dynamic primary port selection also enables the user to configure a primary port dynamically of their choice.

NOTE
The user should migrate the physical interface configurations to VE to enable dynamic primary port selection.

The following system log message notifies the change in the primary port.

```
"LAG: Primary port for trunk-id <trunk_id> has been changed from port <old_primary> to
<new_primary>"
```

You can use the **show running-configuration** command to verify that the primary port has been changed on the LAG.

The following example verifies that the primary port has changed from 1/1 to 3/3.

```
Verify running configuration before changing the primary port
=============================================================
device(config-lag-test)#show running-configuration lag test
!
lag port-primary-dynamic
lag "test" dynamic id 10
 ports ethernet 1/1 to 1/4 ethernet 3/1 to 3/4
 primary-port 1/1
 deploy
!
!

Change primary port
===================

device(config-lag-test)#primary-port 3/3

SYSLOG: <14>Jul 18 12:47:57 LAG: test (id=10) , Configured primary port has been changed from 1/1 to 3/3


Verify running configuration with new primary port
==================================================

device(config-lag-test)#show running-config lag test
!
lag port-primary-dynamic
lag "test" dynamic id 10
 ports ethernet 1/1 to 1/4 ethernet 3/1 to 3/4
 primary-port 3/3
 deploy
!
!
!
```

Limitations

Dynamic primary port selection has the following limitations:

- Dynamic primary port selection is available on MLX Series devices only.
- Dynamic port selection and migration of the configurations to the newly elected primary port is supported on Layer 2 protocols such as PBR, L2 ACL, UDA ACL, PBR , transparent VLAN flooding (TVF) load balancing, and link fault signaling.
- Dynamic port selection is not supported on Layer 2 protocols such as L2VPN VLAN, MPORT, GPRS Tunneling Protocol (GTP) profiles, L2 MCT, STP, RSTP, MSTP, MRP, Ethernet ring, port loop detect, multicast protocol and configurations [IGMP and others], Provider Backbone Bridges (PBB) and Virtual Switch Redundancy Protocol (VSRP), and Uni-directional Link Detection (UDLD) protocols.

  NOTE
  The device does not check for these configuration conditions.

- Though dynamic port selection is not supported on QoS policies and rate limiting features, it does not impact their functionality.
- Dynamic port selection is supported on Layer 3 routing protocols, MPLS, and ACL configuration running on VE where LAG is member of the VE. The Layer 3 configurations that are running directly on the LAG or PHY interface are not supported. The Layer 3 multicast protocol and configurations such as PIM over MCT, IGMP over MCT, and others are not supported.

Upgrade/ downgrade considerations

Dynamic primary port selection has the following upgrade/ downgrade considerations:

- It is not supported on NetIron devices running versions earlier to NI 6.0.00a.
- Since the new configuration commands are saved in the configuration, they are ignored for downgrades of software and there is no impact to the behavior in previous releases prior to NI 6.0.00a release.
- During a NI device downgrade, since the dynamically selected primary port is present in startup configuration, the primary port configuration is retained.

  NOTE
  To find the full description of a specific command, including all required and optional keywords and variables, refer to the *Extreme NetIron Command Reference* for your software release.

Configurations allowed with dynamic primary port selection

- Deploying a LAG without specifying the primary port is supported.

```
device(config)# lag 1
device(config-lag-1)# ports ethernet 1/1 ethernet 1/2
device(config-lag-1)# deploy
```

- If we remove the port old primary 1/1 along with old member port 1/2 from the LAG, then the next available port in the LAG (least port ID) 2/1 is selected as the configured primary port (also active primary port) with no effect to VE and applications running on VE.

```
device(config)# lag 1
device(config-lag-1)# no ports ethernet 1/1 ethernet 1/2
```

  NOTE
  From the original configuration, if we remove the old primary port 1/1 alone from LAG, then the next available port in the LAG (least port ID) say 1/2 is selected as the primary port.

```
device(config)# lag 1
device(config-lag-1)# no ports ethernet 1/1
```

- Configure the existing LAG member port as the new primary port.

```
device(config)# lag 1
device(config-lag-1)# primary-port 2/1
```

- Remove the existing LAG primary port.

```
device(config)# lag 1
device(config-lag-1)# no primary-port 1/1
```

  NOTE
  If we remove the old primary port 1/1, then the next available port in the LAG (least port ID) 1/2 is selected as the primary port.

Assigning a name to a port within a LAG

You can assign a name to an individual port within a LAG using the **port-name** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# port-name orange ethernet 3/1
```

Syntax: [no] port-name text ethernet [**slot/port**]

The *text* variable specifies the port name. The name can be up to 50 characters long.

Use the **ethernet** option with the appropriate *slot/ port* variable to apply the specified name to an Ethernet port within the LAG.

NOTE
The port name and LAG name cannot use the same name.

Enabling sFlow forwarding on a port in a LAG

You can enable sFlow forwarding on an individual port within a LAG using the **sflow-forwarding** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-forwarding ethernet 3/1
```

Syntax: [no] sflow-forwarding ethernet [**slot/port**] | **port-name** [**text**]

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to enable sFlow forwarding for.

Use the **port-name** option with the appropriate *[text]* variable to specify a named port within the LAG that you want to enable sFlow forwarding for.

Setting the sFlow sampling rate for a port in a LAG

NOTE
The CES 2000 Series and CER 2000 Series devices support sflow sampling rate configuration per port basis. The MLX Series and XMR Series devices support sflow sampling rate configuration per packet processer basis.

You can set the sFlow sampling rate for an individual port within a LAG using the **sflow-subsampling** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-subsampling ethernet 3/1 512
```

Syntax: [no] sflow-subsampling ethernet [**slot/port**] | **port-name** [**text**] **num**

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify the Ethernet port within the LAG that you want to configure the sampling rate for.

Use the **port-name** option with the appropriate *[text]* variable to specify the named port within the LAG that you want to configure the sampling rate for.

The *num* variable specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. This can be a value between 512 - 1048576.

Configuring a dynamic LAG within a VRF

When configuring a dynamic LAG within a VRF, the following conditions must be considered:

- The dynamic LAG must be configured before adding it to a VRF.
- Before the LAG is deployed, all members must be in the default VRF.
- After the LAG is deployed, all LAG ports are in the LACP BLOCK state until the LACP protocol completes negotiation with the other end of the LAG.
- Once the LACP protocol negotiation is completed with the other end of the LAG, all the LAG ports are set to the FORWARD state.
- When a dynamic LAG within a VRF is undeployed, the primary port will stay in the VRF where the LAG was configured and the secondary ports of the LAG will return to the default VRF.

The following example uses the LAG and VRF commands to configure a LAG within a VRF.

```
device(config)# lag red dynamic
device(config-lag-red)# primary-port 3/2
device(config-lag-red)# ports ethernet 3/1 ethernet 7/2
device(config-lag-red)# exit
device(config)# interface ethernet 3/2
device(config-if-e10000-3/2)# vrf forwarding VPN1
device(config)# lag red dynamic
device(config-lag-red)# deploy
```

Configuring multicast dynamic load rebalancing on a LAG

In multicast, each forwarding (S,G) entry that has a Link Aggregation Group (LAG) port as an Outgoing Interface (OIF) is allocated only one of the member ports of the LAG for forwarding purposes. This member port is referred to as the forwarding port of the OIF of the (S,G) entry. A LAG is said to have balanced Multicast flows if all its member ports carry outgoing traffic for the same number of forwarding entries.

There are two ways of applying this feature, through a global configuration command that will force dynamic load rebalancing at all times, or via an exec level command that will trigger dynamic load rebalancing on demand.

Limitations

- The load balancing metric is determined by the number of forwarding (S,G) entries per LAG member port. Dynamic rebalancing attempts to balance this metric. This does not necessarily guarantee that multicast traffic bandwidth (bytes/sec) will be equally balanced across all member ports.
- When dynamic rebalancing is taking place, there will be dropped packets. This is because as the forwarding port for a LAG OIF changes, the FID and the MVID of the forwarding entry change as well, thus requiring a reprogramming of the PRAM before the changes take effect. This results in a brief disruption in traffic.
- Multicast dynamic load rebalancing on a LAG is only available on the XMR Series and MLX Series. This feature is not available in the CES 2000 Series and CER 2000 Series.

Configuring multicast dynamic load rebalancing

Once the dynamic load rebalancing has been configured, anytime a port in a LAG interface becomes active or a new port is added to the LAG interface, multicast traffic over the LAG interface will be rebalanced across all VRFs.

Dynamic load rebalancing has to be explicitly enabled (on all trunks and on all VRFs in the system) using the **ip multicast-routing lag rebalance** command.

```
device(config)#ip multicast-routing lag rebalance
```

Syntax: [no] ip multicast-routing lag rebalance

When enabling dynamic load balancing the specific IP version must be indicated. Use **ip** for IPv4 traffic and **ipv6** for IPv6 traffic.

Displaying LAG information

You can display LAG information for the device in either a **full** or **brief** mode.

The following example displays the **brief** option of the **show lag** command.

```
device# show lag brief
Total number of LAGs   : 2, 100g : 2
Total number of deployed LAGs  : 2, 100g : 2
Total number of trunks created : 2 (254 total available), 100g : 2 (14 total available)
LACP System Priority / ID      :1 / 0024.3883.3600
LACP Long timeout      :90, default: 90
LACP Short timeout     :3, default: 3
LAG              Type     Deploy Trunk Primary       Port List
100g_lag         static      Y    1     3/1        e 3/1
10g_lag          static      Y    2     2/1        e 2/1
1g_lag           static      Y    3     1/21       e 1/21
lag2            dynamic    Y    4     3/2       e 3/2
```

Syntax: show lag brief

Table 4 describes the information displayed by the **show lag brief** command.

The following example displays the full option of the **show lag** command.

```
device# show lag
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of s created:3 (125 available)
LACP System Priority / ID:     0001 / 0004.80a0.4000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "d1" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 13/2 to 13/3 ethe 32/2
   Primary Port:  32/2
    Type:    hash-based
   LACP Key:      104
Deployment:   ID 3, Active Primary 3/2
Port  Link L2 State  Dupl Speed  Tag Priori MAC           Name
3/2   Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
13/3  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
32/2  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
13/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
13/3       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
32/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
=== LAG "e" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 2/1 ethe 2/3 ethe 2/5
   Primary Port:  2/3
    Type:    hash-based
   LACP Key:      105
Deployment:   ID 1
Port  Link L2 State  Dupl Speed  Tag Priori MAC           Name
2/1   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/3   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/5   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
```

```
2/1        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/3        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/5        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
```

Syntax: show lag lag-name [**ID**] [**name**] [**deployed**] [**dynamic**] [**Ethernet**] [**keep-alive**] [**static**]

Using command this without options displays information for all LAGs configured on the device.

The *lag-name* variable allows you to limit the display to information for a specific LAG.

The **ID** option displays the output for the LAG specified by the ID.

The **name** displays the output for the LAG specified by the LAG name.

The **deployed** option limits the display to LAGs that are currently deployed.

The **dynamic** option limits the display to dynamic LAGs.

The **Ethernet** option displays the output for the specified Ethernet port.

The **keep-alive** option limits the display to keep alive LAGs.

The **deployed** option limits the display to static LAGs.

Optional commands include:

Syntax: show lag id

Syntax: show lag id *num_id*

Syntax: show lag ethernet *slot/port*

To display long port names, **set-lag-port-mode-wid** command. This command is useful if the ports names are long. In wide mode, the complete port name will be displayed and port type will not be displayed. In standard (non-wide) mode, only a portion of the port name is displayed if the port name is long and the port type is displayed. The following example shows the wide-mode display of a **show lag** command.

: Wide-Mode Display

```
device(config)#set-lag-port-mode-wid
device((config)#sh lag
Total number of LAGs:          2
Total number of deployed LAGs: 2
Total number of trunks created:2 (126 available)
LACP System Priority / ID:     1 / 00da.1111.2200
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "1234567890$%'-_@~`!(){}^#&abcdefghijklmnopqrstuvwXYZ" ID 4 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/3 to 31/10 e 31/15 to 31/20
   Port Count:    14
   Primary Port:  31/3
   Trunk Type:    hash-based
   LACP Key:      101
Port Individual Configuration:
   Port Name
   31/3 test2
Deployment:  Trunk ID 4, Active Primary none, base fid: 0x0810
Port  Link Port-State  Speed Tag MAC            Name
31/3  DisabNone        None  No  00da.1111.27a2 test2
31/4  DisabNone        None  No  00da.1111.27a2
31/5  DisabNone        None  No  00da.1111.27a2
31/6  DisabNone        None  No  00da.1111.27a2
31/7  DisabNone        None  No  00da.1111.27a2
31/8  DisabNone        None  No  00da.1111.27a2
31/9  DisabNone        None  No  00da.1111.27a2
31/10 DisabNone        None  No  00da.1111.27a2
31/15 DisabNone        None  No  00da.1111.27a2
31/16 DisabNone        None  No  00da.1111.27a2
```

```
31/17 DisabNone        None  No  00da.1111.27a2
31/18 DisabNone        None  No  00da.1111.27a2
31/19 DisabNone        None  No  00da.1111.27a2
31/20 DisabNone        None  No  00da.1111.27a2
=== LAG "NN" ID 6 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/11 to 31/14 e 31/21 to 31/24
   Port Count:    8
   Primary Port:  31/11
   Trunk Type:    hash-based
   LACP Key:      100
Port Individual Configuration:
   Port Name
   31/11test
Deployment:  Trunk ID 6, Active Primary 31/12, base fid: 0x0800
Port  Link Port-State  Speed Tag MAC            Name
31/11 Up   Forward     1G    No  00da.1111.27aa test
31/12 Up   Forward     1G    No  00da.1111.27aa
31/13 Up   Forward     1G    No  00da.1111.27aa
31/14 Up   Forward     1G    No  00da.1111.27aa
31/21 Up   Forward     1G    No  00da.1111.27aa
31/22 Up   Forward     1G    No  00da.1111.27aa
31/23 Up   Forward     1G    No  00da.1111.27aa
31/24 Up   Forward     1G    No  00da.1111.27aa
```

Syntax: [no] set-lag-port-mode-wid

Table 4 describes the information displayed by the **show lag** command.

TABLE 4 Show LAG information

This field...	Displays...
Total number of LAGS	The total number of LAGs that have been configured on the device.
Total number of Deployed LAGS	The total number of LAGs on the device that are currently deployed.
Total number of LAGs Created	The total number of LAGs that have been created on the LAG. The total number of LAGs available are shown also. Since keep-alive LAGs do not use a LAG ID, they are not listed and do not subtract for the number of LAGs available.
LACP System Priority /ID	The system priority configured for the device.The ID is the system priority which is the base MAC address of the device.
LACP Long timeout	The number of seconds used for the LACP Long timeout mode. This is only applicable for dynamic or keep-alive LAGs.
LACP Short timeout	The number of seconds used for the LACP Short timeout mode. This is only applicable for dynamic or keep-alive LAGs.
LACP BPDU Forwarding	Status of LACP BPDU forwarding on a static LAG: Disabled- LACP BDPU forwarding is disabled for all ports of the LAG, default setting. Enabled- LACP BDPU forwarding is enabled for all ports of the LAG.
The following information is displayed per-LAG in the show lag brief command.	
LAG	The name of the LAG.
Type	The configured type of the LAG: static, dynamic, or keep-alive
Deploy	Status of LAG deployment: Y - yes, LAG is deployed. N - no, LAG is not deployed.
LAG	The LAG ID number.
Primary	The primary port of the LAG.
Port List	The list of ports that are configured in the LAG.

TABLE 4 Show LAG information (continued)

This field...	Displays...
The following information is displayed per-LAG the show lag command for each LAG configured.	
LAG Configuration	
Ports:	List of ports configured with the LAG.
Primary Port:	The primary port configured on the LAG.
LAG Type:	The load sharing method configured for the LAG: either hash-based or per-packet.
LACP Key	The link aggregation key for the LAG.
Deployment	
LAG ID	The LAG ID number.
Active Primary	The port within the LAG where most protocol packets are transmitted. This is not the same as the configured Primary Port of the LAG.
Port	The chassis slot and port number of the interface.
Link	The status of the link which can be one of the following: • up • down
L2 State	The Layer 2 state for the port.
Dupl	The duplex state of the port, which can be one of the following: • Full • Half • None
Speed	The bandwidth of the interface.
LAG	The LAG ID of the port.
Tag	Indicates whether the ports have 802.1q VLAN tagging. The value can be Yes or No.
Priori	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 - 7.
MAC	The MAC address of the port.
Name	The name (if any) configured for the port.
Sys P	Lists the system priority configured for the device.
Port P	Lists the port's link aggregation priority.
Key	Lists the link aggregation key.
Act	Indicates the link aggregation mode, which can be one of the following: • No - The mode is passive on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link. • Yes - The mode is active. The port can send and receive LACPDU messages.
Tio	Indicates the timeout value of the port. The timeout value can be one of the following: • L - Long. The LAG group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used as confirmation of the health of the aggregate link.

TABLE 4 Show LAG information (continued)

This field...	Displays...
	• S - Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	Indicates the link aggregation state of the port. The state can be one of the following: • Agg - Link aggregation is enabled on the port. • No - Link aggregation is disabled on the port.
Syn	Indicates the synchronization state of the port. The state can be one of the following: • No - The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a LAG link. • Syn - The port is in sync with the remote port. The port understands the status of the LACPDU message exchange process, and therefore knows the LAG group to which it belongs, the link aggregation state of the remote port, and so on.
Col	Indicates the collection state of the port, which determines whether the port is ready to send traffic over the LAG link: • Col - The port is ready to send traffic over the LAG link. • No - The port is not ready to send traffic over the LAG link.
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the LAG link. • Dis - The port is ready to receive traffic over the LAG link. • No - The port is not ready to receive traffic over the LAG link.
Def	Indicates whether the port is using default link aggregation values. The port uses default values if it has not received link aggregation information through LACP from the port at the remote end of the link. This field can have one of the following values: • Def - The port has not received link aggregation values from the port at the other end of the link and is therefore using its default link aggregation LACP settings. • No - The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values: • Exp - The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No - The link aggregation values that this port negotiated with the port at the other end of the link have not expired. The port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a LAG group. An LACP port is blocked until it becomes part of a LAG. Also, an LACP is blocked if its state becomes

TABLE 4 Show LAG information (continued)

This field...	Displays...
	"default". To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.

Displaying LAG statistics

You can display LAG statistics for a device in either a **full** or **brief** mode. Full mode is the default and is displayed when the **show statistics lag** command is executed without the **brief** option. The examples below show both options of the **show statistics lag** command.

```
device# show statistics brief lag
LAG                  Packets            Collisions                  Errors
                      [Receive            Transmit]   [Recv  Txmit]      [InErr OutErr]
        LAG d1         1173               1018        0       0        0       0
         LAG e         1268               1277        0       0        0       0
device# show statistics lag
LAG d1 Counters:
         InOctets             127986          OutOctets             107753
           InPkts               1149            OutPkts                996
  InBroadcastPkts                 0    OutBroadcastPkts                  0
  InMulticastPkts               852    OutMulticastPkts                684
    InUnicastPkts               297      OutUnicastPkts                312
       InDiscards                 0         OutDiscards                  0
         InErrors                 0           OutErrors                  0
     InCollisions                 0       OutCollisions                  0
                                      OutLateCollisions                  0
        Alignment                 0                 FCS                  0
        GiantPkts                 0           ShortPkts                  0
     InBitsPerSec                 0       OutBitsPerSec                  0
     InPktsPerSec                 0       OutPktsPerSec                  0
    InUtilization              0.0%      OutUtilization               0.0%
```

Available variations of this command include:

Syntax: show statistics [**brief**] **lag** [*lag_name*]

Syntax: show statistics brief lag

Syntax: show statistics brief lag *lag_name*

Syntax: show statistics lag

Syntax: show statistics lag *lag_name*

Displaying multicast LAG member port usage

Use the **show ip pim count lag** command to display the multicast LAG member port usage. The Forwarding entries correlate to the multicast (S, G) entries.

```
device# show ip pim count lag lag-member-port
PORT ID FORWARDING ENTRIES
e2/43 0
e2/44 0
e2/45 0
e2/46 0
```

Syntax: show ip pim count lag lag-member-port

For IPv6, use the **show ipv6 pim count lag** command.

Enter a LAG member port in the *lag-member-port* parameter.

Displaying LAG information for a specified LAG name or LAG ID

This **show interface lag** command displays LAG information of a LAG specified by the LAG name or LAG ID. Detailed information about each LAG interface, including counters, is displayed.

```
device# show interface lag lag1
Total number of LAGs: 1
Total number of deployed LAGs: 1
Total number of trunks created:1 (127 available)
LACP System Priority / ID: 1 / 0000.0001.c000
LACP Long timeout: 90, default: 90
LACP Short timeout: 3, default: 3
=== LAG "lag1" ID 123 (static Deployed) ===
LAG Configuration:
Ports: e 1/1 to 1/2
Port Count: 2
Primary Port: 1/1
Trunk Type: hash-based
Deployment: Trunk ID 123, Active Primary none, base fid: 0x0800
Port Link Port-State Dupl Speed Trunk Tag Priori MAC Name Type
1/1 DisabNone None None 123 No level0 0000.0001.c000
default-port
1/2 DisabNone None None 123 No level0 0000.0001.c000
default-port
LAG lag1 Counters:
InOctets 2237519128754 OutOctets 1050988054740
InPkts 1968838581 OutPkts 2030408443
InBroadcastPkts 0 OutBroadcastPkts 0
InMulticastPkts 0 OutMulticastPkts 0
InUnicastPkts 1968838581 OutUnicastPkts 2030448142
InDiscards 0 OutDiscards 0
InErrors 0 OutErrors 0
InCollisions 0 OutCollisions 0
OutLateCollisions 0
Sample Output Cont....
Alignment 0 FCS 0
GiantPkts 0 ShortPkts 0
InBitsPerSec 782177316 OutBitsPerSec 466226351
InPktsPerSec 90896 OutPktsPerSec 99992
InUtilization 7.96% OutUtilization 4.82%
```

Syntax: show interface lag lag-name

The *lag-name* or *lag ID* parameter can be used to display the detailed information of a specified the LAG. If no LAG name or LAG ID is specified, the detailed information of all the LAGs configured in the system will be displayed.

Displaying the running configuration for a LAG

The **show running-config lag** command displays the running configuration for a specified LAG or all LAGs as specified in the parameters.

```
device# show running-config lag detailed
!
lag "lag1" static id 1
ports ethernet 1/1
ports ethernet 1/2
ports ethernet 1/3
primary-port 1/1
deploy
!
lag "lag2" static id 2
ports ethernet 1/4
primary-port 1/4
```

Syntax: show running-config lag lag name

The *lag name* option displays the running configuration for the specified LAG. The *lag id* option may also be used to display the same information.

Use the *detailed* option to display the running-config on a specific *lag name* or *lag id*. If no LAG name or LAG id is specified, the information of the entire LAG configured in the system will be displayed.

Available variations of the command include:

Syntax: show running-config lag

Syntax: show running-config lag detailed

Syntax: show running-config lag detailed lag id

Syntax: show running-config lag detailed lag name

Syntax: show running-config lag lag id

Syntax: show running-config lag lag name

Displaying LACP information for a specified LAG name or LAG ID

Use the **show lacp** command to display LACP information for a specified LAG name or LAG ID. For each LAG port configured, the **show lacp** command displays the system identifier, system priority, port priority, and various state machine variables for both the actor and the partner of the system. The **show lacp** command also displays the LACP packets received on a port, LACP packets transmitted on a port, marker packets received on a port, and LACP error packets received on a port.The following example output displays LACP information for LAG ID 4.

NOTE

The **show lacp** command is supported on MLX Series, XMR Series, CER 2000 Series, and CES 2000 Series devices.

```
device#show lacp lag_id 4
[ACTR - ACTOR] [PRTR - PARTNER] [Act - Activity] [Tio - Timeout]
[Agg - Aggregation] [Syn - Synchronization] [Col - Collecting] [Dis - Distributing] [Def - Defaulted] [Exp
- Expired] [Ope - Operating]
=== LAG "e4-10g-1" ID 4 ===
Port Role Sys Port Oper [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope][Port]
          Pri Pri  Key                                                 Num

6/1  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   240
6/1  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   96
6/2  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   241
6/2  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   97
6/3  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  242
6/3  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  146
6/4  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  243
6/4  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  147
Actor     System MAC: 001b.ed04.3a00
Port  Partner                          LACP             LACP      LACP      MARKER
      System MAC           Rx Count    Tx Count  Err Count RX Count
6/1   0012.f2f7.3b00                   1495                1518       0         0
6/2   0012.f2f7.3b00          1496                                    1520        0         0
6/3   0012.f2f7.3b00          1497                                 1519             0         0
6/4   0012.f2f7.3b00          1499                                    1520       0         0
```

Syntax: show lacp [**lag_id** *number* | **lag_name** *name*]

The **lag_id** *number* parameter specifies the ID of the LAG you want to display.

The **lag_name** *name* parameter specifies the name of the LAG you want to display.

Use the **show lacp** command without any options to display LACP information for all dynamic or deployed LAGs configured on the system.

Table 5 displays the output information from the **show lacp** command.

TABLE 5 Output from the show lacp command

This Field...	Displays...
Port	Lists the port number of the LAG member. Displayed in slot/port format.
Role	Indicates if the LACP information displayed for a LAG port is for the actor or the partner.
System Priority (Sys Pri)	Lists the system priority configured for this port. The device with the lower system priority value takes the higher priority to be removed at the other end of the link. For example, a device with a system priority value of 1 has a higher priority to be removed than a device with a system priority value of 10. The system priority and System MAC address together form the system identifier of a LAG.
Port Priority (Port Pri)	Lists the priority value configured for this port. The port priority and the port number together form the port identifier of a LAG. The greater the port priority value, the greater the chances are for transmission. Data traffic flow is distributed across multiple links on a LAG. The distribution of data traffic between links on a LAG is based on the port priority value. To configure the port priority value for a dynamic LAG or a keep-alive LAG, use the **lacp-port-priority** command. The priority value range is from 0 through 65535.
Operational Key (Oper key)	Lists the operational key value of a port. All ports on the LAG have the same key value. All ports with the same operational key value are aggregatable. The operational key value is assigned to the Ethernet link by the actor link. The key is dynamically generated based on the various port properties.
LACP_Activity (Act)	Indicates the control state of the link, which can be one of the following: • Yes - The link is active. The port can send and receive LACPDU messages. • No - The link is passive. The port does not initiate LACP messages, but will respond to LACP messages received.
LACP_Time (Tio)	Indicates the timeout value of the port. The timeout control value specifies the periodic transmission interval for LACP packets. The timeout control value can be set for a short timeout (3 seconds) or a long timeout (90 seconds). The LACP timeout control value is configurable using the **lacp-timeout** command. If the LACPDU is not received within the timeout mode configured on the port, the LACP will time out. If "S" is displayed, a short timeout value is used for the link. If "L" is displayed, a long timeout value is used for the link.
Aggregation (Agg)	Indicates the link aggregation state of the port. The state can be one of the following: • Agg - Link aggregation is enabled on the port. • No - Link aggregation is disabled on the port.
Synchronization (Syn)	Indicates whether the link is allocated to the correct Link Aggregation Group. The link Aggregation Group is associated with a compatible aggregator to form the link aggregation. The identity of the group is consistent with the System ID and the operational key information that is transmitted. If "Syn" is displayed, the system considers this link to be IN_SYNC, and the link is allocated to the correct Link Aggregation Group.

TABLE 5 Output from the show lacp command (continued)

This Field...	Displays...
	If "No" is displayed, the link is OUT_OF_SYNC, and the link is not in sync with aggregation.
Collecting (Col)	The collection of incoming frames that are enabled or disabled on the link. If "Col" is displayed, incoming frames is enabled on the link. If "No" is displayed, incoming frames is disabled on the link.
Distributing (Dis)	The distribution of outgoing frames that are enabled or disabled on the link. If "Dis"is displayed, the distribution of outgoing frames are enabled. If "No' is displayed, the distribution of outgoing frames are disabled.
Defaulted (Def)	Defaulted partner information is the set of LACP partner information (the system priority, key, port priority, and state of the partner) that is used when the information is not obtained from the partner through LACPDUs. This occurs when LACPDUs are not properly received on time. When "Def" is displayed, the actor's receive state machine is using the defaulted partner information that is configured administratively. If "No" is displayed, the actor's receive state machine is using the partner operational parameters that is received in a LACPDU.
Expired (Exp)	Indicates the state of the actor receive machine. If the LACP receive machine does not receive any LACPDUs within the timeout period configured on a port, the LACP receive machine goes into an EXPIRED state. The EXPIRED state indicates that the LAG has stopped operating. When the actor receive machine starts receiving LACPDUs from the port at the other end of the link, it will begin operating again. When "Exp" is displayed, the actor receive machine is in the EXPIRED state. When "No" is displayed, the actor receive machine is operating and is not in the EXPIRED state.
Operating (Ope)	Indicates the operating status of the port. The port status can be one of the following: • Ope (operating) - The port is operating normally. • Ina (inactive) - The port is inactive because the port on the other side of the link is down or has stopped transmitting LACP packets. • Blo (blocked) - The port is blocked because the adjacent port of the LAG is not configured with link aggregation. To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.
Port Number (Port Num)	The port number of the LAG.
Actor System MAC	The system MAC address of the actor. The system priority and system MAC address together form the system identifier.
Partner System MAC	The system MAC address of the partner. The system priority and system MAC address together form the system identifier.
LACP RX Count	The number of LACP packets received on a port.
LACP TX Count	The number of LACP packets sent on a port.
LACP Err Count	LACP is under the category of slow protocols. Slow protocol packets are received with an illegal subtype and a reserved subtype. LACP uses subtype 1, and the marker protocol uses subtype 2. Subtype values from 3 to 10 are reserved for future use. Subtype values 0 and 11 through 255 are considered illegal subtype values.
Marker RX Count	The number of marker packets received on a port. When a link is no longer aggregated with the port on the other end of the link, the marker protocol is used to verify that the conversation between the actor and the partner was successful on both ends. The actor and the partner exchange Marker PDUs and Marker Response PDUs to confirm the process.

Error messages displayed for LACP information when specifying a LAG name or LAG ID

If you do not configure a specified LAG name or LAG ID, an error message displays on the console, as shown in the following example.

```
device#show lacp lag_id 5
Error: LAG ID 5 is not configured
```

If you do not deploy a specified LAG name or LAG ID, an error message displays on the console, as shown in the following example.

```
device(config-lag-to-MLX2)#show lacp lag_id 1
Error: LAG 1 is not deployed
```

If you specify a LAG name or a LAG ID, and it is not a dynamic LAG, an error message displays on the console, as shown in the following example.

```
device(config-lag-abcd)#show lacp lag_id 4
Error: LAG 4 is not a dynamic LAG
```

Clearing LACP counter statistics for a specified LAG name or LAG ID

To clear LACP counter statistics for a specified LAG name or LAG ID, or for all LAGs in the system, enter the **clear lacp counters** command. The **clear lacp counters** command clears LACP packets that are received and transmitted on a LAG, in addition to clearing the LACP error count and the LACP marker packets that are received on all ports of the LAG.

> NOTE
> The **clear lacp counters** command is supported on MLX Series, XMR Series, CER 2000 Series and CES 2000 Series devices.

```
device# clear lacp counters lag_id 1
```

Syntax: clear lacp counters [**lag_id** *number* | **lag_name** *name*]

The **lag_id** *number* parameter specifies the ID of the LAG for which you want to clear statistics.

The **lag_name** *name* parameter specifies the name of the LAG for which you want to clear statistics.

Use the **clear lacp counters** command without any options to clear all dynamic and deployed LAG statistics.

> NOTE
> Configuring a port as a member of an undeployed or deployed LAG resets LACP counter statistics to 0. Enabling or disabling a port does not clear LACP counters. After a switchover, LACP counter statistics display in the standby management module.

CES 2000 Series and CER 2000 Series Link Aggregation

CES 2000 Series and CER 2000 Series Link Aggregation overview

This chapter describes how to configure Link Aggregation Groups (LAG) for CES 2000 Series and CER 2000 Series devices.

NOTE
The terms LAG and LAG groups are used interchangeably in this guide.

Transparent forwarding of L2 and L3 protocols for CES and CER 2000 Series devices

Use the **forward-all-protocol** command to

- Add per port Layer 2 and Layer 3 (L2/L3) protocols ACL filters on the VLL end-point port (not on the MPLS normal interface).
- Add per port Layer 2 and Layer 3 protocols ACL filters on the normal L2 switching interface.

The command **no forward-all-protocol** removes the L2/L3 protocols ACL filters.

NOTE
The **forward-all-protocol** command is only applicable to the CER 2000 Series and CES 2000 Series devices.

To implement per port Layer 2 and Layer 3 (L2/L3) protocols ACL filters, enter a command similar to the following:

```
device(config)# int eth 1/1
device(config-if-e1000-1/1)# forward-all-protocol
```

Syntax: [no] forward-all-protocol

The command **no forward-all-protocol** deletes VLL endpoint port L2/L3 protocols ACL filters. For LAG, only the primary port needs to be configured.

NOTE
The **forward-all-protocol** command lets L2/L3 protocols on the port go with hardware forwarding without going to the CPU. If the **no forward-all-protocol** command is executed, the L2/L3 functions may be impacted.

The **show interfaces ethernet** *slot/port* command displays the configuration status of the **forward-all-protocol** command.

The following output example shows the **show interfaces ethernet** *slot/port* command with the **forward-all-protocol** command disabled.

```
device# show interfaces ethernet 1/1
GigabitEthernet1/1 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 001b.eda3.f841 (bia 001b.eda3.f841)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Member of 1 L2 VLAN(S) (tagged), port is in tagged mode, port state is Forwarding
  STP configured to ON, Priority is level0, flow control enabled
  Priority force disabled, Drop precedence level 0, Drop precedence force disabled
  dhcp-snooping-trust configured to OFF
  mirror disabled, monitor disabled
  LACP BPDU Forwarding:Disabled
  LLDP BPDU Forwarding:Disabled
  L2L3 protocols Forwarding:Disabled
  Not member of any active trunks
...
```

The following output example shows the **show interfaces ethernet** *slot/port* command with the **forward-all-protocol** command enabled.

```
device(config-if-e1000-1/1)# forward-all-protocol
device(config-if-e1000-1/1)# show interfaces ethernet 1/1
GigabitEthernet1/1 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 001b.eda3.f841 (bia 001b.eda3.f841)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Member of 1 L2 VLAN(S) (tagged), port is in tagged mode, port state is Forwarding
  STP configured to ON, Priority is level0, flow control enabled
  Priority force disabled, Drop precedence level 0, Drop precedence force disabled
  dhcp-snooping-trust configured to OFF
  mirror disabled, monitor disabled
  LACP BPDU Forwarding:Disabled
  LLDP BPDU Forwarding:Disabled
  L2L3 protocols Forwarding:Enabled
  Not member of any active trunks
...
```

The **forward-all-protocol** command forwards the following protocols by hardware instead of the CPU.

- For L2: UDLD (drop), FDP, CDP and MRP.
- For L3: IP broadcast (255.255.255.255), IP multicast ((224.0.0.x, 224.0.1.x) including RIP, OSPF, PIM, VRRP), ARP, DHCP, BOOTP, IS-IS, OSPF, ND6, RIPng, OSPFv3, PIMv6, anycast solicited node, DHCPv6.

  NOTE
  The **forward-all-protocol** command cannot be used on an interface running the above listed protocols because those protocol frames will not be processed by the CES/CER CPU.

LAG formation rules

NetIron OS software supports the use a single interface to configure any of the following LAG types:

- **Static LAGs** - Manually-configured aggregate link containing multiple ports.
- **Dynamic LAG** - Uses the Link Aggregation Control Protocol (LACP), to maintain aggregate links over multiple ports. LACP PDUs are exchanged between ports on each device to determine if the connection is still active. The LAG then shuts down ports whose connection is no longer active. A syslog message is generated when the LAG is brought down because of he trunk-threshold being reached.
- **Keepalive LAG** - Establishes a single connection between a single port on 2 devices. LACP PDUs are exchanged between the ports to determine if the connection between the devices is still active. If it is determined that the connection is no longer active, the ports are blocked.

Follow these rules when configuring LAGs:

- You cannot configure a port concurrently as a member of a static, dynamic, or keepalive LAG
- Any number or combination of ports between 1 and 12 within the same device can be used to configure a LAG. The maximum number of LAG ports is checked when adding ports.
- All ports configured in a LAG must be of equal bandwidth. For example all 10 Gbps ports.
- All ports configured in a LAG must be configured with the same port attributes.
- LAG formation rules are checked when a static or dynamic LAG is deployed.
- A LAG must have the primary port selected before it can be deployed.
- All ports configured in a LAG must reside in the same VLAN.
- All dynamic LAG ports must have the same LACP BPDU forwarding configuration.

Layer 2 requirements

The LAG is rejected if the LAG ports:

- Do not have the same untagged VLAN component.
- Do not share the same VLAN membership and do not share the same uplink VLAN membership.
- Are configured as MRP primary and secondary interfaces.
- LAG deployment will fail if the LACP BPDU forwarding is disabled on the primary port and enabled on one or more of the secondary ports.

Layer 3 requirements

The LAG is rejected if any secondary LAG ports have any Layer 3 configuration, such as IPv4s, OSPF, RIP, RIPng, IS-IS, etc.

Layer 4 (ACL) requirements

- All LAG ports must have the same ACL configurations; otherwise, the LAG is rejected.
- A LAG cannot be deployed if a member port has ACL-based mirroring configured.
- A port with ACL-based mirroring configured cannot be added to a LAG.
- The device can support from 1-64 manually-configured LAGs.
- Ports can be in only one LAG group. All the ports in a LAG group must be connected to the same device at the other end. For example, if port 1/4 and 1/5 in Device 1 are in the same LAG group, both ports must be connected to ports in Device 2 or in Device 3. You cannot have one port connected to Device 2 and another port connected to Device 3.
- All LAG member properties must match the primary port of the LAG with respect to the following parameters:
  - Port tag type (untagged or tagged port)
  - Port speed and duplex
  - TOS-based configuration - All ports in the LAG must have the same TOS-based QoS configuration before LAG deployment, During deployment the configuration on the primary port is replicated to all ports and when deployment is ended, each port inherits the same TOS-based QoS configuration.

You must change port parameters on the primary port. The software automatically applies the changes to the other ports in the LAG.

- Make sure the device on the other end of the LAG group can support the same number of links in the LAG group.
- Dynamic LAGs are not supported for ports that are member of a VLAN within an ESI. Static LAGs are supported in such configurations.

Figure 4 displays an example of a valid, keepalive LAG link between two devices. A keepalive LAG does not aggregate ports but uses LACP PDUs to check the connection status between the two devices at either end of a LAG.

FIGURE 4 Example of a 1-port keepalive LAG

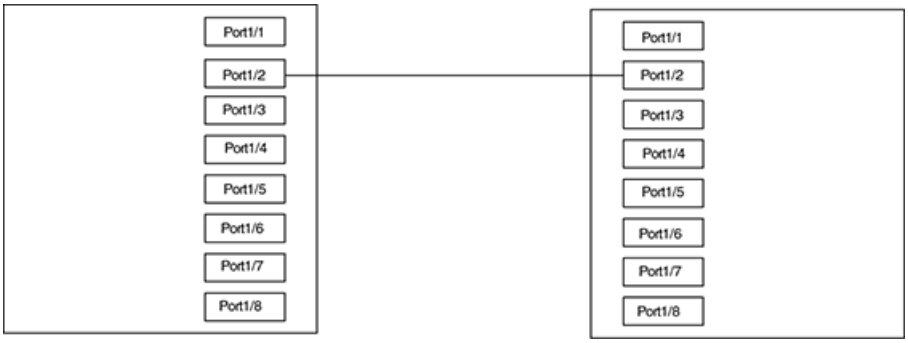
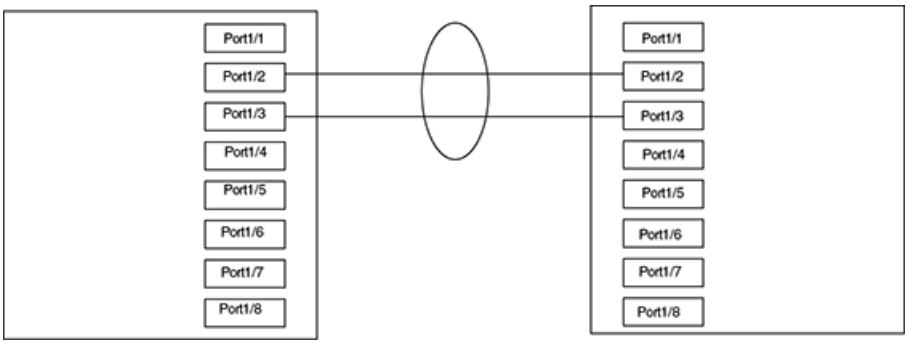


Figure 5 shows an example of a valid 2-port LAG between devices where the ports on each end are on the same interface module. Ports in a valid 2-port LAG on one device are connected to two ports in a valid 2-port LAG on another device.

FIGURE 5 Example of 2-port LAG



LAG load sharing

Extreme devices can be configured for load sharing over a LAG using hash-based load sharing.

Hash based load sharing

CES 2000 Series and CER 2000 Series devices share the traffic load evenly across the ports in LAG group, while ensuring that packets in the flow are not reordered. Individual flows are assigned a LAG index to identify them.

Traffic from each flow is then distributed across the ports in the LAG group using a hash index as follows:

The hash value is calculated based on the packet:

- hash[5:0] = MAC_SA[5:0]^MAC_DA[5:0];

If the packet is IP (v4 or v6), the hash is further calculated as the following:

- hash[5:0] = hash[5:0]^SIP[5:0]^SIP[21:16]^DIP[5:0]^DIP[21:16];

If the packet is TCP or UDP, the hash is further calculated as the following:

- hash[5:0] = hash[5:0]^L4_SrcPort[5:0]^L4_SrcPort[13:8]^L4_TrgPort[5:0]^L4_TrgPort[13:8];

If the packet is not IP:

- If the packet is MPLS, the hash is further calculated as the following:
  - hash[5:0] = hash[5:0]^MPLS_L0[5:0]^MPLS_L1[5:0]^MPLS_L2[5:0];
- The 4-bit hash value is: hash[3:0] = hash[5:0]% num_of__members;

NOTE
The hashing in CES/CER is based on the values present in the header of the incoming traffic (L2, L3, and L4). If the incoming traffic is equal to the total bandwidth of the outgoing LAG, there is no guarantee that the packet will be equally shared among the LAG links.

Not supported:

- Speculate UDP or TCP headers
- Mask Layer-4 source
- Hash diversification

Deploying a LAG

After configuring a LAG, you must explicitly enable it using the **deploy** command before it begins aggregating traffic. Once the **deploy** command is executed, the LAG enters aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG.

To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keepalive LAGs. Once a non keepalive LAG is deployed, a LAG is formed. If there is only one port in the LAG, a single-port LAG is formed. For a dynamic LAG, LACP is started for each LAG port. For a keepalive LAG, no LAG is formed and LACP is started on the LAG port.

You can deploy a LAG as shown for the "blue" LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
```

Syntax: [no] deploy [**forced** | **passive**]

When the **deploy** command is executed:

- For a static and dynamic LAGs, the LAG veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a LAG is formed with all the ports in the LAG.
- For dynamic LAGs, LACP is activated on all LAG ports. When activating LACP, use **active** mode if **passive** is not specified; otherwise, use **passive** mode.
- For keepalive LAGs, no LAG is formed, and LACP is started on the LAG port.
- Once the **deploy** command is issued, all LAG ports will behave like a single port.
- If the **no deploy** command is executed, the LAG is removed. For dynamic LAGs, LACP is deactivated on all LAG ports.
- If the **no deploy** command is issued and more than 1 LAG port is not disabled, the command is aborted and the following error message is displayed: "Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted." Using the **forced** keyword with the **no deploy** command in the previous situation, the LAG deployment is cancelled.

Commands available under LAG once it is deployed

Once a LAG has been deployed, the following configurations can be performed:

- Configuring ACL-based Mirroring
- Disabling Ports within a LAG
- Enabling Ports within a LAG
- Monitoring and Individual LAG Port
- Assigning a name to a port within a LAG
- Enabling sFlow Forwarding on a port within a LAG
- Setting the sFlow Sampling Rate for a port within a LAG

Configuring ACL-based mirroring

To configure ACL-based mirroring for all ports in a LAG, configure it on the primary port of the LAG at the interface configuration level (see Configuring IP Chapter). ACL-based mirroring can be configured for an individual member port within a LAG by using the **acl-mirror-port** command.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# acl-mirror-port ethe-port-monitored 3/1 ethernet 3/2
```

In this example, traffic on Ethernet port 3/1 (a secondary member port of LAG "blue") will be mirrored to Ethernet port 3/2.

Syntax: [no] acl-mirror-port { **ethe-port-monitored** *slot/port* | **named-port-monitored** *name* } **ethernet** *slot/port*

Use the **ethe-port-monitored** option with the appropriate *slot/port* variable to specify an Ethernet port for which you want to provide ACL mirroring.

Use the **named-port-monitored** option with the appropriate *slot/port* variable to specify a named port for which you want to provide ACL mirroring.

The **ethernet** keyword precedes the *slot/port* variable, identifying the port which will receive the mirrored packets.

A port with ACL-based mirroring already configured cannot be added to a LAG, and a LAG cannot be deployed if any member ports are configured for ACL-based mirroring. To use ACL-based mirroring on a LAG member port, deploy the LAG, then configure mirroring on the member port. If a port is removed from a LAG, ACL-based mirroring is removed from that port, and if a LAG is deleted mirroring is removed from all member ports.

Disabling ports within a LAG

You can disable an individual port within a LAG using the **disable** command within the LAG configuration.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# disable ethernet 3/1
```

Syntax: [no] disable ethernet [*slot/port*] | **named** *name*

Use the **ethernet** option with the appropriate *slot/port* variable to specify a Ethernet port within the LAG that you want to disable.

Use the **named** option with the appropriate *slot/port* variable to specify a named port within the LAG that you want to disable.

When a port is deleted from a deployed static LAG, the LACP BDPU forwarding state of the LAG will be retained for the deleted port.

Enabling ports within a LAG

You can enable an individual port within a LAG using the **enable** command.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# enable ethernet 3/1
```

Syntax: [no] enable ethernet [*slot/port*] | **named** *name*

Use the **ethernet** option with the appropriate *slot/port* variable to specify an Ethernet port to be enabled in the LAG.

Use the **named** option with the appropriate *slot/port* variable to specify a named port in the LAG that you want to enable.

When adding a port to a currently deployed dynamic LAG the LACP BPDU Forwarding configuration must be the same as the LAG. Follow the procedure Enabling and Disabling LACP BPDU Forwarding on a Port on page 64.

Monitoring an individual LAG port

By default, when you monitor the primary port in a LAG group, aggregated traffic for all the ports in the LAG is copied to the mirror port. You can configure the device to monitor individual ports in a LAG, including Ethernet, or named ports. You can monitor the primary port or a secondary port individually.

You can use only one mirror port for each monitored LAG port. To monitor traffic on an individual port in a LAG group, enter commands such as the following:

This command enables monitoring of an individual port within a LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# monitor ethe-port-monitored 3/1 ethernet 10/3 both
```

Syntax: [no] monitor ethe-port-monitored *slot/port* | **named-port-monitored** *name* | **ethernet** *slot/port* [**input** | **output** | **both**]

Use the **ethe-port-monitored** option with the appropriate *slot/port* variable to specify an Ethernet port in the LAG that you want to monitor.

Use the **named-port-monitored** option with the appropriate *slot/port* variable to specify a named port in the LAG that you want monitor.

The **ethernet** *slot/port* parameter specifies the port to which the traffic analyzer is attached.

The **input, output,** and **both** parameters specify the traffic direction to be monitored.

Naming a port in a LAG

You can name an individual port in a LAG using the **port-name** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# port-name orange ethernet 3/1
```

Syntax: [no] port-name text ethernet [**slot/port**]

The *text* variable specifies the port name. The name can be up to 50 characters long.

Use the **ethernet** option with the appropriate *[slot/port]* variable to apply the specified name to an Ethernet port within the LAG.

NOTE

The port name and LAG name cannot use the same name.

Enabling sFlow forwarding on a port in a LAG

You can enable sFlow forwarding on an individual port within a LAG using the **sflow-forwarding** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-forwarding ethernet 3/1
```

Syntax: [no] sflow-forwarding ethernet *slot/port* | **port-name** *name*

Use the **ethernet** option with the appropriate *slot/port* variable to specify an Ethernet port in the LAG where you want to enable sFlow forwarding.

Use the **port-name** option with the appropriate *name* variable to specify a named port within the LAG where you want to enable sFlow forwarding.

Setting the sFlow sampling rate for a port in a LAG

NOTE
The CES 2000 Series and CER 2000 Series devices support sflow sampling rate configuration per port basis. The MLX Series and XMR Series devices support sflow sampling rate configuration per packet processer basis.

You can set the sFlow sampling rate for an individual port within a LAG using the **sflow-subsampling** command as shown.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-subsampling ethernet 3/1 512
```

Syntax: [no] sflow-subsampling *num*

The *num* variable specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. This can be a value between 512 - 1048576.

Static LAG Considerations

Enabling and Disabling LACP BPDU Forwarding on a Port

NOTE
The forward-lacp command must be issued on the physical port configuration, not in LAG configuration.

For scenarios in which dynamic LAG ports require LACP BPDU packet forwarding, you can issue the **forward-lacp** command in the interface mode. Once LACP Forwarding has been enabled on a dynamic LAG, all the LACP BPDUs will follow regular packet forwarding actions.

When LACP forwarding is enabled, the link OAM packets received on the LACP forwarding enabled interface will be processed and flooded on the VLAN. If the LACP forwarding is not enabled, the link OAM packets will be processed and then dropped.

To enable LACP BPDU forwarding, enter the following command:

```
device(config-if-e1000-3/5)# forward-lacp
```

To disable LACP BPDU forwarding, enter the lacp-forwarding command as follows.

```
device(config-if-e1000-3/5)# [no]
forward-lacp
```

Enabling and Disabling LACP BPDU Forwarding on a Trunk

NOTE
The forward-lacp command must be issued on the physical port configuration, not in LAG configuration.

When the LACP forwarding is enabled on the primary port of the dynamic LAG, the LACP BPDU forwarding is enabled on all ports of the LAG when the LAG is deployed. When the static LAG is undeployed the BPDU forwarding state is retained.

- If LACP BPDU forwarding is enabled on the primary and secondary ports, the LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is enabled on the primary port and disabled on the secondary ports, the LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is disabled on the primary and secondary ports, the LAG deployment will be successful and LACP BPDU forwarding will be disabled on the LAG ports.

NOTE
LACP BPDU forwarding is not supported for any port of dynamic or keep alive LAGs.

Displaying LAG information

You can display LAG information for by entering the **show lag** command.

```
device# show lag
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of s created:3 (125 available)
LACP System Priority / ID:     0001 / 0004.80a0.4000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "d1" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 13/2 to 13/3 ethe 32/2
   Primary Port:  32/2
    Type:    hash-based
   LACP Key:      104
Deployment:   ID 3, Active Primary 3/2
Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
3/2   Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
13/3  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
32/2  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
13/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
13/3       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
32/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
=== LAG "e" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 2/1 ethe 2/3 ethe 2/5
   Primary Port:  2/3
    Type:    hash-based
   LACP Key:      105
Deployment:   ID 1
Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
2/1   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/3   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/5   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
```

```
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
2/1         1         1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/3         1         1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/5         1         1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
```

Syntax: show lag *lag-name* [**brief**] [**deployed**] [**dynamic**] [**ID**] [**Ethernet**] [**keepalive**] [**static**]

Using this command without options displays information for all LAGs configured on the device. In addition, the following arguments may be used to provide additional LAG information.

The *lag-name* variable allows you to limit the display to information for a specific LAG.

The **ID** option will display the **show lag** command output for the LAG specified by the ID.

The **brief** option displays summary information for any or all configured LAGs.

The **deployed** option limits the display to LAGs that are currently deployed.

The **dynamic** option limits the display to dynamic LAGs.

The **Ethernet** option displays the output for the specified Ethernet port.

The **keep-alive** option limits the display to keep alive LAGs.

The **deployed** option limits the display to static LAGs.

Table 6 describes the information displayed by the **show lag** command.

TABLE 6 Show LAG information

This field...	Displays...
Total number of LAGS	The total number of LAGs that have been configured on the device.
Total number of Deployed LAGS	The total number of LAGs on the device that are currently deployed.
Total number of LAGs Created	The total number of LAGs that have been created on the LAG. The total number of LAGs available are shown also. Since keepalive LAGs do not use an ID, they are not listed and do not subtract for the number of LAGs available.
LACP System Priority /ID	The system priority configured for the device.The ID is the system priority which is the base MAC address of the device.
LACP Long timeout	The number of seconds used for the LACP Long timeout mode. This is only applicable for dynamic or keepalive LAGs.
LACP Short timeout	The number of seconds used for the LACP Short timeout mode. This is only applicable for dynamic or keepalive LAGs.
The following information is displayed per-LAG in the show lag brief command.	
LAG	The name of the LAG.
Type	The configured type of the LAG: static, dynamic, or keepalive
Deploy	Status of LAG deployment: Y - yes, LAG is deployed. N - no, LAG is not deployed.
LAG	The LAG ID number.
Primary	The primary port of the LAG.
Port List	The list of ports that are configured in the LAG.
The following information is displayed per-LAG the show lag command for each LAG configured.	
LAG Configuration	
Ports:	List of ports configured in the LAG.
Primary Port:	The primary port for the LAG.

TABLE 6 Show LAG information (continued)

This field...	Displays...
LAG Type:	The load sharing method configured for the LAG: hash-based.
LACP Key	The link aggregation key for the LAG.
Deployment	
LAG ID	The LAG ID number.
Active Primary	The LAG port where most protocol packets are transmitted. This is not the same as the configured Primary Port of the LAG.
Port	The chassis slot and port number of the interface.
Link	The status of the link, which can be up or down.
L2 State	The Layer 2 state for the port.
Dupl	The duplex state of the port, which can be one of the following: • Full • Half • None
Speed	The bandwidth of the interface.
LAG	The LAG ID of the port.
Tag	Indicates whether the ports have 802.1q VLAN tagging. The value can be Yes or No.
Priori	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 - 7.
MAC	The MAC address of the port.
Name	The name (if any) configured for the port.
Sys P	The system priority configured for the device.
Port P	Link aggregation priority of the port
Key	Lists the link aggregation key.
Act	Indicates the link aggregation mode, which can be one of the following: • No - The mode is passive on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link. • Yes - The mode is active. The port can send and receive LACPDU messages.
Tio	Indicates the timeout value of the port. The timeout value can be one of the following: • L - Long. The LAG group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used to confirm the health of the aggregate link. • S - Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	Indicates the link aggregation state of the port. The state can be one of the following: • Agg - Link aggregation is enabled on the port. • No - Link aggregation is disabled on the port.

TABLE 6 Show LAG information (continued)

This field...	Displays...
Syn	Indicates the synchronization state of the port. The state can be one of the following: • No - The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a LAG link. • Syn - The port is in sync with the remote port. The port understands the status of the LACPDU message exchange process, and knows the LAG group to which it belongs, the link aggregation state of the remote port, and so on.
Col	Indicates the collection state of the port, which determines whether the port is ready to send traffic over the LAG link: • Col - The port is ready to send traffic over the link. • No - The port is not ready to send traffic over the link.
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the LAG link: • Dis - The port is ready to receive traffic over the LAG link. • No - The port is not ready to receive traffic over the LAG link.
Def	Indicates whether the port is using default link aggregation values. The port uses default values if it has not received link aggregation information through LACP from the port at the remote end of the link. This field can have one of the following values: • Def - The port has not received link aggregation values from the port at the other end of the link and is therefore using its default link aggregation LACP settings. • No - The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values: • Exp - The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No - The link aggregation values that this port negotiated with the port at the other end of the link have not expired, so the port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a LAG group. An LACP port is blocked until it becomes part of a LAG group. Also, an LACP is blocked if its state becomes "default". To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.

To display long port names, **set-lag-port-mode-wid** command. This command is useful if the ports names are long. In wide mode, the complete port name will be displayed and port type will not be displayed. In standard (non-wide) mode, only a portion of the port name is displayed if the port name is long and the port type is displayed. The following example shows the wide-mode display of a **show lag** command.

: Wide-Mode Display

```
device(config)#set-lag-port-mode-wid
device((config)#sh lag
Total number of LAGs:          2
Total number of deployed LAGs: 2
Total number of trunks created:2 (126 available)
LACP System Priority / ID:     1 / 00da.1111.2200
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "1234567890$%'-_@~`!(){}^#&abcdefghijklmnopqrstuvwXYZ" ID 4 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/3 to 31/10 e 31/15 to 31/20
   Port Count:    14
   Primary Port:  31/3
   Trunk Type:    hash-based
   LACP Key:      101
Port Individual Configuration:
   Port Name
   31/3 test2
Deployment:  Trunk ID 4, Active Primary none, base fid: 0x0810
Port  Link Port-State  Speed Tag MAC            Name
31/3  DisabNone        None  No  00da.1111.27a2 test2
31/4  DisabNone        None  No  00da.1111.27a2
31/5  DisabNone        None  No  00da.1111.27a2
31/6  DisabNone        None  No  00da.1111.27a2
31/7  DisabNone        None  No  00da.1111.27a2
31/8  DisabNone        None  No  00da.1111.27a2
31/9  DisabNone        None  No  00da.1111.27a2
31/10 DisabNone        None  No  00da.1111.27a2
31/15 DisabNone        None  No  00da.1111.27a2
31/16 DisabNone        None  No  00da.1111.27a2
31/17 DisabNone        None  No  00da.1111.27a2
31/18 DisabNone        None  No  00da.1111.27a2
31/19 DisabNone        None  No  00da.1111.27a2
31/20 DisabNone        None  No  00da.1111.27a2
=== LAG "NN" ID 6 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/11 to 31/14 e 31/21 to 31/24
   Port Count:    8
   Primary Port:  31/11
   Trunk Type:    hash-based
   LACP Key:      100
Port Individual Configuration:
   Port Name
   31/11test
Deployment:  Trunk ID 6, Active Primary 31/12, base fid: 0x0800
Port  Link Port-State  Speed Tag MAC            Name
31/11 Up   Forward     1G    No  00da.1111.27aa test
31/12 Up   Forward     1G    No  00da.1111.27aa
31/13 Up   Forward     1G    No  00da.1111.27aa
31/14 Up   Forward     1G    No  00da.1111.27aa
31/21 Up   Forward     1G    No  00da.1111.27aa
31/22 Up   Forward     1G    No  00da.1111.27aa
31/23 Up   Forward     1G    No  00da.1111.27aa
31/24 Up   Forward     1G    No  00da.1111.27aa
```

Syntax: [no] set-lag-port-mode-wid

Displaying LAG statistics

You can display LAG statistics in either **full** or **brief** mode. Full mode is the default and is displayed when the **show statistics lag** command is executed without the **brief** option. These examples show both options of the **show statistics lag** command.

```
device# show statistics brief lag
LAG                  Packets           Collisions                 Errors
                      [Receive           Transmit]   [Recv  Txmit]      [InErr OutErr]
```

```
        LAG d1          1173             1018        0       0        0       0
         LAG e          1268             1277        0       0        0       0
device# show statistics lag
LAG d1 Counters:
          InOctets             127986           OutOctets             107753
            InPkts               1149             OutPkts                996
  InBroadcastPkts                  0    OutBroadcastPkts                  0
  InMulticastPkts                852    OutMulticastPkts                684
    InUnicastPkts                297      OutUnicastPkts                312
       InDiscards                  0         OutDiscards                  0
         InErrors                  0           OutErrors                  0
     InCollisions                  0       OutCollisions                  0
                                       OutLateCollisions                  0
        Alignment                  0                 FCS                  0
        GiantPkts                  0           ShortPkts                  0
     InBitsPerSec                  0       OutBitsPerSec                  0
     InPktsPerSec                  0       OutPktsPerSec                  0
    InUtilization              0.0%      OutUtilization              0.0%
```

Syntax: show statistics [**brief**] **lag** [*lag-name*]

The following syntax options can be used for a brief display:

Syntax: show statistics brief lag

Syntax: show statistics brief lag *lagname*

The following syntax options can be used for a full display:

Syntax: show statistics lag

Syntax: show statistics lag *lagname*

Displaying LAG information for a specified LAG name or LAG ID

The **show interfaces lag** command displays LAG information for a LAG specified by LAG name or LAG ID. If no LAG name or LAG ID is specified, it shows detailed information of all the LAGs configured in the system. For each port of a LAG, detailed information about the LAG interface, including counters is displayed.

```
device#show interfaces lag
Total number of LAGs:          2
Total number of deployed LAGs: 2
Total number of trunks created:2 (62 available)
LACP System Priority / ID:     1 / 001b.edb3.f181
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "151-188" ID 2 (static Deployed) ===
LAG Configuration:
   Ports:         e 1/3 to 1/4 e 1/15 to 1/16 e 1/27 e 1/39 to 1/40
   Port Count:    7
   Primary Port:  1/3
   Trunk Type:    hash-based
   LACP BPDU Forwarding: Disabled
Deployment:  Trunk ID 2, Active Primary 1/3, base fid: 0x0000
Port  Link Port-State  Dupl Speed Trunk Tag Priori MAC            Name           Type
1/3   Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/4   Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/15  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/16  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/27  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/39  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/40  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
LAG 151-188 Counters:
          InOctets           71899883           OutOctets            6816239
            InPkts             865449             OutPkts              23691
  InBroadcastPkts              25684    OutBroadcastPkts                  0
  InMulticastPkts             839765    OutMulticastPkts              23691
    InUnicastPkts                  0      OutUnicastPkts                  0
```

```
       InDiscards                  0         OutDiscards                  0
         InErrors                  0           OutErrors                  0
     InCollisions                  0       OutCollisions                  0
                                       OutLateCollisions                  0
        Alignment                  0                 FCS                  0
        GiantPkts                  0           ShortPkts                  0
     InBitsPerSec               7846       OutBitsPerSec                  0
     InPktsPerSec                 10       OutPktsPerSec                  0
    InUtilization               0.0%      OutUtilization               0.0%
=== LAG "151-189" ID 1 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 1/1 to 1/2 e 1/13 to 1/14 e 1/25 to 1/26 e 1/37 to 1/38
   Port Count:    8
   Primary Port:  1/1
   Trunk Type:    hash-based
   LACP Key:      100
Deployment:  Trunk ID 1, Active Primary 1/1, base fid: 0x0000
Port  Link Port-State  Dupl Speed Trunk Tag Priori MAC            Name           Type
1/1   Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/2   Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/13  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/14  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/25  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/26  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/37  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/38  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1       1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/2       1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/13      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/14      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/25      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/26      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/
LAG 151-189 Counters:
         InOctets          13357835          OutOctets          73729556
           InPkts             54050            OutPkts            885774
  InBroadcastPkts                 0    OutBroadcastPkts             25682
  InMulticastPkts             54050    OutMulticastPkts            860092
    InUnicastPkts                 0      OutUnicastPkts                 0
       InDiscards                 0         OutDiscards                 0
         InErrors                 0           OutErrors                 0
     InCollisions                 0       OutCollisions                 0
                                       OutLateCollisions                 0
        Alignment                 0                 FCS                 0
        GiantPkts                 0           ShortPkts                 0
     InBitsPerSec                 0       OutB
```

Syntax: show interfaces lag *lag-name*

The *lag-name* or *lag ID* parameter can be used to display the detailed information of a specified the LAG.

Displaying the running configuration for a LAG

The **show running-config lag** command displays the running configuration for a specified LAG or all LAGs as specified in the parameters.

```
device# show running-config lag detailed
!
lag "lag1" static id 1
ports ethernet 1/1
ports ethernet 1/2
ports ethernet 1/3
primary-port 1/1
deploy
!
lag "lag2" static id 2
```

```
    ports ethernet 1/4
    primary-port 1/4
```

Syntax: show running-config lag *lagname*

The *lag name* option displays the running configuration for the specified LAG. The *lag id* option may also be used to display the same information.

Use the **detailed** option to display the running-config on a specific *lag name* or *lag id*. If no LAG name or LAG id is specified, the information of the entire LAG configured in the system will be displayed.

The following command options may be used:

Syntax: show running-config lag

Syntax: show running-config lag detailed

Syntax: show running-config lag detailed *lag_id*

Syntax: show running-config lag detailed *lagname*

Syntax: show running-config lag *lag_id*

Syntax: show running-config lag *lagname*

VLANs

VLANs overview

A Virtual Local Area Network (VLAN) lets you segment traffic in a network by placing ports and interfaces into separate broadcast domains. Each broadcast domain is uniquely identified by a VLAN ID. These broadcast domains can span multiple devices.

NOTE

The CES 2000 Series devices support the Ethernet Service Instance (ESI) framework. A user can configure ESIs in the process of configuring Provider Bridges and Provider Backbone Bridging. By default, the device has a "default ESI" configured in which VLANs 1 - 4090 exist. This chapter refers to configuration and use VLANs under the default ESI framework.

The Extreme device supports two types of VLANs: *port-based VLANs* and *protocol-based VLANs* . A port-based VLAN consists of interfaces that constitute a Layer 2 broadcast domain. (Protocol-based VLANs are described in Protocol-based VLANs on page 75.) By default, all interfaces on a device are members of the *default* VLAN, which is VLAN 1. Thus, by default, all interfaces on all devices on a network constitute a single Layer 2 broadcast domain. Once you create a port-based VLAN and assign an interface to that VLAN, that interface is automatically removed from the default VLAN if the interface is assigned to the VLAN as an untagged interface. If the interface is assigned as a tagged interface, then the interface is a member of both the default VLAN, and the VLAN to which it is assigned.

Tagged, untagged, and dual mode ports

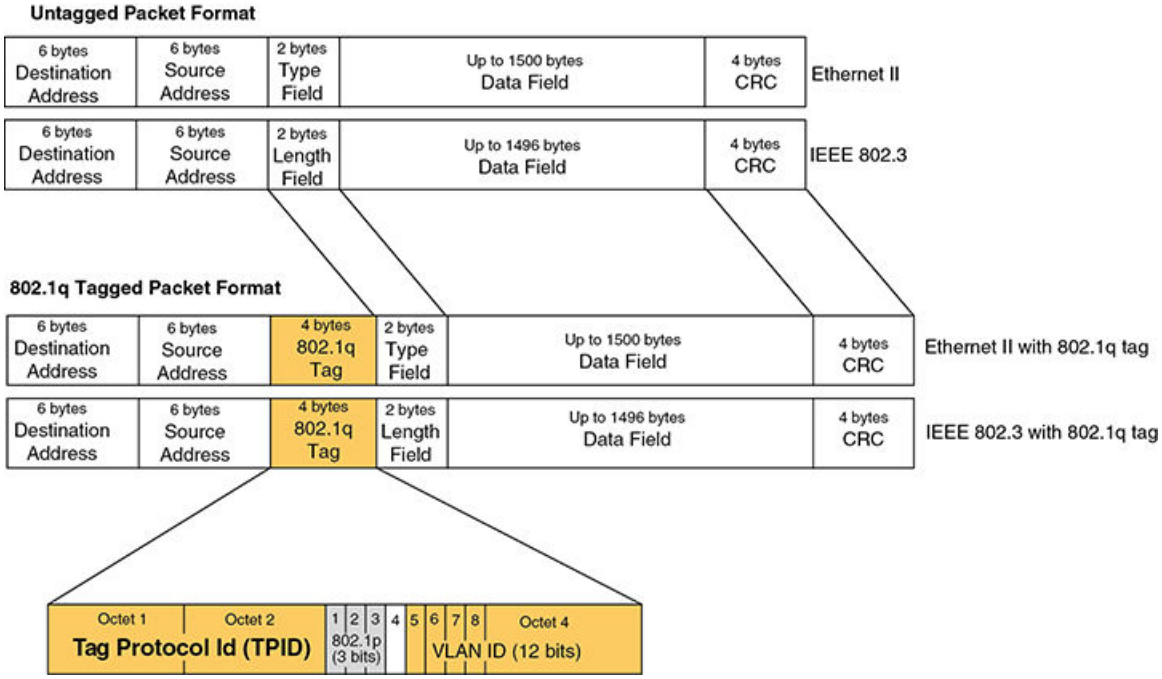
Interfaces assigned to port-based VLANs can be defined as untagged, tagged, and dual-mode ports. An untagged port is a member of only one VLAN, while a tagged port can be a member of more than one VLAN. Thus a tagged port can be a member of more than one broadcast domain. Dual-mode ports are configured by adding one or more tagged VLANs and one untagged VLAN to a port.

Tagged ports allow the Extreme device to add a four-byte 802.1q tag to the packet. 802.1q tagging is an IEEE standard that allows a networking device to add information to Layer 2 packets. This information identifies the VLAN membership of the packet, as well as the VLAN ID of the VLAN from which the packet is sent. Furthermore, the default tag value of the 802.1q tag is 8100 (hexadecimal). This value comes from the IEEE 802.1q specification. You can change this tag value on a per-port or on a global basis on a device if needed to be compatible with other vendors' equipment.

NOTE
On CES 2000 Series devices, you can change the tag value on the global basis for each VLAN component (B-VLAN, C-VLAN, or S-VLAN).

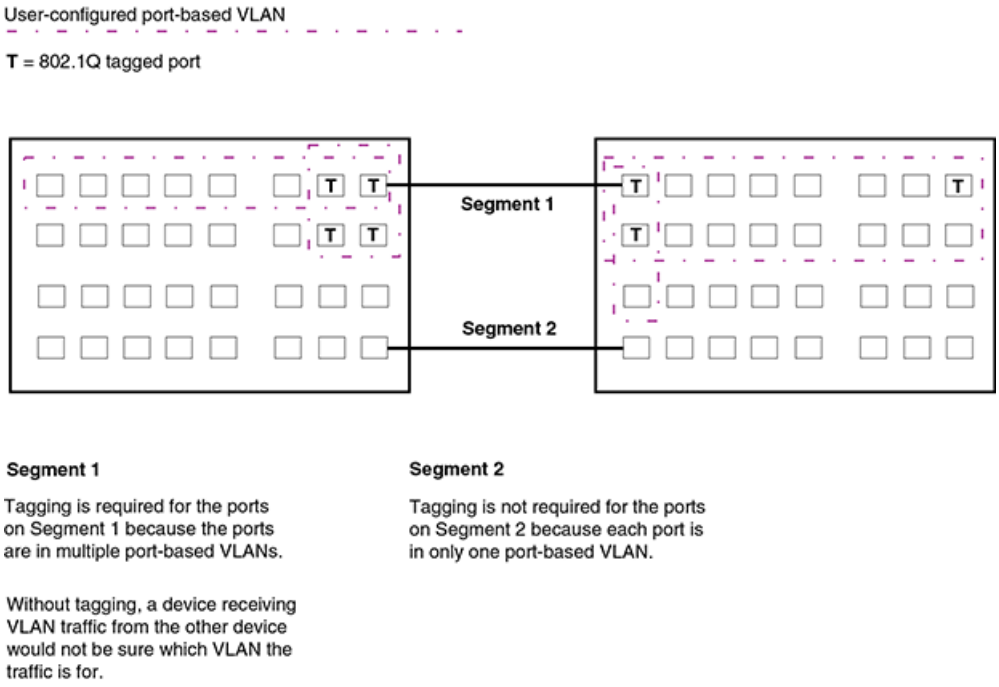
Figure 6 shows the format of packets with and without the 802.1q tag.

FIGURE 6 Packet containing Extreme's 802.1QVLAN tag



If you configure a VLAN that spans multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If a port connecting one device to the other is a member of only a single port-based VLAN, tagging is not required. Figure 7 shows an example of two devices that have the same Layer 2 port-based VLANs configured across them. Notice that only one of the VLANs requires tagging.

FIGURE 7 VLANs configured across multiple devices



Protocol-based VLANs

Interfaces that belong to a port-based VLAN can further be divided into Layer 3 broadcast domains by using protocol-based VLANs. Protocol-based VLANs accept broadcasts of a specified protocol type. For example, an IP subnet VLAN accepts broadcasts for the specified IP subnets only. This feature enables you to limit the amount of broadcast traffic to end-stations, servers, and routers.

You can configure the following protocol-based VLANs within a port-based VLAN in a device:

- **AppleTalk** - The device sends AppleTalk broadcasts to all ports within the AppleTalk protocol VLAN.
- **IP** - The device sends IP broadcasts to all ports within the IP protocol VLAN.
- **IPX** - The device sends IPX broadcasts to all ports within the IPX protocol VLAN.
- **IPv6** - The device sends IPv6 broadcasts to all ports within the IPv6 protocol VLAN.

  NOTE
  You can configure a protocol-based VLAN as a broadcast domain for IPv6 traffic. When the Extreme device receives an IPv6 multicast packet (a packet with 06 in the version field and 0xFF as the beginning of the destination address), the Extreme device forwards the packet to all other ports in the VLAN except to the port that received the packet.

Protocol-based VLANs can be configured to have *static* or *excluded* port memberships. Static ports are permanent members of a protocol-based VLAN. They remain active members of the protocol-based VLAN regardless of whether they receive traffic for the VLAN's protocol.

NOTE
The dynamic port membership is not supported on Extreme devices.

If you want to exclude certain ports in a port-based VLAN from protocol-based VLANs, the protocol-based VLAN can be explicitly configured to exclude those ports.

VLAN configuration rules

To create any type of VLAN on a device, Layer 2 forwarding must be enabled. When Layer 2 forwarding is enabled, the Extreme device becomes a switch on all ports for all non-routable protocols.

In addition to this rule, the sections below summarize the rules for configuring VLANs.

NOTE
To enable Layer 2 forwarding, use the **no route-only command.** On CES 2000 Series devices, Layer 2 forwarding is enabled by default.

VLAN ID range

The upper range of VLAN IDs available for user VLANs (including the default VLAN) has been reduced to 4090 (formerly 4094). The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

Tagged VLANs

When configuring VLANs across multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If you are configuring tagged VLANs across multiple devices, make sure all the devices support the same tag format.

VLAN hierarchy

A hierarchy of VLANs exists between the Layer 2 and Layer 3 protocol-based VLANs:

- Port-based VLANs are at the lowest level of the hierarchy.
- Layer 3 protocol-based VLANs are at the highest level of the hierarchy.

As a device receives packets, the VLAN classification starts from the highest level VLAN first. Therefore, if an interface is configured as a member of a port-based VLAN and a protocol-based VLAN, packets coming into the interface are classified as members of the protocol-based VLAN because that VLAN is higher in the VLAN hierarchy.

When a port in a VLAN receives a packet, the device forwards the packet based on the following VLAN hierarchy:

- If it is a Layer 3 packet and the port is a member of a Layer 3 protocol-based VLAN for the packet's protocol, the device forwards the packet on all the Layer 3 protocol-based VLAN ports that have been configured or drops the packet if the port is explicitly excluded from the protocol VLAN.
- If the packet cannot be forwarded based on its VLAN membership types but the packet can be forwarded at Layer 2, the device forwards the packet on all the ports within the receiving port's port-based VLAN.

Multiple VLAN membership rules

The multiple VLAN membership rules are listed below:

- A port can belong to multiple, overlapping Layer 2 port-based VLANs only if the port is a tagged port. Packets sent out of a tagged port use an 802.1q-tagged frame.
- A port can belong to multiple, unique, overlapping Layer 3 protocol-based VLANs.
- When both port and protocol-based VLANs are configured on a given device, all protocol-based VLANs must be strictly contained within a port-based VLAN. A protocol-based VLAN cannot include ports from multiple port-based VLANs. This rule is required to ensure that port-based VLANs remain loop-free Layer 2 broadcast domains.
- One of each type of protocol-based VLAN can be configured within each port-based VLAN on the Extreme device.
- Removing a configured port-based VLAN from a device automatically removes any protocol-based VLAN, or any virtual routing interfaces defined within the port-based VLAN.

Dual-mode default VLAN

As previously described, ports can be defined as dual-mode, which means that they can exist in both tagged and untagged VLANs. As such, they can coexist untagged in the default or a non-default VLAN and be added as a tagged port into non-default VLAN. One way that ports become dual-mode is by adding a port to a non-default, tagged VLAN. The normal behavior is for the port to remain in the default VLAN as an untagged port.

Changing the dual-mode default VLAN behavior

The **no dual-mode-default-vlan** command has been added to change this behavior. This is useful in situations where there is a danger of loops being created if Spanning Tree is not or can not be configured on the default VLAN such as when ports are facing a service provider network and STP BPDUs are not welcome on those ports.

Once the **no dual-mode-default-vlan** command is applied at the global level, a port will not be entered into the dual-mode state by default. If the **no dual-mode-default-vlan** command is configured, when a port is added as tagged to a non-default user-defined VLAN, it is automatically removed from the default VLAN and added to the non-default VLAN as a pure tagged port. Once in this state, a port can only be placed in dual-mode by explicitly configuring it as an untagged port into a non-default VLAN.

When the **no untagged ethernet** command is applied under the default VLAN against a port in dual mode, the port will go into pure tagged mode in contrast to the default operating conditions where the port is automatically placed in the dual mode state with regard to the default VLAN. To change the default condition of a device regarding the dual-mode, default VLAN behavior, enter the **no dual-mode-default-vlan** command as shown in the following.

```
device(config)# no dual-mode-default-vlan
```

Syntax: [no] dual-mode-default-vlan

The default state is for ports added as tagged to a non-default VLAN to remain as untagged ports in the default VLAN and become dual-mode ports.

Using this command with the **no** option, changes the default state and automatically removes a port from the default VLAN when it is added as a tagged port to a non-default VLAN. Using the command without the **no** option, will return the systems behavior to its normal operating condition.

NOTE
When a device is operating in the default state regarding the **no dual-mode-default-vlan** command (which is not configured), syslog messages are generated whenever a port is moved out of the default VLAN. This is normal and expected behavior. Because when the **no dual-mode-default-vlan** command is configured, it is normal operating behavior for a port to be moved out of the default VLAN whenever it enters the dual-mode state syslog messages may be generated when the ports are moved, which is not expected. These messages may be generated in the following situations:

- When a port is added "tagged" into the default VLAN, it is automatically deleted from the default VLAN and a syslog message is generated.
- When a port is in dual-mode, and a user issues the **no untagged** command within the port VLAN configuration, the port is added back to the default VLAN. However, because the **no dual-mode-default-vlan** command is configured, the port is transitioned out of the default VLAN which generates an additional syslog message.

Restrictions for use of this command

The **no dual-mode-default-vlan** command can only be applied if the device does not currently have any ports configured in dual-mode. If any port is currently in the dual-mode state when the **no dual-mode-default-vlan** command is executed, the command is rejected without it being applied to any ports on the device. Consequently, using the **no dual-mode-default-vlan** command does not cause any action but enables a new behavior for ports that are added to a VLAN.

If there are ports configured into the dual-mode (default VLAN) state, they can be moved from that state by removing untagged ports the default VLAN that also exist as tagged in a non-default VLAN or removing tagged ports from the non-default VLANs.

Disabling dual-mode for tagged ports

Unless you have a specific need to operate a tagged port in dual-mode, you should make it strictly tagged by removing it from the set of untagged ports in the default VLAN.

If you leave a tagged port in dual-mode as an untagged member of the default VLAN, then any untagged broadcast, multicast, and unknown unicast frames received on that port will be flooded out on all other ports in the default VLAN. This is expected behavior. The **no dual-mode-default-vlan** command has been provided to change this behavior, but this command can only be added if no ports are in dual-mode. If you already have tagged ports and if you do not want them to forward untagged frames, you should remove them from the untagged ports of the default VLAN as shown in this example.

```
device(config)#vlan 2
device(config-vlan-2)#tagged e 1/1 to 1/8
device(config-vlan-2)#vlan 1
device(config-vlan-1)#no untagged e 1/1 to 1/8
```

After you add a port to a VLAN as a tagged member, you should then make it strictly tagged by removing it from the default VLAN as an untagged member.

NOTE
If the device already has tagged ports, Extreme strongly recommends that you disable dual-mode for tagged ports by removing all the tagged ports from the set of untagged ports in the default VLAN. Unless the device already has "no dual-mode-default-vlan" configured, or if you intend to use the default VLAN for Layer2 switching of traffic, or if Layer2 switching in the default VLAN is explicitly required for other functions, or if you have your tagged ports configured for dual-mode with untagged traffic from a non-default VLAN.

Layer 2 control protocols on VLANs

Layer 2 protocols such as STP, RSTP, ERP, Foundry MRP, and VSRP can be enabled on a port-based VLAN.

The Layer 2 state associated with a VLAN and port is determined by the Layer 2 control protocol. Layer 2 broadcasts associated with the VLAN will not be forwarded on this port if the Layer 2 state is not FORWARDING.

It is possible that the control protocol, for example STP, will block one or more ports in a protocol-based VLAN that uses a virtual routing interface to route to other VLANs. For IP protocol and IP subnet VLANs, even though some of the physical ports of the virtual routing interface are blocked, the virtual routing interface can still route as long as at least one port in the virtual routing interface's protocol-based VLAN is not blocked by STP.

You can also enable Single STP (SSTP) on the device; however, the ports in all VLANs on which SSTP is enabled become members of a single spanning tree. The ports in VLANs on which SSTP is disabled are excluded from the single spanning tree. A VLAN can also be selectively added or removed from the single spanning tree domain.

Virtual interfaces and CPU protection co-existence on VLANs

CPU protection can be configured on VLANs regardless of whether there are virtual-interfaces configured on them (Previously, CPU protection was only configurable if a virtual-interface was not configured on the VLAN).

There is a difference in the behavior of CPU protection in each of the following situations:

- When virtual-interfaces are configured on a VLAN, the CPU-protection is done only on unknown-unicast packets from the VLAN. Multicast and broadcast packets from the VLAN will be sent to the CPU. This allows the CPU to process packets such as ARP and OSPF "hello" packets that may be relevant to the device.
- When virtual-interface is not configured on the VLAN, the CPU-protection is performed for all packets (unknown-unicast, multicast and broadcast) from the CPU.

Configuring port-based VLANs

As explained above, you can place ports into VLANs to segment traffic into broadcast domains. When you create a VLAN, you specify if ports added to that VLAN are tagged or untagged.

NOTE
When adding a port to a VLAN you might get an error message concerning IP routing or IPv6 routing information on the port. If you receive this message, check to see if the port was previously configured for routing protocols such as OSPFv2 or OSPFv3 where the routing protocol was removed globally without first being de-configured on that port. If this is the case, re-enable the routing protocol globally to view the interface configuration and then disable the routing protocol from the port. You can then add the port to the VLAN.

To create a VLAN, perform the tasks listed below.

1. At the global CONFIG level assign an ID to the VLAN.

```
device(config)# vlan 2
```

VLAN IDs can be in the range of 1 - 4090. Use the **no** form of the command to delete the VLAN from the configuration.

The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

In addition to a VLAN number, you can assign a name to a VLAN by entering name *vlan-name*. Enter up to 35 characters for name.

2. Once a VLAN ID is assigned, the CLI directs you to the VLAN configuration level. At this level, you add ports to that VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-2)# untag e 1/9 to 1/16
device(config-vlan-2)# tagged e 1/1 to 1/8
```

The example above configures a port-based VLAN, VLAN 2. It adds Ethernet ports 1/9 through 1/16 as untagged ports and ports 1/1 through 1/8 as tagged ports. Since ports 1/9 through 1/16 are untagged, they can be members of VLAN 2 only, while ports 1/1 through 1/8 are tagged ports and can be members of other VLANs.

> NOTE
> In the configuration above, ports 1/9 - 1/16 are automatically removed from the default VLAN since they are configured as untagged ports; while port 1/1 - 1/8 are still members of the default VLAN.

Syntax: [no] untagged tagged | **ethernet slot-number/port-number** [**to slot-number/port-number** | **ethernet slot-number/port-number**]

Ports are removed from the default VLAN only when the port is added as an **untagged** member of a different VLAN. The **untag** command also allows the ports to process packets that do not contain 802.1q tagging. A port is removed from a default-VLAN when the port is added as an untagged member of a different VLAN.

The **tagged** parameter allows the Extreme device to add a four-byte tag 802.1q tag to the packets that go through the tagged ports. It also allows the ports to be members of other VLANs.

Enter the port that you want to assign to the VLAN for the **ethernet***slot-number* /*port-number* parameter. When you add the LAG group's primary port, all the ports on the LAG group become members of the VLAN.

Use the **no** form of the command to remove the ports from a VLAN.

```
device(config)# vlan 4
device(config-vlan-4)# no untag ethernet 1/11
```

Strictly or explicitly tagging a port

If you want a port to be strictly or explicitly tagged, that port has to be removed from the default VLAN. Enter a command such as the following.

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/8
device(config-vlan-2)# vlan 1
device(config-vlan-1)# no untagged e 1/1 to 1/8
```

Assigning or changing a VLAN priority

> NOTE
> When you apply the vlan priority command with running traffic, it may drop packets for a short period of time and flush out the MAC addresses. This is normal behavior.

You can prioritize traffic on a VLAN by assigning a priority to a VLAN. All packets associated with the VLAN will be classified to the configured priority.

```
device(config-vlan-2)# priority 2
```

Syntax: [no] priority num

Possible Values: 0 - 7, "0" assigns the lowest priority and "7," the highest priority. The default is "0."

Assigning a different ID to the default VLAN

As stated above, by default, all ports on a device belong to the default VLAN, which is VLAN 1, until it is assigned to a port-based VLAN. The default VLAN port membership is always untagged; however, if you want to use VLAN ID 1 as a configurable VLANs with tagged port members, you can assign a different VLAN ID as the default VLAN. Enter commands such as the following command.

```
device(config)# default-vlan-id 4000
```

Syntax: [no] default-vlan-id vlan-id

You must specify a VLAN ID that is not already in use. For example, if VLAN 10 exists, do not use "10" as the new VLAN ID for the default VLAN. VLAN IDs are from 1 - 4090. The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

Configuring protocol-based VLANs

Once port-based VLANs are created, you can further segment the broadcast domains by creating protocol-based VLANs, based on Layer 3 protocols. Use the general procedure below for creating protocol-based VLANs.

1. Create the port-based VLAN that contains the interface that you want to segment using Layer 3 protocols.

   ```
   device(config)# vlan 2
   device(config-vlan-2)# untag e 1/9 to 1/16
   device(config-vlan-2)# tagged e 1/1 to 1/8
   ```

2. Under the VLAN configuration level, define the Layer 3 protocol you want to use to segment packets that go through the ports assigned to the port-based VLAN.

   ```
   device(config-vlan-2)# ipv6-proto name Blue
   ```

   Syntax: [no] ip-proto ipv6-proto | | ipx-proto | atalk-proto | other-proto name protocol-vlan-name

   Enter:

   - – **ip-proto** to create a IP protocol VLAN.
     – **ipv6-proto** to create a IPv6 protocol VLAN.
     – **ipx-proto** to create a IPX protocol VLAN.
     – **atalk-proto** to create an Appletalk protocol VLAN.
     – **other-proto** to create a protocol VLAN for protocols other than an IP protocol, IPv6, IPX, or Appletalk protocol.

   Enter **name** *vlan-name* if you want to assign a name to the protocol-based VLAN. Enter up to 35 characters for name.

   Use the **no** form of the command to remove the protocol-based VLAN.

3. Assign or exclude specific ports to the protocol-based VLAN.

   ```
   device(config-vlan-group-ipv6-proto)# static e 1/1 e 1/24
   device(config-vlan-group-ipv6-proto)# exclude e 1/2 to 1/4
   ```

   Syntax: [no] static exclude | ethernet slot-number/port-number [**to slot-number/port-number**]

   The **static** ethernet *slot-number /port-number* [to *slot-number /port-number*] parameter adds the specified ports within the port-based VLAN as static ports to the protocol-based VLAN. Packets of the specified protocol will be forwarded on these ports.

   The **exclude** ethernet *slot-number /port-number* [to *slot-number /port-number*] parameter excludes the specified ports from the protocol-based VLAN. Packets of the specified protocol will be dropped if received on these ports.

Configuring virtual routing interfaces

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

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

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

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/2
device(
config-vlan-2)# router-interface ve 1
```

The Extreme device can locally route IP packets between VLANs that are defined within a single device.

If you do not need to further partition the port-based VLAN into protocol-based VLANs, you can define a single virtual routing interface at the port-based VLAN level and enable routing on a single virtual routing interface.

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/2
device(config-vlan-2)# router-interface ve 2
device(config-vlan-2)# exit
device(config)# interface ve 2
device(config-
ve-2)# ip address 10.1.1.1/24
```

Syntax: router-interface ve ve-number

Enter 1 to the maximum number of virtual routing interfaces supported on the device for *ve-number*.

Integrated Switch Routing

Integrated Switch Routing (ISR) feature enables VLANs configured on the Extreme device to route Layer 3 traffic from one protocol-based VLAN to another instead of forwarding the traffic to an external router. The VLANs provide Layer 3 broadcast domains for the protocols, but do not in themselves provide routing services. This is true even if the source and destination protocols are on the same device.

ISR eliminates the need for an external router by allowing you to route between VLANs using virtual routing interfaces (ves). You configure a separate virtual routing interface on each VLAN that you want to use to route packets. For example, if you configure two IP protocol VLANs on a device, you can configure a virtual routing interface on each of the IP protocol VLAN, then configure IP routing parameters for the IP protocol VLAN. Thus, the Extreme device forwards IP broadcasts within each VLAN at Layer 2 but routes Layer 3 traffic between the VLANs using the virtual routing interfaces.

> NOTE
> The Extreme device uses the lowest MAC address on the device (the MAC address of port 1/1) as the MAC address for all ports within all virtual routing interfaces you configure on the device.

The routing parameters and the syntax for configuring them are the same as when you configure a physical interface for routing (for example, **interface ve 10**). The logical interface allows the Extreme device to internally route traffic between the protocol-based VLANs without using physical interfaces.

All the ports within a protocol-based VLAN must be in the same port-based VLAN. The protocol-based VLAN cannot have ports in multiple port-based VLANs, unless the ports in the port-based VLAN to which you add the protocol-based VLAN are 802.1q tagged.

You can configure multiple protocol-based VLANs within the same port-based VLAN. In addition, a port within a port-based VLAN can belong to multiple protocol-based VLANs of the same type or different types. For example, if you have a port-based VLAN that contains ports 1/1 - 1/10, you can configure port 1/5 as a member of an AppleTalk protocol VLAN, an IP protocol VLAN, and an IPX protocol VLAN, and so on.

If the router interface for IP is configured on physical ports, then routing occurs independent of the Spanning Tree Protocol (STP). However, if the router interfaces are defined for IP VLAN, they are virtual routing interfaces and are subject to the rules of STP.

If your backbone consists of virtual routing interfaces all within the same STP domain, it is a bridged backbone, not a routed one. This means that the set of backbone interfaces that are blocked by STP will be blocked for routed protocols as well. The routed protocols will be able to cross these paths only when the STP state of the link is FORWARDING. This problem is easily avoided by proper network design.

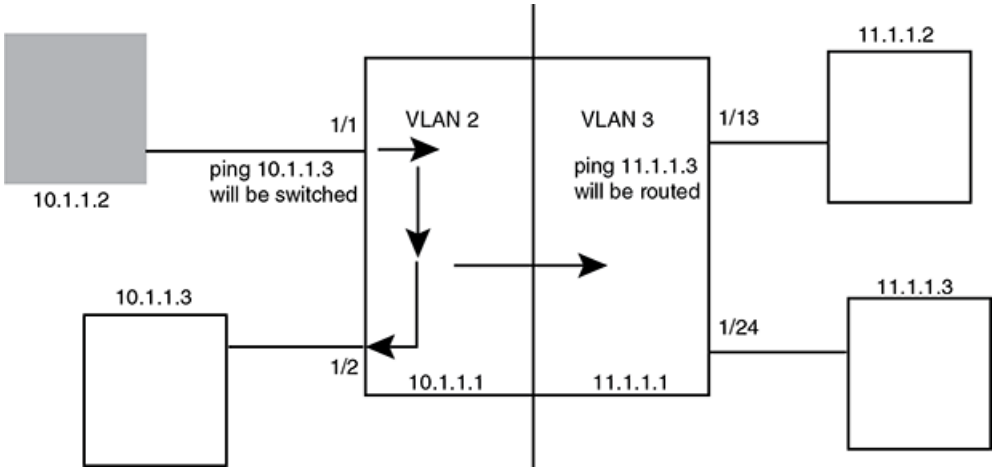
When designing an ISR network, pay attention to your use of virtual routing interfaces and the spanning-tree domain. If Layer 2 switching of your routed protocols (IP, IPX, AppleTalk) is not required across the backbone, then the use of virtual routing interfaces can be limited to edge switch ports within each router. Full backbone routing can be achieved by configuring routing on each physical interface that connects to the backbone. Routing is independent of STP when configured on a physical interface.

If your ISR design requires that you switch IP, IPX, or Appletalk at Layer 2 while simultaneously routing the IP protocol over a single backbone, then create multiple port-based VLANs and use VLAN tagging on the backbone links to separate your Layer 2 switched and Layer 3 routed networks.

There is a separate STP domain for each port-based VLAN. Routing occurs independently across port-based VLANs or STP domains. You can define each end of each backbone link as a separate tagged port-based VLAN. Routing will occur independently across the port-based VLANs. Because each port-based VLAN's STP domain is a single point-to-point backbone connection, you are guaranteed to never have an STP loop. STP will never block the virtual router interfaces within the tagged port-based VLAN, and you will have a fully routed backbone.

The Extreme device offers the ability to create a virtual routing interface within a Layer 2 STP port-based VLAN or within each IP protocol VLAN. This combination of multiple Layer 2 or Layer 3 broadcast domains and virtual routing interfaces are the basis for the very powerful Integrated Switch Routing (ISR) technology. ISR is very flexible and can solve many networking problems.

FIGURE 8 Example of two separate backbones for the same protocol



The following is a sample configuration for the illustration above.

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/2
device(config-vlan-2)# router-inter ve 2
device(config-vlan-2)# exit
device(config)# vlan 3
device(config-vlan-3)# tagged e 1/13 to 1/24
device(config-vlan-3)# router-int ve 3
device(config-vlan-3)# exit
device(config)# interface ve 2
device(config-
ve-2)# ip address 10.1.1.1/24
device(config-if-e1000-2/1)# exit
device(config)# interface ve 3
device(config-
ve-3)# ip address 10.2.1.1/24
```

IP packets are bridged (switched) within the same protocol VLAN if they are on the same subnet; they are routed if they are on a different VLAN.

VLAN groups

To simplify VLAN configuration when you have many VLANs with the same configuration, you can configure *VLAN groups* . When you create a VLAN group, the VLAN parameters you configure for the group apply to all the VLANs within the group.

The VLAN group feature allows you to create multiple port-based VLANs with identical port members. Since the member ports are shared by all the VLANs within the group, you must add the ports as tagged ports. This feature not only simplifies VLAN configuration but also allows you to have a large number of identically configured VLANs in a startup configuration file on the device's flash memory module. Normally, a startup configuration file with a large number of VLANs might not fit on the flash memory module. By grouping the identically configured VLANs, you can conserve space in the startup configuration file so that it fits on the flash memory module.

On the Extreme devices, you can create up to 128 VLAN groups per system.

> NOTE
> Depending on the size of the VLAN ID range you want to use for the VLAN group, you might need to allocate additional memory for VLANs. To allocate additional memory, refer to Allocating memory for more VLANs or virtual routing interfaces on page 104.

Configuring a VLAN group

To configure a VLAN group, perform the tasks listed below.

1. Create the VLAN group and assign the VLANs to that group.

```
device(config)# vlan-group 1 vlan 2 to 1000
```

Syntax: [no] vlan-group num vlan vlan-id to vlan-id

The *num* parameter specifies the VLAN group ID. On the Extreme devices, you can create up to 128 VLAN groups per system.

The **vlan***vlan-id***to***vlan-id* parameters specify a continuous range (with no gaps) of VLAN IDs that have not been configured in the CLI. Specify the low VLAN ID first and the high VLAN ID second. The command adds all the VLANs in the range to the VLAN group.

If a VLAN within the range you specify is already configured, the CLI does not add the group but instead displays an error message. If this happens, create the group by specifying a valid contiguous range that does not include the VLAN. Then add more VLANs to the group after the CLI changes to the configuration level for the group.

> NOTE
> The device's memory must be configured to contain at least the number of VLANs you specify for the higher end of the range. For example, if you specify 2048 as the VLAN ID at the high end of the range, you first must increase the memory allocation for VLANs to 2048 or higher. Refer to Allocating memory for more VLANs or virtual routing interfaces on page 104.

2. The CLI directs you to the VLAN group configuration level. Add tagged ports to the group. Since all the VLANs in the group share the ports, you must add the ports as tagged ports.

```
device(config-vlan-group-1)# tagged e 1/1 to 1/2
```

Syntax: [no] tagged ethernet [**to slot-number/port-number** | **ethernet slot-number/port-number**]

Using the **no tagged ethernet** command causes the following error message such as the following to appear.

```
device(config-vlan-10)#no tagged ethernet 4/2
error - ports ether 4/1 to 4/2 are not tagged members of vlan 10
```

This message is normal and indicates that the configuration has take effect. It does not indicate that an error condition has occurred.

3. If required, you can add and remove individual VLANs or VLAN ranges from the VLAN group configuration level. For example, to add VLANs 1001 and 1002 to VLAN group 1 and remove VLANs 900 through 1000, enter the following commands.

```
device(config-vlan-group-1)# add-vlan 1001 to 1002
device(config-vlan-group-1)# remove-vlan 900 to 1000
```

Syntax: [no] add-vlan vlan-id [**to vlan-id**]

Syntax: [no] remove-vlan vlan-id [**to vlan-id**]

Verifying VLAN group configuration

To verify configuration of VLAN groups, display the running configuration file. If you have saved the configuration to the startup configuration file, you also can verify the configuration by displaying the startup configuration file. The following example shows the running configuration information for the VLAN group configured in the previous examples. The information appears in the same way in the startup configuration file.

```
device(config)# show running-config
```

lines not related to the VLAN group omitted...

```
vlan-group 1 vlan 2 to 900
add-vlan 1001 to 1002
tagged ethernet 1/1 to 1/2
```

Displaying information about VLAN groups

To display VLAN group configuration information, enter the following command.

```
device# show vlan-group 10
Configured VLAN-Group entries : 1
Maximum VLAN-Group entries : 32
VLAN-GROUP 10
Number of VLANs: 4
VLANs: 10 to 13
Tagged ports: ethernet 3/1
```

The example shows configuration information for two VLAN groups, group 1 and group 2.

Syntax: show vlan-group [**group-id**]

The *group-id* specifies a VLAN group. If you do not use this parameter, the configuration information for all the configured VLAN groups is displayed.

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:

- STP
- MRP
- VSRP
- RSTP
- Ethernet RIng Protection (ERP)

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 Foundry MRP, the topology group's master VLAN contains the ring 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. VPLS VLANs can become member VLANs within a topology group.
- Member VLAN groups—A VLAN group is a named set of VLANs. The VLANs within a VLAN group have the same ports and use the same values for other VLAN parameters.

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.

Master VLANs and customer VLANs in Foundry MRP

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as Foundry MRP. For more information on topology group and Foundry MRP, refer to the *VLANs* chapter.

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 255 topology groups. Each group can control up to 4000 VLANs. A VLAN cannot be controlled by more than one topology group.
- After you add a VLAN as a member of a topology group, the device deletes all the Layer 2 protocol information on that VLAN.
- If you add a new master VLAN to a topology group that already has a master VLAN, the new master VLAN replaces the older master VLAN. All member VLANs and VLAN groups follow the Layer 2 protocol settings of the new master VLAN.
- If you remove the master VLAN (by entering the **no master-vlan** command with the *vlan-id* variable), the software selects the new master VLAN from member VLANs. A new candidate master VLAN will be in configured order to a member VLAN so that the first added member VLAN will be a new candidate master VLAN. Once you save and reload, a member VLAN with the youngest VLAN ID will be a new candidate master. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.
- After you add a VLAN or VLAN group as a member of a topology group, all the Layer 2 protocol configuration information for the VLAN or group is deleted. For example, if STP is configured on a VLAN and you add the VLAN to a topology group, the

STP configuration is removed from the VLAN. After you add the VLAN to a topology group, the VLAN uses the Layer 2 protocol settings of the master VLAN. If you remove a member VLAN or VLAN group from a topology group, you need to reconfigure the Layer 2 protocol information in the VLAN or VLAN group.

- On platforms where the Ethernet Service Instance (ESI) framework is supported, master VLANs in a topology group must either be in the default ESI or within the same ESI. Master and member VLANs cannot span multiple ESIs.

Configuring a topology group

To configure a topology group, enter commands such as the following.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan 2
device(config-topo-group-2)# member-vlan 3
device(config-topo-group-2)# member-vlan 4
device(config-topo-group-2)# member-vlan 5
device(config-topo-group-2)# member-group 2
```

The commands configure topology group 2 and add the following to it:

- VLAN 2 as master VLAN
- VLANs 3, 4, and 5 as member VLANs
- Member VLAN group 2

Syntax: [no] topology-group group-id

The **topology-group** command creates a topology group. The *group-id* parameter assigns an ID 1 to 255 to the topology group.

Syntax: [no] master-vlan vlan-id

This command adds the master VLAN to the topology group. The VLAN must already be configured. Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

NOTE
When a port is added to a master VLAN, it will be added as a free port. Similarly when a port has to be removed from master VLAN, first disable any the Layer 2 protocol on the port, then remove the port from the master VLAN.

Syntax: [no] member-vlan vlan-id

This command adds a member VLAN to the topology group. The VLAN must already be configured.

Syntax: [no] member-group num

This command adds a VLAN group to the topology group. The *num* specifies a VLAN group ID. The VLAN group must already be configured.

Adding VPLS VLANs to topology groups

To add *single-tagged or untagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following example.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan 2
device(config-topo-group-2)# member-vlan vpls id 34 vlan 42 to 45
```

To add *dual-tagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following configuration example.

```
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 10
device(config-topo-group-1)# member-vlan 20
device(config-topo-group-1)# member-vlan vpls id 5 vlan 300 inner-vlan 20 to 25
```

Syntax: [no] member-vlan vpls [**id vpls-id** | **name vpls-name** } **vlan vlan-id** [**to vlan-id**]

OR

Syntax: [no] member-vlan vpls [**id** *vpls-id* | **name** *vpls-name*] **vlan** *vlan-id* [**inner-vlan** *inner-vlan-id* [**to** *inner-vlan-id*]]

The **id** option allows you to specify the VPLS instance that you are configuring into the topology group by using the VPLS ID of the instance. A value in the range of 1 - 4294967294 can be entered for VPLS ID.

The **name** option allows you to specify the VPLS instance that you are configuring into the topology group by using the name of the instance.

The *vlan-id* variable is used with the **vlan** keyword to specify the VPLS VLAN being configured into topology group. You can specify multiple *vlan-id* values or specify a range of VLANs using the **to** option.

The **inner-vlan** option allows you to specify a VPLS dual-tagged (double-tagged) VLAN configuration.

NOTE
The **inner-vlan** option does not allow both outer VLAN ranges and inner VLAN ranges for a given VPLS instance. Once an outer VLAN range is specified, the inner VLAN option is not allowed. However, if a single outer VLAN is specified, the inner VLAN option and range is allowed.

NOTE
You cannot delete a topology master VLAN if the topology group has only VPLS VLAN members and no Layer 2 VLAN members because the normal procedure for deleting a topology master VLAN is to elect another Layer 2 VLAN as the new master. Because a VPLS VLAN cannot be a master VLAN, you must have at least one Layer 2 VLAN as a member. If it does not currently exist, you must add a Layer 2 VLAN before deleting a topology master.

NOTE
A maximum of 4000 VPLS member VLANs can be added to a topology group.

Topology group support within an ESI

Topology groups can be configured with VLANs that are part of a user-defined ESI. (Consult Topology Groups on page 86 to see which platform supports topology groups within an ESI.) When you configure topology groups in such a scenario, both the master and member VLANs must be part of the same ESI. If an ESI is not specified, the system assumes a reference to the default ESI. Below is an example of configuring topology groups with VLANs that are part of a user-defined ESI.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan service-esi 2
device(config-topo-group-2)# member-vlan service-esi 3
device(config-topo-group-2)# member-vlan service-esi 4
device(config-topo-group-2)# member-vlan service-esi 5
device(config-topo-group-2)# member-group service-esi 2
```

The commands configure topology group 2 and add the following to it:

- VLAN 2 in ESI "service-esi" as master VLAN
- VLANs 3, 4, and 5 in ESI "service-ESI" as member VLANs
- Member VLAN group 2

Syntax: [no] topology-group group-id

This command creates a topology group. The *group-id* parameter assigns an ID in the range 1 to 255 to the topology group.

Syntax: [no] master-vlan esi-name vlan-id

This command adds the master VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name". Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

Syntax: [no] member-vlan esi-name vlan-id

This command adds a member VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name".

Syntax: [no] member-group esi-name num

This command adds a VLAN group in ESI identified by "esi-name" to the topology group. The *num* specifies a VLAN group ID. The VLAN group must already be configured.

Displaying topology group information

This section contains examples of the **show topology-group** command output. Support for topology groups within an ESI is supported on a minority of platforms (listed in Displaying topology group information on a CES 2000 Series device on page 92), so its example appears at the end of this section.

Displaying topology group information on an XMR Series or MLX Series device

The **show topology-group** command offers a choice between one of two mandatory parameters. The command syntax (on an XMR Series or MLX Series device) is as follows.

Syntax: show topology-group *group-id* | **hw-index-table** [*hw-index*]

The first example in this section utilizes the first possible mandatory parameter, *group-id*. The second example utilizes the second possible mandatory parameter, **hw-index-table**, along with an optional variable, a hardware index number.

Display topology group information by using a Group ID

To display topology group information for group 10, enter the **show topology-group** command.

```
device#show topology-group 10
Topology Group 10
==================
Topo HW Index   : 0
Master VLAN     : 10
VPLS VLAN exist : TRUE
Member VLAN     : 20
Member Group  : None
Control Ports : ethe 3/11 to 3/12 ethe 3/15 to 3/16
Free Ports :
```

Syntax: show topology-group group-id

This display shows the following information:

TABLE 7 CLI display of topology group information

This field...	Displays...
Topology Group	The ID of the topology group. The range for *group-id* is 1 - 256.

TABLE 7 CLI display of topology group information (continued)

This field...	Displays...
Topo HW Index	A topology hardware index is a unique hardware ID that is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The range for *hw-index* is 0 - 511. (The **show topology-group hw-index-table** command output shows the mapping of a topology hardware index to a VLAN.)
Master-VLAN	The master VLAN for the topology group. The settings for STP, Foundry MRP, ERP, RSTP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
VPLS VLAN exist	Indicates whether the topology group has one or more VPLS VLANs as a topology group member. The content of this field is TRUE or FALSE.
Member-VLAN	The VLAN ID of the member of the topology group.
Control ports	The master VLAN ports that are configured with Layer 2 protocol information. The Layer 2 protocol configuration and state of these ports in the master VLAN applies to the same port numbers in all the member VLANs.
Free ports	A list of all free ports in the topology group. A free port is not controlled by the Layer 2 protocol information in the master VLAN. In the example screen output, the absence of any number indicates that no ports are free.

Display topology group information by using hardware index table numbers

Display the information for hardware index table 0.

```
device#show topology-group hw-index-table  0
Total Instances : 512
Free Instances  : 511
Topo HW Index    Vlan ID
-------------    -------
0                10
```

Syntax: show topology-group hw-index-table [**hw-index**]

The range for *hw-index* is 0 - 511. If you do not specify a number for *hw-index*, the output screen lists all entries.

TABLE 8 Topology group information with hardware index table

This field...	Displays...
Total Instances	Total number of topology hardware indexes that have been initialized in the system.
Free Instances	Number of free topology hardware indexes that are left in the system.
Topology HW Index	A topology hardware index is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The **show topology-group hw-index-table** command output shows the mapping of a topology hardware index to a VLAN. The range for is 0 - 511. In the example, hardware index table 0 is mapped to the VLAN with an ID of 10.
VLAN ID	The ID of the port-based VLAN that owns the protocol instance on that VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on a device, all protocol information is for VLAN 1.

Displaying topology group information on a CES 2000 Series device

To display topology group information within an ESI, enter the **show topology-group** command, as in the following example.

```
device(config)# show topology-group 3
Topology Group 3
================
 master-vlan 2
 member-vlan none
 Common control ports          L2 protocol
 ethernet 1/1                  MRP
 ethernet 1/2                  MRP
 ethernet 1/5                  VSRP
 ethernet 2/22                 VSRP
 Per vlan free ports
 ethernet 2/3                   Vlan 2
 ethernet 2/4                   Vlan 2
 ethernet 2/11                  Vlan 2
 ethernet 2/12                  Vlan 2
```

Syntax: show topology-group group-id

This display shows the following information.

TABLE 9 CLI display of topology group information

This field...	Displays...
master-vlan	The master VLAN for the topology group. The settings for STP, Foundry MRP, RSTP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
member-vlan	The member VLANs in the topology group.
Common control ports	The master VLAN ports that are configured with Layer 2 protocol information. The Layer 2 protocol configuration and state of these ports in the master VLAN applies to the same port numbers in all the member VLANs.
L2 protocol	The Layer 2 protocol configured on the control ports. The Layer 2 protocol can be one of the following: • Foundry MRP • STP • RSTP • VSRP • ERP
Per vlan free ports	The ports that are not controlled by the Layer 2 protocol information in the master VLAN.

Configuring super aggregated VLANs

A super aggregated VLAN allows multiple VLANs to be placed within another VLAN. This feature allows you to construct Layer 2 paths and channels. A path contains multiple channels, each of which is a dedicated circuit between two end points. The two devices at the end points of the channel appear to each other to be directly attached. The network that connects them is transparent to the two devices.

You can aggregate up to 4090 VLANs within another VLAN. This provides a total VLAN capacity on one Extreme device of 16,728,100 channels (4090 * 4090).

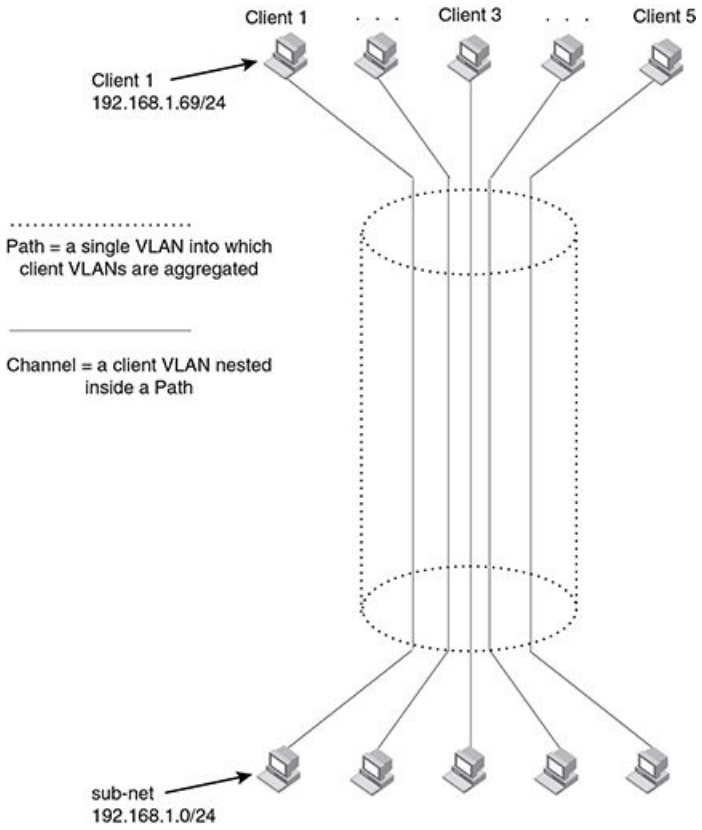
The devices connected through the channel are not visible to devices in other channels. Therefore, each client has a private link to the other side of the channel.

Super aggregated VLANs are useful for applications such as Virtual Private Network (VPN) or Transparent LAN Services (TLS) in which you need to provide a private, dedicated Ethernet connection to individual clients to transparently reach its subnet across multiple networks. The feature allows point-to-point and point-to-multipoint connections.

Figure 9 shows a conceptual picture of the service that aggregated VLANs provide.

In Super Aggregated VLANs, the outer VLAN (path) and the inner VLAN (channel) use different tag types. For example, the outer VLAN tag-type can be 9100 and the inner VLAN tag-type can be 8100 as shown in Figure 9.

FIGURE 9 Conceptual model of the super aggregated VLAN application



Each client connected to the edge device is in its own port-based VLAN. All the clients' VLANs are aggregated by the edge device into a single VLAN for connection to the core.

The device that aggregates the VLANs forwards the aggregated VLAN traffic through the core. The core can consist of multiple devices that forward the aggregated VLAN traffic. The edge device at the other end of the core separates the aggregated VLANs into the individual client VLANs before forwarding the traffic. The edge devices forward the individual client traffic to the clients. For the clients' perspective, the channel is a direct point-to-point link.

Figure 10 shows an example application that uses aggregated VLANs. This configuration includes the client connections shown in Figure 9.

FIGURE 10 Example super aggregated VLAN application

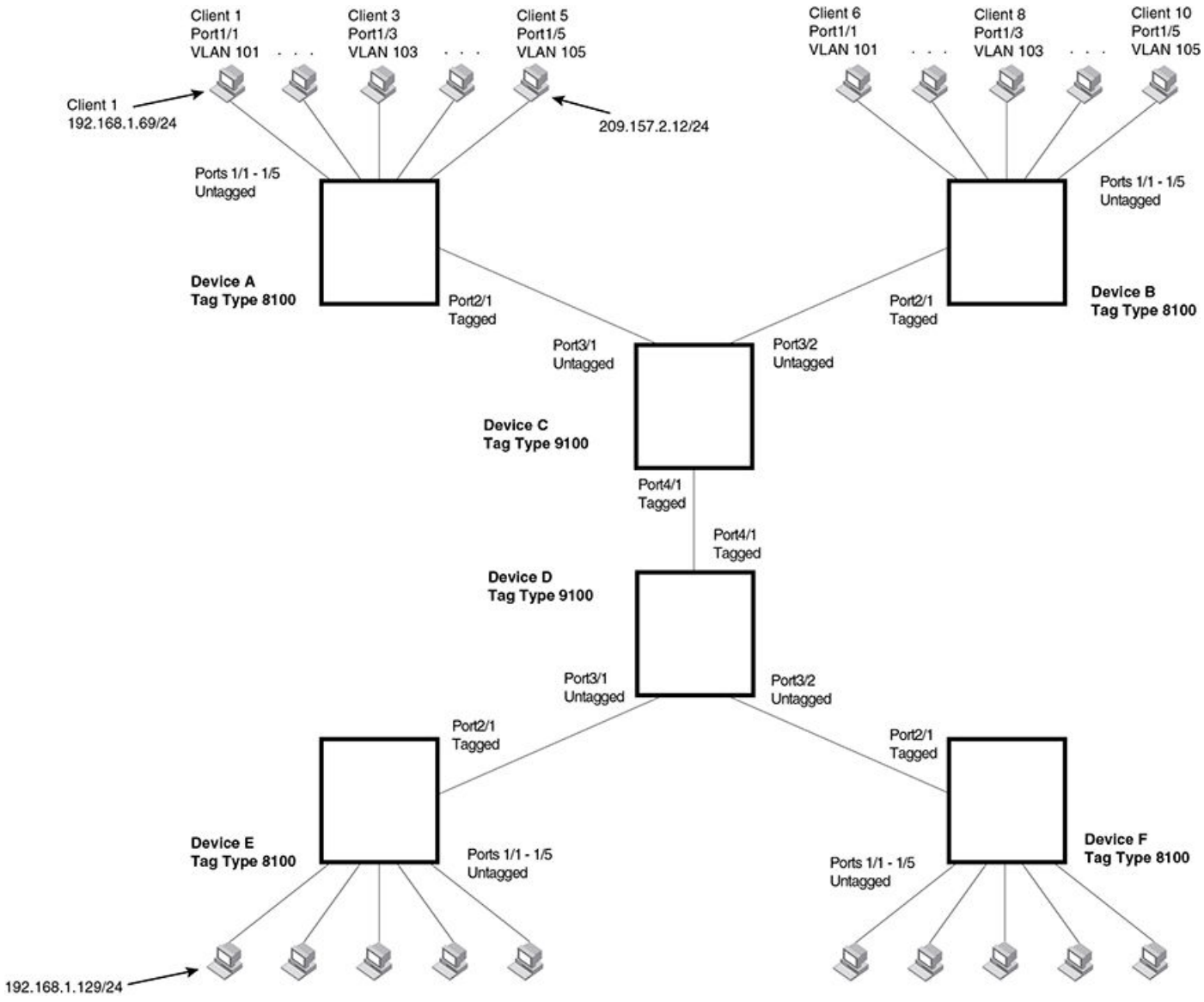


Figure 10 shows a collocation service provides private channels for multiple clients. Although the same devices are used for all the clients, the VLANs ensure that each client receives its own Layer 2 broadcast domain, separate from the broadcast domains of other clients. For example, client 1 cannot ping client 5.

The clients at each end of a channel appear to each other to be directly connected and thus can be on the same subnet and use network services that require connection to the same subnet. In this example, client 1 is in subnet 192.168.1.0/24 and so is the device at the other end of client 1's channel.

Since each VLAN configured on the core devices is an aggregate of multiple client VLANs, the aggregated VLANs greatly increase the number of clients a core device can accommodate.

This example shows a single link between the core devices. However, you can use a LAG group to add link-level redundancy.

Configuring aggregated VLANs

A maximum of 1526 bytes are supported on ports where super-aggregated VLANs are configured. This allows for an additional 8 bytes over the untagged port maximum to allow for support of two VLAN tags.

To configure aggregated VLANs, configure tagged and untagged VLANs on the edge device, then configure the aggregated and other VLANs on the core device. Perform the tasks listed below.

1. On each edge device, configure a separate port-based VLAN for each client connected to the edge device. In each client VLAN:
   - – Add the port connected to the client as an untagged port.
     – Add the port connected to the core device (the device that will aggregate the VLANs) as a tagged port. This port must be tagged because all the client VLANs share the port as an uplink to the core device.

   For example, to configure device A in Configuring super aggregated VLANs on page 92, enter commands such as the following.

```
device(config)# vlan 101
device(config-vlan-101)# tagged ethernet 2/1
device(config-vlan-101)# untagged ethernet 1/1
device(config-vlan-101)# exit
device(config)# vlan 102
device(config-vlan-102)# tagged ethernet 2/1
device(config-vlan-102)# untagged ethernet 1/2
device(config-vlan-102)# exit
device(config)# vlan 103
device(config-vlan-103)# tagged ethernet 2/1
device(config-vlan-103)# untagged ethernet 1/3
device(config-vlan-103)# exit
device(config)# vlan 104
device(config-vlan-104)# tagged ethernet 2/1
device(config-vlan-104)# untagged ethernet 1/4
device(config-vlan-104)# exit
device(config)# vlan 105
device(config-vlan-105)# tagged ethernet 2/1
device(config-vlan-105)# untagged ethernet 1/5
device(config-vlan-105)# exit
device(config)# write memory
```

   Syntax: [no] vlan vlan-id

   Syntax: [no] untagged tagged | **ethernet slot-number/port-number** [**to slot-number/port-number** | **ethernet slot-number/port-number**]

   The **tagged** command adds the port that the device uses for the uplink to the core device.

   The **untagged** command adds the ports connected to the individual clients.

2. On each core device:
   - Configure a VLAN tag type (tag ID) that is different than the tag type used on the edge devices. If you use the default tag type (8100) on the edge devices, set the tag type on the core devices to another value, such as 9100. The tag type must be the same on all the core devices. The edge devices also must have the same tag type but the type must be different from the tag type on the core devices.

     NOTE
     You can enable the Spanning Tree Protocol (STP) on the edge devices or the core devices, but not both. If you enable STP on the edge devices and the core devices, STP will prevent client traffic from travelling through the core to the other side.

   For example, to configure the aggregated VLANs on device C in Configuring super aggregated VLANs on page 92, enter the following commands.

```
device(config)# tag-type 9100
device(config)# vlan 101
device(config-vlan-101)# tagged ethernet 4/1
device(config-vlan-101)# untagged ethernet 3/1
device(config-vlan-101)# exit
device(config)# vlan 102
device(config-vlan-102)# tagged ethernet 4/1
device(config-vlan-102)# untagged ethernet 3/2
device(config-vlan-102)# exit
device(config)# write memory
```

Syntax: [no] tag-type num [**ethernet slot/port**]

The *num* variable is the hexadecimal ethernet tag type. Default value is 8100.

Complete CLI examples

The following sections show all the Aggregated VLAN configuration commands on the devices in Configuring super aggregated VLANs on page 92.

NOTE
In these examples, the configurations of the edge devices (A, B, E, and F) are identical. The configurations of the core devices (C and D) also are identical. The aggregated VLAN configurations of the edge and core devices on one side must be symmetrical (in fact, a mirror image) to the configurations of the devices on the other side. For simplicity, the example in Configuring super aggregated VLANs on page 92 is symmetrical in terms of the port numbers. This allows the configurations for both sides of the link to be the same. If your configuration does not use symmetrically arranged port numbers, the configurations should not be identical but must use the correct port numbers.

Commands for device A

```
device-A(config)# vlan 101
device-A(config-vlan-101)# tagged ethernet 2/1
device-A(config-vlan-101)# untagged ethernet 1/1
device-A(config-vlan-101)# exit
device-A(config)# vlan 102
device-A(config-vlan-102)# tagged ethernet 2/1
device-A(config-vlan-102)# untagged ethernet 1/2
device-A(config-vlan-102)# exit
device-A(config)# vlan 103
device-A(config-vlan-103)# tagged ethernet 2/1
device-A(config-vlan-103)# untagged ethernet 1/3
device-A(config-vlan-103)# exit
device-A(config)# vlan 104
device-A(config-vlan-104)# tagged ethernet 2/1
```

```
device-A(config-vlan-104)# untagged ethernet 1/4
device-A(config-vlan-104)# exit
device-A(config)# vlan 105
device-A(config-vlan-105)# tagged ethernet 2/1
device-A(config-vlan-105)# untagged ethernet 1/5
device-A(config-vlan-105)# exit
device-A(config)# write memory
```

Commands for device B

The commands for configuring device B are identical to the commands for configuring device A. Notice that you can use the same channel VLAN numbers on each device. The devices that aggregate the VLANs into a path can distinguish between the identically named channel VLANs based on the ID of the path VLAN.

```
device-B(config)# vlan 101
device-B(config-vlan-101)# tagged ethernet 2/1
device-B(config-vlan-101)# untagged ethernet 1/1
device-B(config-vlan-101)# exit
device-B(config)# vlan 102
device-B(config-vlan-102)# tagged ethernet 2/1
device-B(config-vlan-102)# untagged ethernet 1/2
device-B(config-vlan-102)# exit
device-B(config)# vlan 103
device-B(config-vlan-103)# tagged ethernet 2/1
device-B(config-vlan-103)# untagged ethernet 1/3
device-B(config-vlan-103)# exit
device-B(config)# vlan 104
device-B(config-vlan-104)# tagged ethernet 2/1
device-B(config-vlan-104)# untagged ethernet 1/4
device-B(config-vlan-104)# exit
device-B(config)# vlan 105
device-B(config-vlan-105)# tagged ethernet 2/1
device-B(config-vlan-105)# untagged ethernet 1/5
device-B(config-vlan-105)# exit
device-B(config)# write memory
```

Commands for device C

Since device C is aggregating channel VLANs from devices A and B into a single path, you need to change the tag type and enable VLAN aggregation.

```
device-C(config)# tag-type 9100
device-C(config)# vlan 101
device-C(config-vlan-101)# tagged ethernet 4/1
device-C(config-vlan-101)# untagged ethernet 3/1
device-C(config-vlan-101)# exit
device-C(config)# vlan 102
device-C(config-vlan-102)# tagged ethernet 4/1
device-C(config-vlan-102)# untagged ethernet 3/2
device-C(config-vlan-102)# exit
device-C(config)# write memory
```

Commands for device D

Device D is at the other end of path and separates the channels back into individual VLANs. The tag type must be the same as tag type configured on the other core device (Device C). In addition, VLAN aggregation also must be enabled.

```
device-D(config)# tag-type 9100
device-D(config)# vlan 101
device-D(config-vlan-101)# tagged ethernet 4/1
device-D(config-vlan-101)# untagged ethernet 3/1
device-D(config-vlan-101)# exit
device-D(config)# vlan 102
device-D(config-vlan-102)# tagged ethernet 4/1
```

```
device-D(config-vlan-102)# untagged ethernet 3/2
device-D(config-vlan-102)# exit
device-D(config)# write memory
```

Commands for device E

Since the configuration in Configuring super aggregated VLANs on page 92 is symmetrical, the commands for configuring device E are identical to the commands for configuring device A.

```
device-E(config)# vlan 101
device-E(config-vlan-101)# tagged ethernet 2/1
device-E(config-vlan-101)# untagged ethernet 1/1
device-E(config-vlan-101)# exit
device-E(config)# vlan 102
device-E(config-vlan-102)# tagged ethernet 2/1
device-E(config-vlan-102)# untagged ethernet 1/2
device-E(config-vlan-102)# exit
device-E(config)# vlan 103
device-E(config-vlan-103)# tagged ethernet 2/1
device-E(config-vlan-103)# untagged ethernet 1/3
device-E(config-vlan-103)# exit
device-E(config)# vlan 104
device-E(config-vlan-104)# tagged ethernet 2/1
device-E(config-vlan-104)# untagged ethernet 1/4
device-E(config-vlan-104)# exit
device-E(config)# vlan 105
device-E(config-vlan-105)# tagged ethernet 2/1
device-E(config-vlan-105)# untagged ethernet 1/5
device-E(config-vlan-105)# exit
device-E(config)# write memory
```

Commands for device F

The commands for configuring device F are identical to the commands for configuring device E. In this example, since the port numbers on each side of the configuration in Configuring super aggregated VLANs on page 92 are symmetrical, the configuration of device F is also identical to the configuration of device A and device B.

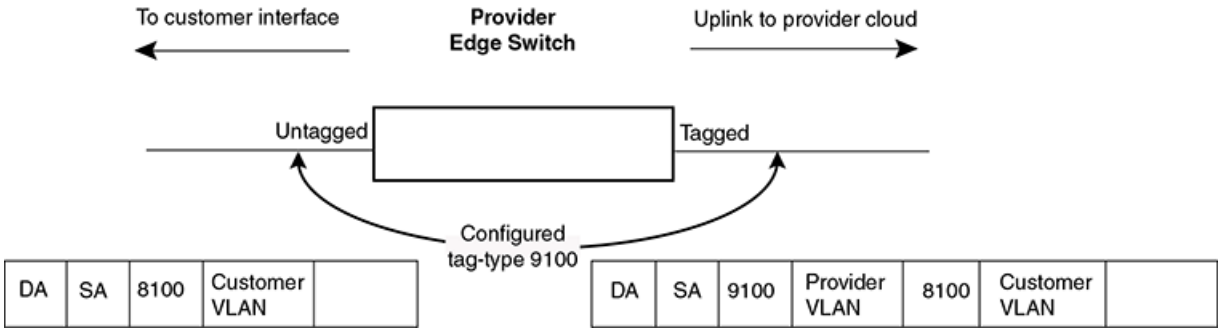
```
device-F(config)# vlan 101
device-F(config-vlan-101)# tagged ethernet 2/1
device-F(config-vlan-101)# untagged ethernet 1/1
device-F(config-vlan-101)# exit
device-F(config)# vlan 102
device-F(config-vlan-102)# tagged ethernet 2/1
device-F(config-vlan-102)# untagged ethernet 1/2
device-F(config-vlan-102)# exit
device-F(config)# vlan 103
device-F(config-vlan-103)# tagged ethernet 2/1
device-F(config-vlan-103)# untagged ethernet 1/3
device-F(config-vlan-103)# exit
device-F(config)# vlan 104
device-F(config-vlan-104)# tagged ethernet 2/1
device-F(config-vlan-104)# untagged ethernet 1/4
device-F(config-vlan-104)# exit
device-F(config)# vlan 105
device-F(config-vlan-105)# tagged ethernet 2/1
device-F(config-vlan-105)# untagged ethernet 1/5
device-F(config-vlan-105)# exit
device-F(config)# write memory
```

Configuring 802.1q-in-q tagging

802.1Q-in-Q tagging enables you to configure 802.1Q tag-types on a group of ports, such as LAG ports, thereby enabling the creation of two identical 802.1Q tags (802.1Q-in-Q tagging) on a single device. This feature improves SAV interoperability between Extreme devices and other vendors' devices that support the 802.1Q tag-types, but are not very flexible with the tag-types they accept.

Figure 11 shows an 802.1Q configuration example.

FIGURE 11 SAV configuration example

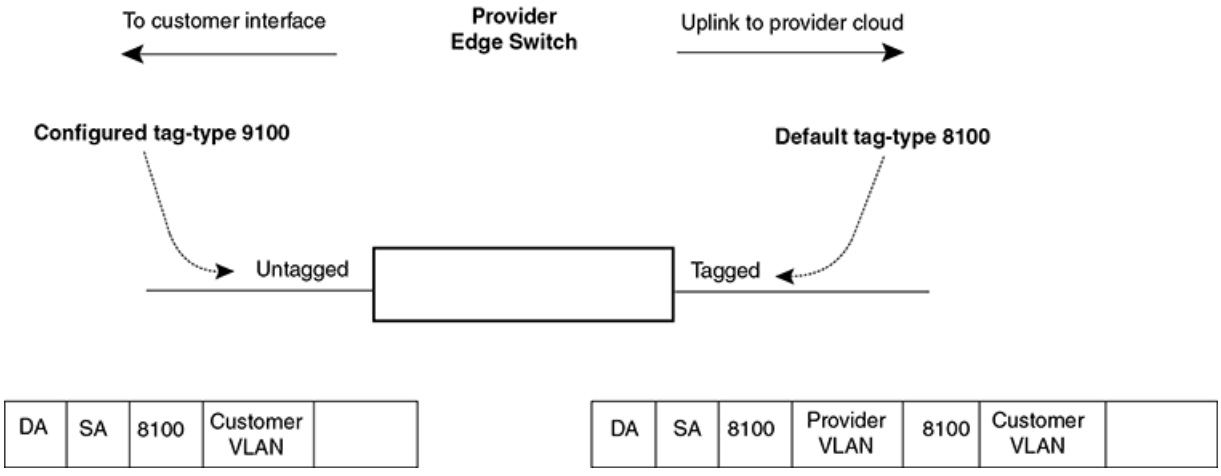


As shown in Figure 11, the ports to customer interfaces are untagged, whereas the uplink ports to the provider cloud are tagged, because multiple client VLANs share the uplink to the provider cloud. In this example, the Extreme device treats the customer's private VLAN ID and 8100 tag type as normal payload, and adds the 9100 tag type to the packet when the packet is sent to the uplink and forwarded along the provider cloud.

As long as the switches in the provider's network support the 9100 tag type, the data gets switched along the network. However, devices that do not support the 9100 tag type may not properly handle the packets.

Figure 12 and Example configuration on page 100 show an example application of 802.1Q-in-Q.

FIGURE 12 802.1Q-in-Q configuration example



In Configuring 802.1q tag-type translation on page 101, the untagged ports (to customer interfaces) accept frames that have any 802.1Q tag other than the configured tag-type 9100. These packets are considered untagged on this incoming port and are re-tagged

when they are sent out of the uplink towards the provider. The 802.1Q tag-type on the uplink port is 8100, so the Extreme device will switch the frames to the uplink device with an additional 8100 tag, thereby supporting devices that only support this method of VLAN tagging.

Configuration rules

Follow the rules below when configuring 802.1q-in-q tagging:

- The Extreme device supports per port tag-type configuration. Consequently, each port can have its own tag-type setting.
- The default tag-type for a port is 8100.
- The Extreme device supports 802.1q-in-q tagging where the inner and outer tag can have different or same tag-type values. This feature maximizes interoperability with third-party devices.

Enabling 802.1Q-in-Q tagging

To enable the 802.1Q-in-Q feature, configure an 802.1Q tag type on the untagged edge links (the customer ports) to any value other than the 802.1Q tag for incoming traffic.

For example, in Example configuration on page 100, the 802.1Q tag on the untagged edge links (ports 11 and 12) is 9100, whereas, the 802.1Q tag for incoming traffic is 8100.

To configure 802.1 Q-in-Q tagging as shown in Example configuration on page 100, enter commands such as the following on the untagged edge links of devices C and D.

```
device(config)# tag-type 9100 e 3/1 to 3/2
```

Syntax: [no] tag-type num [**ethernet slot-number/port-number** [**to slot-number/port-number**]]

The *num* parameter specifies the tag-type number and can be a hexadecimal value from 0 - ffff. The default is 8100.

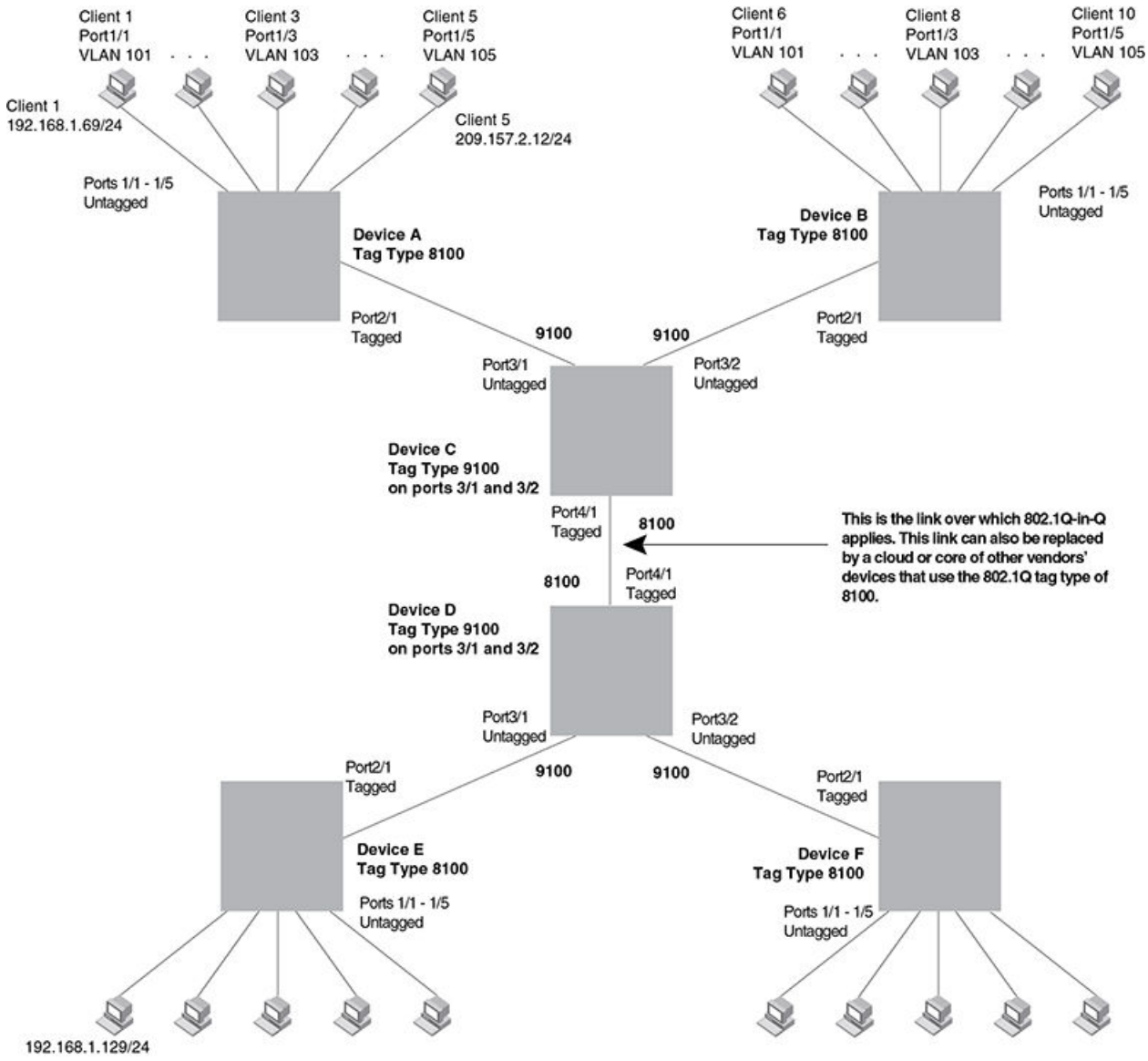
The **ethernet** *port number* **to** *port number* parameter specifies the ports that will use the defined 802.1Q tag. This parameter operates with the following rules:

- If you do not specify a port or range of ports, the 802.1Q tag applies to all Ethernet ports on the device.

Example configuration

Figure 13 shows an example 802.1Q-in-Q configuration.

FIGURE 13 Example 802.1Q-in-Q configuration



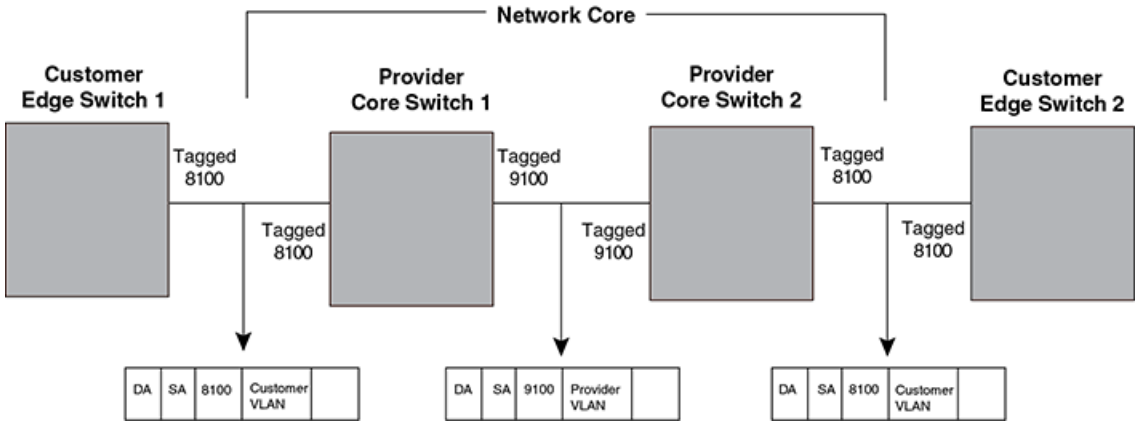
Configuring 802.1q tag-type translation

The introduction of 802.1q tag-type translation provides finer granularity for configuring multiple 802.1q tag-types on a single device, by enabling you to configure 802.1q tag-type per port. This enhancement allows for tag-type translation from one port to the next on tagged interfaces.

802.1Q tag-type translation enables you to configure a separate 802.1q tag-type per port, allowing for tag-type translation from one port to the next on tagged interfaces.

Figure 14 shows a basic example application of the 802.1q tag-type translation feature.

FIGURE 14 802.1q Tag-type translation configuration example 1



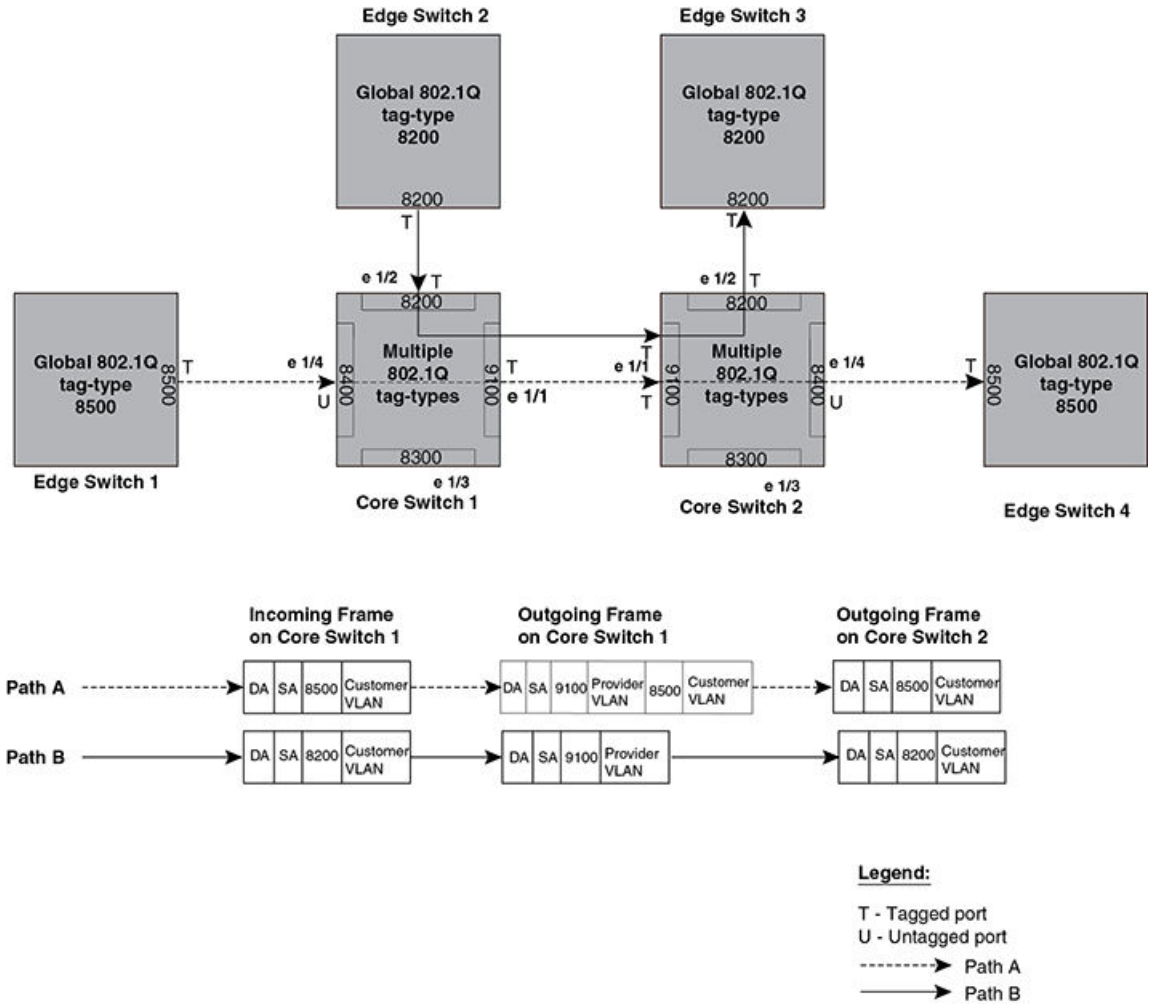
As illustrated in Figure 14, the devices process the packet as follows:

- Customer Edge Switch 1 sends a packet with an 802.1q tag-type of 8100 to Provider Core Switch 1.
- Since the customer-facing interface on Provider Core Switch 1 has the same 802.1q tag-type as the incoming packet, it removes the 8100 tag-type and replaces (translates) it with the 9100 tag-type as it sends the packet to the uplink (Provider Core Switch 2).
- The same process occurs between Provider Core Switch 2 and Customer Edge Switch 2.

Figure 14 shows a simple application of the 802.1q tag-type translation in which all of the ports are tagged and the tag-types between devices match. In this example, each device performs the 802.1q tag-type translation as the packet traverses the network.

Figure 15 shows a more complex example application in which some ports are untagged, not all tag-types between devices match, and the core devices have multiple tag-types. In this example, the tag-type translation feature integrates packets that have single and double tag-types.

FIGURE 15 802.1q Tag-type translation configuration example 2



As illustrated in Figure 15, the devices process the packets as follows:

- Path A: When Core Switch 1 receives the tagged packet from Edge Switch 1, it keeps the 8500 tag-type in the frame header (because the incoming port on Core Switch 1 is untagged) and adds the 9100 tag-type as it sends the packet to the uplink (Core Switch 2). In this case, the packet is double-tagged as it travels between the core devices.
- Path B: When Core Switch 1 receives the tagged packet from Edge Switch 2, it removes the 8200 tag-type and replaces (translates) it with the 9100 tag-type as it sends the packet to the uplink (Core Switch 2).

Configuration rules

Configuration of tag-type ports on the Extreme device are on a per-port basis and follow the same rules as described in Configuration rules on page 100.

Enabling 802.1q tag-type translation

To enable 802.1q tag-type translation, configure an 802.1q tag-type on the provider core link, between the provider core switches (refer to Configuring 802.1q tag-type translation on page 101). Enter commands such as the following.

```
device(config)# tag-type 9100 e 1/1
device(config)# tag-type 8200 e 1/2
device(config)# tag-type 8300 e 1/3
device(config)# tag-type 8400 e 1/4
```

Syntax: [no] tag-type num [**ethernet slot-number/port-number** [**to slot-number/port-number**]]

The *num* parameter specifies the tag-type number and can be a hexadecimal value from 0 - ffff. The default is 8100.

The *slot-number/port-number* [**to***slot-number/port-number*] parameter specifies the ports that will use the defined 802.1q tag-type. This parameter operates with the following rules:

- If the port that you specify is part of a multi-slot LAG, the device automatically applies the 802.1q tag-type to all of the ports that are part of the multi-slot LAG.
- If you do not specify a port or range of ports, the 802.1q tag-type applies to all Ethernet ports on the device.

Miscellaneous VLAN features

Allocating memory for more VLANs or virtual routing interfaces

By default, you can configure up to 512 VLANs and virtual routing interfaces on the router. Although this is the default maximum, the Extreme device can support up to 4090 VLANs and 4090 virtual routing interfaces.

> NOTE
> If many of your VLANs will have an identical configuration, you might want to configure VLAN groups.

If you need to configure more than 512 VLANs, enter commands such as the following at the global CONFIG level of the CLI.

```
device(config)# system-max vlan 2048
device(config)# write memory
device(config)# end
device# reload
```

Syntax: [no] system-max vlan num

The *num* parameter specifies the maximum number of VLANs that can be configured.

Syntax: [no] system-max virtual-interface

The *num* parameter specifies the maximum number of virtual-interfaces that can be configured.

> NOTE
> You must reload the system for the new parameters to take effect.

Configuring uplink ports within a port-based VLAN

You can configure a subset of the ports in a port-based VLAN as uplink ports. When you configure uplink ports in a port-based VLAN, the device sends all broadcast and unknown-unicast traffic from a port in the VLAN to the uplink ports, but not to other ports within the VLAN. Thus, the uplink ports provide tighter broadcast control within the VLAN.

For example, if two ports within a port-based VLAN are Gigabit ports attached to the network and the other ports in the VLAN are 10/100 ports attached to clients, you can configure the two ports attached to the network as uplink ports. In this configuration, broadcast and unknown-unicast traffic in the VLAN does not go to all ports in the VLAN. The traffic goes only to the uplink ports. The clients on the network do not receive broadcast and unknown-unicast traffic from other ports, including other clients.

To configure a port-based VLAN containing uplink ports, enter commands such as the following.

```
device(config)# vlan 10
device(config-vlan-10)# untag ethernet 1/1 to 1/20
device(config-vlan-10)# untag ethernet 2/1 to 2/2
device(config-vlan-10)# uplink-switch ethernet 2/1 to 2/2
```

Syntax: [no] uplink-switch ethernet port-number [**to port-number** | **ethernet port-number**]

In this example, ports 1 - 20 on slot 1 and ports 1 - 2 on slot 2 are added to port-based VLAN 10. The two ports on slot 2 are then configured as uplink ports.

Configuring control protocols in VLANs

You can configure the following protocols on a VLAN:

- Foundry MRP (Refer to *Metro Ring Protocol* Chapter.)
- ERP (Refer to *Ethernet Ring Protocol* Chapter.)
- VSRP (Refer to *Virtual Switch Redundancy Protocol (VSRP)* Chapter.)
- STP (Refer to *Configuring Spanning Tree Protocol* Chapter.)
- RSTP (Refer to *Configuring Rapid Spanning Tree Protocol* Chapter.)

Removing tagged or untagged ports

Use the following commands to remove tagged or untagged ports from a VLAN.

Syntax

remove-tagged-ports

remove-untagged-ports

Command Default

This command can only be used when tagged ports and untagged ports have been applied to a VLAN.

Modes

VLAN configuration mode (config-vlan).

Examples

The following example displays the remove-tagged-ports command.

```
device(config-vlan-100)# remove-tagged-ports
Vlan : 100, Ports removed : ethe 1/1 to 1/2 ethe 4/1 to 4/8
device(config-vlan-100)#
```

The following example displays the remove-untagged-ports command.

```
device(config-vlan-100)# remove-untagged-ports
Vlan : 100, Ports removed : ethe 3/1 to 3/24
device(config-vlan-100)#
```

History

Release version	Command history
5.8.00	This command was introduced.

Removing a VLAN

This command removes tagged and untagged ports from all or defined VLANs.

Syntax

remove-vlan { **all** | **vlan** } [**vlan** *vlan_id*][**to** *vlan_id*]

Command Default

This command can only be used when tagged or untagged ports are defined as part of a VLAN.

Parameters

all

Removes all configured VLANs.

vlan

Use with the modifiers to indicate the VLAN range to remove.

vlan_id

Specifies the VLAN where the ports should be removed.

to

Use with the modifiers to indicate the VLAN range to remove.

vlan_id

Specifies the range of VLANs to beremoved.

Modes

User configuration level.

Examples

The following example displays the command with the all option.

```
device(config-if-e100000-1/1)# remove-vlan all
Port ethe 1/1 removed from tagged vlan : 300 400 500 600 700 800 900 1000 2000 3000
4000 and untagged vlan : 200 .
device(config-if-e100000-1/1)#
```

The following example displays the command with a specified VLAN range.

```
device(config-if-e100000-1/2)# remove-vlan vlan 2 to 4090
Port ethe 1/2 removed from tagged vlan : 300 400 500 600 700 800 900 1000 2000 3000
4000 and untagged vlan : 200 .
device(config-if-e100000-1/2)#
```

The following example displays the command that remove a specific VLAN.

```
device(config-if-e10000-4/1)# remove-vlan vlan 500
Vlan : 500, Ports removed : ethe 4/1
device(config-if-e10000-4/1)#
```

History

Release version	Command history
5.8.00	This command was introduced.

Hardware flooding for layer 2 multicast and broadcast packets

Broadcast and multicast packets do not have a specific recipient. In order for these "special" packets to reach their intended recipient, they need to be sent on all ports of the VLAN (or "flooded" across the VLAN).

You must enable hardware flooding for Layer 2 multicast and broadcast packets on the Extreme device. (Layer 2 multicast packets have a multicast address in the destination MAC address field.)

NOTE
This feature is enabled by default on CES 2000 Series devices.

You can enable hardware flooding for Layer 2 multicast and broadcast packets on a per-VLAN basis.

```
device(config)#
device(config)# vlan 2
device(config-vlan-2)# multicast-flooding
device(config-vlan-2)# exit
```

Syntax: [no] multicast-flooding

NOTE

- This feature cannot be enabled on an empty VLAN; the VLAN must already have ports assigned to it prior to enabling this feature.
- This feature is not supported on Layer 3 protocol-based VLANs.
- If you enable this feature on a VLAN that includes a LAG group, hardware flooding for Layer 2 multicast and broadcast packets occurs only on the LAG group's primary port. Multicast and broadcast traffic for the other ports in the LAG group is handled by software.

Unknown unicast flooding on VLAN ports

Unknown unicast packets do not have a specific (or unicast) recipient. In order for these "special" packets to reach their intended recipient, they needed to be sent on all ports of the VLAN (or "flooded" across the VLAN).

You must enable *hardware* flooding for unknown unicast packets on the Extreme router. It is disabled by default.

NOTE
This feature is enabled by default on the CES 2000 Series devices.

To enable unicast hardware flooding on a VLAN ports and enable software flooding, enter commands such as the following.

```
device(config)# vlan 2
device(config-vlan-2)# unknown-unicast-flooding
device(config-vlan-2)# exit
```

Syntax: [no] unknown-unicast-flooding

Configuring VLAN CPU protection

VLAN CPU protection is recommended for the VLANs which are intended for pure Layer 2 use. This feature will protect the CPU from the flooding of unknown-unicast or multicast or broadcast Layer 2 packets on that VLAN.

When using routing protocols (such as OSPF and others) on a specific VLAN, you need to disable VLAN CPU protection for it to work. This feature is intended for Layer 2 applications and not for Layer 3 routing applications.

CPU protection can be configured on VLANs regardless of whether there are virtual-interface configured on them (Previously, CPU protection was only configurable if a virtual-interface was not configured on the VLAN).

There is a difference in the behavior of CPU protection in each of the following situations:

- When virtual-interfaces are configured on a VLAN, the CPU-protection is done only on unknown-unicast packets from the VLAN. Multicast and broadcast packets from the VLAN will be sent to the CPU. This allows the CPU to process packets such as ARP and OSPF "hello" packets that may be relevant to the device.
- When virtual-interface is not configured on the VLAN, the CPU-protection is performed for all packets (unknown-unicast, multicast and broadcast) from the CPU.
- With vlan-cpu-protection enabled, currently persistent unknown unicast packets are still sent to the CPU for MAC learning purposes. Although the unknown unicast packets are rate limited to the CPU, it may cause high CPU usage and large CPU Traffic Manager queues, which may cause issues.
- Using the **unknown-unicast-mac-entry** command will forward Layer 2 unknown unicast traffic without going to the CPU.

  NOTE
  This feature is enabled by default on the CES 2000 Series devices and cannot be disabled.

VLAN CPU protection is enabled per VLAN. To enable VLAN CPU protection on a VLAN, enter the following command.

```
device(config)# vlan 247
device(config-vlan-24)# tagged ethe 4/1 ethe 4/3
device(config-vlan-24)# vlan-cpu-protection
device(config-vlan-24)# unknown-unicast-mac-entry
```

Syntax: [no] vlan-cpu-protection

NOTE
If **vlan-cpu-protection** command is configured for a VLAN, you should not configure **unknown-unicast-flooding** command or **multicast flooding** command on the same VLAN since these features are redundant to **vlan-cpu-protection**.

Syntax: [no] unknown-unicast-mac-entry

NOTE
The **unknown-unicast-mac-entry** command must be configured with the **vlan-cpu-protection** command, as shown in the example above.

Command changes to support Gen-2 modules

The following commands changed to support Gen-2 modules.

Deprecated commands

vlan-counter exclude-overhead

The **vlan-counter exclude-overhead** command has been deprecated in the XMR Series and MLX Series devices only. The new command is the **exclude-ethernet-overhead** command.

> NOTE
> The **statistics - exclude-ethernet-overhead** command will replace the **vlan-counter exclude-overhead** command when upgrading to the new image.

By default, the VLAN byte counters include the 20-byte Ethernet overhead. You can use the **exclude-ethernet-overhead** command to direct the Extreme device to exclude this overhead when it counts the bytes, as shown in the example below.

```
device(config-statistics)#exclude-ethernet-overhead
```

Syntax: [no] exclude-ethernet-overhead

To disable the configuration, use the **no exclude-ethernet-overhead** command.

> NOTE
> The **vlan-counter exclude-overhead** command is still supported for the CER 2000 Series and CES 2000 Series platforms.

byte-accounting

The **byte-accounting** command has been replaced by the **vlan-accounting on|off** command at the VLAN configuration level for the devices.

All Extreme platforms use **vlan-accounting** command at the global (config-vlan-policy) level.

In addition a new global **vlan-policy - vlan-accounting** command has also been introduced to enable/disable accounting for all VLANs.

The **vlan-accounting on | off** command at the VLAN level takes precedence over global configuration. For example, if VLAN accounting is globally enabled, and the user disables VLAN accounting on VLAN 10, then VLAN accounting for VLAN 10 is disabled.

You can configure MLX Series devices to count bytes received on a VLAN globally or at the VLAN level. By default, Layer 2 VLAN accounting is globally enabled for all VLANs. The VLAN counters are polled every 50 seconds.

To disable VLAN accounting globally for all VLANs, enter the following command at the config-vlan-policy level of the CLI.

```
device(config-vlan-policy)#no vlan-accounting
```

Syntax: [no] vlan-accounting

To disable VLAN accounting globally, enter the **no vlan-accounting** command.

To configure VLAN accounting for specific VLAN, enter the following command.

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

Syntax: [no] vlan-accounting on | off

The **vlan-accounting on** command enables counters for a specific VLAN. The **vlan-accounting off** command disables counters for a specific VLAN.

clear vlan all-vlans statistics

The **clear vlan byte-accounting all-vlans** command has been deprecated. The new command is the **clear vlan all-vlans statistics** command.

To clear VLAN counters for all VLANs, enter the following command.

```
device# clear vlan all-vlans statistics
```

Syntax: clear vlan all-vlans statistics

clear vlan byte-accounting

The **clear vlan byte-accounting** command has been has been deprecated. The new command is the **clear vlan statistics** command.

To clear the VLAN counters on a specific VLAN, say VLAN 10, enter the following command.

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

Syntax: clear vlan *vlan-id* **statistics**

Use the *vlan-id* parameter to specify the name of the VLAN to clear statistic counter on.

Existing display command

The byte counter displayed by the output of **show vlan** command is the number of received bytes across all ports (both G2 and non-G2 ports) in the specified VLAN.

```
device# show vlan
Configured PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 4090
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols : NONE
Untagged Ports : ethernet 2/1 to 2/20 ethernet 3/1 to 3/20 ethernet
PORT-VLAN 2, Name [None], Priority Level0
L2 protocols : NONE
ip-protocol VLAN, Dynamic port disabled
Name: basic
PORT-VLAN 1001, Name [None], Priority Level0
L2 protocols : MRP
Tagged Ports : ethernet 3/1 ethernet 3/12 to 3/13 ethernet 3/20
Bytes received : 6000
```

TABLE 10 show vlan output details

Configured PORT-VLAN entries	Number of port-based VLANs in the configuration.
Maximum PORT-VLAN entries: 4090	Maximum number of port-based VLANs that you can configure.
Default PORT-VLAN id	ID of the default VLAN.
PORT-VLAN	ID of the port-based VLAN.
Name	Name of the port-based VLAN. [None] appears if a name has not been assigned.
Priority Level	Priority level assigned to the port-based VLAN.
L2 protocols	Layer 2 control protocol configured on the VLAN.
Untagged or Tagged Ports	ID of the untagged or tagged ports that are members of the VLAN.
(protocol-based VLANs)	If protocol based VLANs are configured, their type and name appear after the list of ports.
Bytes received	Displays the number of received bytes across all ports in the specified VLAN.

Extended VLAN counters for 8x10G modules

The NI-MLX-10x8G module supports VLAN counters on the ingress and egress ports. The NI-MLX-10x8G module supports packet and byte accounting for 32K counters on both inbound and outbound traffic on a per-VLAN, per-port, per-priority basis. The 8x10G module supports 64 bit VLAN counters for both packet and byte accounting.

To support extended VLAN accounting, the following modes allow you to configure packet and byte accounting on a 8x10G module.

Priority mode - This mode allows accounting to be performed on a per VLAN, per port, per-priority basis. Priority mode is configured on per-module basis. By default, the per-priority accounting mode is disabled, or in other words, accounting is done per VLAN, per port.

Switched or routed separate mode - This mode allows you to specify whether the switched packet and routed packets should be counted separately or not. This mode is configured globally, and by default, switched packets and routed packets are counted together.

Refer to Table 11 on the number of unique port, VLAN's supported per PPCR based on the configuration of "Priority mode" and "Switched or routed separate mode"..

TABLE 11 Internal priority of switched and routed packets

Switched and routed packets	Account based on the internal priority of the packet- Yes or No	Number of unique port-VLANs that have counters (per-PPCR).
Switch or Route separately	Yes	2047 on ingress and 2047 on egress; each set having 16 counters
Switch or Route separately	No	16383 in ingress and 16383 on egress; each set having 2 counters
Switch or route combined	Yes	4095 on ingress and 4095 on egress; each set having 8 counters
Switch or route combined	No	32767 on ingress and 32767 on egress; each set having 1 counter

Configuring extended VLAN counters

The Gen-2 modules supports the following global configuration commands.

Enabling accounting on per-slot basis

You can enable or disable per-VLAN priority accounting mode on all or a per-slot basis on the ingress and egress counters. To enable accounting on per-slot basis, enter the following command.

Layer 2 VLAN accounting is enabled by default. Counters are polled once every 50 seconds.

```
device(config)#statistics
device(config-statistics)#extended-counters priority all
```

Syntax: [no] extended-counters priority all | *slot-number*

The *slot-number* variable specifies the ID of 8x10 module on which you can perform accounting on per-slot basis.

If the all option is specified, the configuration command is remembered in the system when the 8x10 module is removed.

If you dynamically enable or disable the **extended-counters priority** configuration, the sum of counters displayed on a per-priority basis will not be same as the aggregate count displayed on a per-port or per-VLAN basis.

Enabling accounting on switched or routed packets

To enable or disable accounting on switched packets and routed packets separately, enter the following example:

```
device(config)#statistics
device(config-statistics)#extended-counters routed-switched
```

Syntax: [no] extended-counters routed-switched

If you dynamically enable or disable the **extended-counters routed-switched** configuration, the current counters are saved and added to the count of aggregate packet and byte counters on a per-port or per-VLAN basis and displayed in the output of combined counters.

Displaying VLAN counters

The **show vlan** commands changed to display port-vlan counters for 8x10G modules.

To display VLAN counters information for specific VLAN, enter the following command.

```
device# show vlan 10 statistics
VLAN 10: Extended Routed/Switched Counters (only applicable for G2 modules):
Slot 12: < -- module with per-VLAN/port/priority based accounting
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                           <snip>
     p6   0                0                0                0
     p7   0                0                0                0
eth 12/2  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                           <snip>
     p6   0                0                0                0
     p7   0                0                0                0
eth 12/3         -- Extended-counter resource allocation failed -   < -- On encountering Stats ID
allocation failure
eth 12/4  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
<snip>
     p6   0                0                0                0
     p7   0                0                0                0
Slot 14: < -- module with per-VLAN/port based accounting
```

Syntax: show vlan *vlan-id* **statistics** [**detail** | **routed** | **switched**]

The *vlan-id* parameter specifies the VLAN ID of the port.

The slot/port parameter specifies the interface module location of a 8x10g module.

Use the **routed** option to view the counters of routed packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **switched** option to view the counters of switched packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **detail** option to view the counters of routed packets, counters of switched packets, and counters of both routed and switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

If the "routed" or "switched" option is specified, counters for only routed and switched packets are displayed respectively, otherwise the output displays combined statistics for both routed and switched packets.

TABLE 12 Output descriptions of the show vlan command

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

Displaying VLAN counters for a specific port

To display VLAN counters information for specific port on a VLAN, enter the following command.

```
device# show vlan 10 statistics ethernet 14/1
VLAN 10: Extended Routed/Switched Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
```

To display VLAN counters information for routed packets on a specific port on a VLAN, enter the following command.

```
device# show vlan 10 statistics ethernet 14/1 routed
VLAN 10: Extended Routed Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
To display VLAN counters information for switched packets on a specific port on a VLAN, enter the following
command.
device# show vlan 10 statistics ethernet 14/1 switched
VLAN 10: Extended Switched Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
```

To display detailed VLAN counters information on a specific port on a VLAN, enter the following command.

```
device# show vlan 10 statistics ethernet 14/1 detail
VLAN 10: Extended Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1
 Routed   0                0                0                0
 Switched 0                0                0                0
 Combined 0                0                0                0
```

Syntax: show vlan *vlanid* **statistics ethernet** *port-id* [**detail** | **routed** | **switched**]

The *vlan-id* parameter specifics the VLAN ID of the port.

The *slot/port* parameter specifies the interface module location of a 8x10g module.

Use the **routed** option to view the counters of routed packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **switched** option to view the counters of switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **detail** option to view the counters of routed packets, counters of switched packets, and counters of both routed and switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

If the "routed" or "switched" option is specified, counters for only routed and switched packets are displayed respectively, otherwise the output displays combined statistics for both routed and switched packets.

TABLE 13 Output description for the show vlan command

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

Clearing extended VLAN counters

You can use the following commands to clear the extended VLAN counters.

Clearing counters for all VLANs

To clear the ingress and egress packet and byte counters for routed packets and switched packets on all VLANs, enter the following command.

```
device# clear vlan all-vlans statistics
```

Syntax: clear vlan all-vlans statistics [**switched**]

Enter the **switched** keyword to clear only the counters of switched packets. If you do not specify the **switched** keyword, the counters for both routed packets and switched packets are cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing counters for a specific VLAN

To clear the VLAN counters for a specific VLAN, enter the following command.

device# clear vlan 10 statistics

Syntax: clear vlan *vlan-id* **statistics** [**switched**]

Use the *vlan-id* option to specify the VLAN ID of the port for which you want to clear the counters.

Enter the **switched** keyword to clear only the counters of switched packets. If you do not specify the **switched** keyword, the counters for both routed packets and switched packets are cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing VLAN and port counters

To clear VLAN, port, and priority counters for specific VLAN and port combinations, enter the following command.

```
device# clear vlan 10 statistics ethernet 1/2 switched
```

Syntax: clear vlan *vlan-id* **statistics ethernet** *port-id* [**switched**]

Use the *vlan-id* option to specify the VLAN ID of the port for which you want to clear the counters.

Use the *port-id* option to specify the port for with you want to clear the counters.

Specify the **switched** keyword to clear counters for the switched packets. If the **switched** keyword is not specified the counters for both routed packets and switched packets will be cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing VLAN counters on a port with a specific priority

To clear counters for a specific port in a VLAN with specific priority, enter the following command.

```
device# clear vlan 10 statistics ethernet 1/2 priority 3 switched
```

Syntax: clear vlan *vlan-id* **statistics ethernet** *port-id* **priority** *0-7* [**switched**]

Use the *vlan-id* option to specify the VLAN ID of the port for which you want to clear the counters.

Use the *port-id* option to specify the port for with you want to clear the counters.

Specify the **switched** keyword to clear counters for the switched packets. If the **switched** keyword is not specified the counters for both the routed packets and switched packets will be cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing extended counters statistics on a port

To clear all extended counters statistics simultaneously for a single port, enter the following command.

```
device# clear statistics ethernet 1/2 extended counters
```

Syntax: clear statistics ethernet *port_id orrange* **extended counters**

Use the *port_id or range* option to specify the port or range for which you want to clear the extended counters.

Clearing extended counters statistics on specific slot

To clear all extended counter statistics simultaneously for a slot, enter the following command.

```
device# clear statistics slot 2 extended counters
```

Syntax: clear statistics slot *slot-id* **extended counters**

Use the *slot-id* option to specify the slot number for which you want to clear the extended counters.

IP interface commands

You can display and clear the counter details of the physical and virtual IP interfaces.

Displaying IP interface counters

You can display aggregate count of the routed packets and switched packets of an IP interface using the following command.

```
<< If Routed/Switched separate mode >>
device# show ip interface ve 10 statistics
Extended Routed Counters (only applicable for G2 modules):
Total     RxPkts           TxPkts           RxBytes          TxBytes
          0                0                0                0
<< If Routed/Switched combined mode>>
device# show ip interface ve 10 statistics
```

```
Extended Routed/Switched Counters (only applicable for G2 modules):
Total     RxPkts           TxPkts           RxBytes          TxBytes
                    0                0                0                0
```

Syntax: show ip interface ethernet *port-id* **statistics**

Specify the *port id* of the interface for which you want to display the routed and switched packets aggregate count.

TABLE 14 show ip interface command output details

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

Displaying IP virtual interface counters

To display the counters for each physical port of a virtual IP interface, use the following command.

```
device#show ip interface ve 10 statistics ethernet 12/1
Extended Routed Counters (applicable for G2 modules only):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                                <snip>
     p6   0                0                0                0
     p7   0                0                0                0
```

Syntax: show ip interface ve *vid* **statistics** [**ethernet** *port-id*]

Specify the *port id* of the virtual interface for which you want to display.

TABLE 15 show ip interface ve statistics command output details

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

Displaying detailed IP virtual interface counters

To display the detailed aggregate count of the routed packets and switched packets of a virtual IP interface are configured separate mode, use the following command.

```
device# show ip interface ve 10 statistics detail
Extended Routed Counters (applicable for G2 modules only):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                                <snip>
     p6   0                0                0                0
     p7   0                0                0                0
```

```
eth 12/2   0                0                0                0
     p0    0                0                0                0
                          <snip>
     p7    0                0                0                0
```

To display the detailed aggregate count of the routed packets and switched packets of a virtual IP interface are configured combined mode, use the following command.

```
device# show ip interface ve 10 statistics detail
Extended Routed/Switched  Counters (applicable for G2 modules only):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                              <snip>
     p6   0                0                0                0
     p7   0                0                0                0
```

Syntax: show ip interface ve *vid* **statistics** [**detail**]

Use the *vid* option to specify the interface name of the virtual IP interface for which you want to display the routed and switched packets aggregate count.

TABLE 16 show ip interface ve output details

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

Clearing IP interface counters

When clearing IP interface counters, the counters that are cleared are dependent on the accounting mode configuration. If you have configured accounting mode to count routed and switched packets separately, only the routed counters will be cleared. If you have configured accounting mode to count routed and switched packets together, the combined counter will be cleared.

To clear the routed packet and switched packet counters of a specific IP interface, enter the following command.

```
device# clear ip interface ethernet 1/2 statistics
```

Syntax: clear ip interface ethernet *port-id* **statistics**

Use the *port-id* option to specify the interface name to clear.

Clearing IP virtual interface counters

When clearing IP virtual interface counters, the counters that are cleared are dependent on the accounting mode configuration. If you have configured accounting mode to count routed and switched packets separately, only the routed counters will be cleared. If you have configured accounting mode to count routed and switched packets together, the combined counter will be cleared.

To clear the routed packet and switched packet counters of a specific virtual IP interface, enter the following command:

```
device# clear ip interface ve 2 statistics
```

Syntax: clear ip interface ve *vid* **statistics**

Use the *vid* variable to specify the virtual interface name to clear.

Transparent VLAN flooding

The transparent VLAN flooding (TVF) allows packets to be forwarded without any form of CPU intervention including MAC learning and MAC destination lookups.

NOTE
Because TVF floods all VLAN packets in hardware, it is not expected to work in conjunction with routing functions such as establishing a routing protocol neighbor and Layer 3 forwarding, even when the VLAN has a VE interface configured.

NOTE
Enabling transparent VLAN flooding causes the bypass of the load balancing mechanisms of a Link Aggregation Group (LAG).

This implementation of transparent VLAN flooding has the following attributes:

- The ability to always distribute traffic to all members of a VLAN in hardware.
- It requires no CPU intervention and consequently can handle line-rate traffic forwarding.
- Because this feature does not use any MAC address entries in the CAM, it is useful when MAC address entries need to be conserved.
- VLAN members can be tagged or untagged ports including a mix of tagged and untagged ports.
- The maximum number of transparent VLAN flooding instances is 4090.
- You can mix and match ports with different speeds.
- Other Layer 2 capabilities such as spanning tree are unaffected.
- Output Layer 3 ACLs may be associated with each port that is part of the VLAN instance being transparently flooded.
- You cannot configure a VE interface on a VLAN when transparent VLAN flooding is used.

This feature is particularly useful in situations where MAC learning is not required for traffic forwarding. Examples of where this feature is useful include:

- A configuration where there are only two ports in a VLAN.
- Where traffic is looped back to a device through another VLAN for firewall or mirroring purposes.
- Where the number of MAC addresses will significantly overwhelm the memory and compute resources of a system.

NOTE
Packets that arrive on an interface with the same destination MAC address as the interface are forwarded in hardware just like packets with other destination addresses.

Enabling VLAN transparent forwarding

To enable VLAN transparent forwarding on VLAN 10, use the following commands.

```
device(config)# vlan 10
device(config-vlan-10)# transparent-hw-flooding
```

Syntax: [no] transparent-hw-flooding

Layer 2 inbound ACLs may be applied to any port in a VLAN that has transparent VLAN flooding (TVF) enabled.

Layer 2 or Layer 3 outbound ACLs may be applied to any port in a VLAN that has TVF enabled.

Input Layer 3 ACLs must be applied to a virtual routing interface on the VLAN and should not be attached to a port directly. Note that the ACL would apply to all ports in the VLAN by default. If this is not desired, a subset of ports in the VLAN may be specified, as in the following configuration.

```
device(config)# interface ve 1
device(config-vif-1)# ip access-group 101 in ethernet 1/1
device(config-vif-1)# ip access-group ve-traffic
```

The preceding example binds the ACL 101 to the virtual routing interface and configures the virtual routing interface to apply the input ACL 101 for switched and routed traffic received on port 1/1.

Enabling VLAN LAG load balancing

VLAN LAG load balancing allows for the transparent VLAN flooding feature to flood an outgoing LAG and load-balance correctly across all ports of the LAG.

TABLE 17 Hardware-enhanced VLAN module and FID pool size matrix

FID pool size —> Number of member ports	512	1024	2048	4096
2	256 (each VLAN requires 2 entries)	512 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)	1024 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)	2048 The maximum number of TVF LAG load balancing instances is limited to 2016. (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)
4	128	256	512 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)	1024 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)
8	64	128	256	512 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)
16	32	64	128	256
32	16	32	64	128

TABLE 18 Software-enhanced limit for 48x1G and 24x1G modules

Ingress —> Egress	48x1G and 24x1G modules
48x1G and 24x1G modules	480 VLANs with up to 8 port member LAG

TABLE 19 Software-enhanced limit for 20x10G, 8x10G, 2x100G, and 4x40G modules

Ingress —> Egress	20x10G, 8x10G, 2x100G, and 4x40G modules
20x10G, 8x10G, 2x100G, and 4x40G modules	A maximum of 1024 VLANs with the 4-port member LAG A maximum of 256 VLANs with the 16-port member LAG

NOTE
Downgrading to prior releases may cause the pool size to be removed from the configuration and disabling of TVF VLAN LAG load balancing when the pool size is set to 4096.

To enable transparent VLAN LAG load balancing on VLAN 10, use the following commands.

```
device(config)# vlan 10
device(config-vlan-10)# transparent-hw-flooding lag-load-balancing
```

Syntax: [no] transparent-hw-flooding lag-load-balancing

NOTE
TVF VLAN LAG load balancing is applicable to Policy Based Routing (PBR) telemetry applications only.

Configuring TVF FID pool size

To configure maximum FID pool size for transparent VLAN flooding LAG load balancing globally, use the following command.

```
device(config)# system-max tvf-lag-lb-fid-pool 512
```

Syntax: [no] system-max tvf-lag-lb-fid-pool *number*

The *number* variable specifies the pool size values which are 0, 512, 1024, 2048, and 4096. The default is 0 (feature is disabled).

For information about the hardware-enhanced VLAN module and FID pool size matrix and software-enhanced limit for different modules, refer to Enabling VLAN LAG load balancing on page 120.

NOTE
Downgrading this device to earlier releases may cause the pool size to be removed from the configuration when the FID pool size is set to 4096.

Configuring TVF FID group size

To configure maximum FID group size for transparent VLAN flooding LAG load balancing globally, use the following command.

```
device(config)# system-max tvf-lag-lb-fid-group 8
```

Syntax: [no] system-max tvf-lag-lb-fid-group *number*

The *number* specifies the group values which are 2, 4, 8, 16, and 32. Default is 4.

(If LAG load balancing is enabled) If TVF FID group size is 32, the total of tvf-domains plus tvf-vlan is 128 for the chassis.

The value of **system-max tvf-lag-lb-fid-group** influences the maximum number of ports allowed in a LAG in a VLAN as follows:

system-max trunk-num	128 ports	64 or 32 ports
Supported values of **system-max tvf-lag-lb-fid-group**	16 or lower	32 or lower

NOTE
For a **system-max tvf-lag-lb-fid-group** value of 32, the maximum supported sum of tvf-domains and "transparent-hw-flooding lag-load-balancing" VLANs is 128.

Transparent VLAN flooding domain

The transparent VLAN flooding (TVF) domain provides an infrastructure to increase the overall egress traffic streams.

Currently Policy-based Routing (PBR) TVF uses flooding on VLANs for traffic streams and supports a maximum of 4090 egress traffic streams. With the TVF domain implementation, traffic can be flooded to the TVF domain by setting the TVF domain as a PBR next hop. The TVF domain supports 2016 TVF instances with LAG load balancing and, together with 4090 TVF instances without LAG load balancing, scales the overall egress traffic flow support to 6106.

NOTE
The TVF domain supports only TVF with LAG load balancing.

NOTE
A maximum of 480 load balancing instances are supported by the hardware in the 24x10G module. The egress streams with TVF LAG load balancing configured over 480 instances will be forwarded without LAG load balancing.

Configuring the TVF domain

The following steps configure the TVF domain and add member ports to the TVF domain.

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

```
device# configure terminal
```

2. Enter the **tvf-domain** command to configure a TVF domain with a named TVF domain ID.

```
device(config)# tvf-domain 1 name domainuser
```

   Valid values for the TVF domain ID are from 1 through 2016. The name can be up to 64 characters in length.

3. Enter the **port** command to add member ports to the TVF domain.

```
device(config-tvf-domain-1)# port ethernet 1/1 ethernet 2/1
```

   The number of ports in the LAG that can be added to the TVF domain is limited based on the maximum FID pool size configured using the **system-max tvf-lag-lb-fid-group** command.

Setting the TVF domain as a PBR next hop

The following steps configure the TVF domain as the next hop for a route map to support transparent VLAN flooding (TVF) with LAG load balancing.

Configure the required IPv4 ACLs and IPv6 ACLs to be added to the route map. For more information about configuring ACLs, refer to the *Extreme NetIron Security Configuration Guide*.

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

```
device# configure terminal
```

2. Enter the **route-map** command to define the route and specify the match criteria and the resulting action if all of the match clauses are met.

```
device(config)# route-map test-route permit 99
```

3. Add IPv4 ACLs or IPv6 ACLs or both to match the IP address that is permitted by the ACL.

```
device(config-routemap test-route)# match ip address SGW_1_ACL
device(config-routemap test-route)# match ipv6 address SGW_2_ACL
```

4. Enter the **set next-hop-tvf-domain** command to configure a TVF domain as the next hop for a route map.

```
device(config-routemap test-route)# set next-hop-tvf-domain 1
```

Configuration example of TVF domain as PBR next hop for TVF with LAG load balancing

Complete the following steps to configure TVF domain as PBR next hop for TVF with LAG load balancing.

1. Configure system maximum requirements to support LAGs and TVF LAG load balancing.

```
device(config)# system-max trunk-num 32
device(config)# system-max tvf-lag-lb-fid-pool 4096
device(config)# system-max tvf-lag-lb-fid-group 4
```

2. Configure the TVF domain and assign the ports.

```
device(config)# tvf-domain 1
device(config-tvf-domain-1)# port ethernet 1/1 ethernet 2/1 ethernet 5/1 ethernet 6/1
```

3. Configure IPv6 ACLs and IPv4 ACLs to be routed using PBR.

```
device(config)# ipv6 access-list v6_Permit_Any
device(config-ipv6-access-list v6_Permit_Any)# permit ipv6 any any
device(config-ipv6-access-list-v6_Permit_Any)# exit
device(config)# ipv6 access-list v6_brc_Test_p000_bi_moof
device(config-ipv6-access-list v6_brc_Test_p000_bi_moof)# permit vlan 1001 ipv6 any any
device(config-ipv6-access-list v6_brc_Test_p000_bi_moof)# exit
device(config)# ip access-list extended v4_Permit_Any
device(config-ext-nacl-v4_Permit_Any)# permit ip any any
device(config-ext-nacl-v4_Permit_Any)# exit
device(config)# ip access-list extended v4_brc_Test_p000_bi_moof
device(config-ext-v4_brc_Test_p000_bi_moof)# permit vlan 1001 ip any any
```

4. Configure route maps and add the configured ACLs.

```
device(config)# route-map Mall permit 1001
device(config-routemap Mall)# rule-name brc_Test_p000_bi_moof
device(config-routemap Mall)# match ip address v4_brc_Test_p000_bi_moof
device(config-routemap Mall)# match ipv6 address v6_brc_Test_p000_bi_moof
```

5. Add the configured TVF domain as the next hop to the route map.

```
device(config-routemap Mall)# set next-hop-tvf-domain 1
```

6. Configure a VLAN and add the interfaces required for the TVF LAG and LAG load balancing.

```
device(config)# vlan 1 name DEFAULT-VLAN
device(config-vlan-1)# no untagged ethernet 1/1 ethernet 2/1 ethernet 5/1 ethernet 6/1
device(config-vlan-1)# no spanning-tree
```

7. Create a LAG and add the interfaces.

```
device(config)# lag blue static
device(config-lag-blue)# ports ethernet 5/1 ethernet 6/1
device(config-lag-blue)# primary-port 5/1
device(config-lag-blue)# deploy
```

8. Apply PBR to the ingress interfaces.

```
device(config)# interface ethernet 11/1
device(config-if-e10000-11/1)# qos multicast shaper best-effort rate 10000
device(config-if-e10000-11/1)# qos multicast shaper guaranteed rate 100000000
device(config-if-e1000-11/1)# enable
device(config-if-e1000-11/1)# ip policy route-map Mall
device(config-if-e1000-11/1)# ipv6 policy route-map Mall
device(config-if-e1000-11/1)# allow-all-vlan pbr
```

Displaying TVF domain information

Use the **show tvf-domain** command to view details of the TVF domain configuration, such as the TVF domain ID, ports added to the TVF domain, system maximum requirements to support TVF LAG load balancing, and so on.

The following example displays information about the TVF domain.

```
device(config)# show tvf-domain
**** TVF LAG Load Balancing ****
TVF LAG Load Balancing is enabled!
TVF FID pool size: 4096, Max FID groups: 2048, FID group size: 2
2047 (VLAN 32, TVF Domain 2015) TVF LAG Load balancing groups are configured
TVF LAG Load balancing FID programming is done

TVF domain memory usage : 735848 bytes
Per entry usage         : 365 bytes

TVF Domain ID 1, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 2, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 3, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 4, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 6, Name [None]
Ports : ethe 8/5 to 8/8
```

The following example displays the information of a specific TVF domain.

```
device(config)# show tvf-domain 1
TVF Domain ID 1, Name [None]
Ports : ethe 8/5 to 8/8
------------------------------------------------------------
Port  Type      Protocol  State
8/5   TRUNK     NONE      UP
8/6   TRUNK     NONE      UP
8/7   TRUNK     NONE      UP
8/8   TRUNK     NONE      UP
Group ID: 33, FID Base 0x00009ffe, FID Count 2
tvf_lag_lb_fid0: 0x00009ffe, mask ethe 8/5 ethe 8/7
tvf_lag_lb_fid1: 0x00009fff, mask ethe 8/6 ethe 8/8
```

The following example displays a brief summary of all the configured TVF domains.

```
device(config)# show tvf-domain brief
**** TVF LAG Load Balancing ****
TVF LAG Load Balancing is enabled!
TVF FID pool size: 4096, Max FID groups: 2048, FID group size: 2
2047 (VLAN 32, TVF Domain 2015) TVF LAG Load balancing groups are configured
TVF LAG Load balancing FID programming is done

TVF domain memory usage : 735848 bytes
Per entry usage         : 365 bytes
TVF   Name       Ports
----- ----       -----
1     [None]     Ports : ethe 8/5 to 8/8
2     [None]     Ports : ethe 8/5 to 8/8
3     [None]     Ports : ethe 8/5 to 8/8
4     [None]     Ports : ethe 8/5 to 8/8
6     [None]     Ports : ethe 8/5 to 8/8
7     [None]     Ports : ethe 8/5 to 8/8
8     [None]     Ports : ethe 8/5 to 8/8
9     [None]     Ports : ethe 8/5 to 8/8
10    [None]     Ports : ethe 8/5 to 8/8
11    [None]     Ports : ethe 8/5 to 8/8
12    [None]     Ports : ethe 8/5 to 8/8
13    [None]     Ports : ethe 8/5 to 8/8
14    [None]     Ports : ethe 8/5 to 8/8
15    [None]     Ports : ethe 8/5 to 8/8
16    [None]     Ports : ethe 8/5 to 8/8
17    [None]     Ports : ethe 8/5 to 8/8
18    [None]     Ports : ethe 8/5 to 8/8
19    [None]     Ports : ethe 8/5 to 8/8
20    [None]     Ports : ethe 8/5 to 8/8
21    [None]     Ports : ethe 8/5 to 8/8
22    [None]     Ports : ethe 8/5 to 8/8
23    [None]     Ports : ethe 8/5 to 8/8
24    [None]     Ports : ethe 8/5 to 8/8
25    [None]     Ports : ethe 8/5 to 8/8
26    [None]     Ports : ethe 8/5 to 8/8
27    [None]     Ports : ethe 8/5 to 8/8
28    [None]     Ports : ethe 8/5 to 8/8
29    [None]     Ports : ethe 8/5 to 8/8
30    [None]     Ports : ethe 8/5 to 8/8
31    [None]     Ports : ethe 8/5 to 8/8
32    [None]     Ports : ethe 8/5 to 8/8
33    [None]     Ports : ethe 8/5 to 8/8
34    [None]     Ports : ethe 8/5 to 8/8
35    [None]     Ports : ethe 8/5 to 8/8
36    [None]     Ports : ethe 8/5 to 8/8
37    [None]     Ports : ethe 8/5 to 8/8
```

The following example displays detailed information of each TVF domain.

```
device(config)# show tvf-domain detail
**** TVF LAG Load Balancing ****
TVF LAG Load Balancing is enabled!
TVF FID pool size: 4096, Max FID groups: 2048, FID group size: 2
2047 (VLAN 32, TVF Domain 2015) TVF LAG Load balancing groups are configured
TVF LAG Load balancing FID programming is done

TVF domain memory usage : 735848 bytes
Per entry usage         : 365 bytes

TVF Domain ID 1, Name [None]
Ports : ethe 8/5 to 8/8
----------------------------------------------------------
Port  Type      Protocol  State
8/5   TRUNK     NONE      UP
8/6   TRUNK     NONE      UP
8/7   TRUNK     NONE      UP
8/8   TRUNK     NONE      UP
Group ID: 34, FID Base 0x00009ffe, FID Count 2

TVF Domain ID 2, Name [None]
Ports : ethe 8/9 to 8/12
----------------------------------------------------------
Port  Type      Protocol  State
8/9   TRUNK     NONE      UP
8/10   TRUNK     NONE      UP
8/11   TRUNK     NONE      UP
8/12   TRUNK     NONE      UP
Group ID: 33, FID Base 0x00009ffe, FID Count 2
```

The following example displays the details of the port configured in the TVF domain.

```
device(config)# show tvf-domain ethernet 8/6
TVF Domain : 1
TVF Domain : 2
TVF Domain : 3
TVF Domain : 4
TVF Domain : 6
TVF Domain : 7
TVF Domain : 8
TVF Domain : 9
TVF Domain : 10
TVF Domain : 11
TVF Domain : 12
TVF Domain : 13
TVF Domain : 14
TVF Domain : 15
TVF Domain : 16
TVF Domain : 17
TVF Domain : 18
TVF Domain : 19
TVF Domain : 20
TVF Domain : 21
TVF Domain : 22
TVF Domain : 23
TVF Domain : 24
```

Transparent firewall mode

The transparent firewall mode allows the device to switch control packets destined to itself. By default, NetIron OS devices will drop control packets received with the device's MAC address as the packet's destination MAC address (that is, packets destined to the switch or router). Under the transparent firewall mode, switching packets destined to itself is allowed. The transparent firewall mode feature is a per VLAN configuration and is disabled by default.

Enabling a transparent firewall

To set the mode to transparent, enter a command such as the following.

```
device(config-vlan-10)# transparent-fw-mode
```

To set the mode to routed, enter a command such as the following.

```
device config-vlan-10)# no transparent-fw-mode
```

Syntax: [no] transparent-fw-mode

NOTE
Transparent firewall mode is available only on the CER 2000 Series and CES 2000 Series devices.

Displaying VLAN information

After you configure the VLANs, you can view and verify the configuration using the commands discussed in this section.

Displaying VLAN information

Use the **show vlan** command under the vlan-policy configuration.

NOTE
VLAN byte counters are displayed in the output of the **show vlan** command on an MPLS enabled VE interface.

```
device (config)# show vlan
Configured PORT-VLAN entries: 4
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level -,
Priority Force 0
Topo HW idx : 0 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : NONE
Untagged Ports : ethe 1/1 to 1/48 ethe 2/1 to 2/2
Associated Virtual Interface Id: NONE
PORT-VLAN 100, Name [None], Priority Level -, Priority
Force 0
Topo HW idx : 2 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : ERP
Tagged Ports : ethe 1/1 ethe 1/10 ethe 1/29
Associated Virtual Interface Id: NONE
PORT-VLAN 200, Name [None], Priority Level -, Priority
Force 0
Topo HW idx : 0 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : NONE
Tagged Ports : ethe 1/1 ethe 1/10 ethe 1/29
Associated Virtual Interface Id: 120
PORT-VLAN 4095, Name CONTROL-VLAN, Priority Level -,
Priority Force 0
Topo HW idx : 1 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : NONE
Associated Virtual Interface Id: NONE
```

Syntax: show vlan [vlan-id] [brief | detail] [Ethernet slot/port] [begin expression | exclude expression | include expression]

The output shows the following information.

TABLE 20 Output of show vlan

This field...	Displays...
Configured PORT-VLAN entries	Number of port-based VLANs in the configuration.
Maximum PORT-VLAN entries: 4090	Maximum number of port-based VLANs that you can configure.
Default PORT-VLAN id	ID of the default VLAN.
PORT-VLAN	ID of the port-based VLAN
Name	Name of the port-based VLAN. [None] appears if a name has not been assigned.
Priority Level	Priority level assigned to the port-based VLAN
Priority Force	The priority force is configured on the ingress port with a priority value between 0 and 7. The default is 0. The priority force option allows you to force the priority on the ingress pram, or choose not to enforce the priority.
Topo HW idx	A topology hardware index is a unique hardware ID that is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The range for hw-index is 0 - 511.
Topo SW idx	The topology group id associated with the VLAN.
Topo next vlan	Next VLAN id in the topology group.
L2 protocols	Layer 2 control protocol configured on the VLAN
Untagged or Tagged Ports	ID of the untagged or tagged ports that are members of the VLAN
Associated Virtual Interface Id	The ve interface id is displayed when a router-interface is configured for the VLAN. If no router-interface is configured, the field displays NONE.
Bytes received	Displays the number of received bytes across all ports for a specified VLAN. By default, the vlan-accounting command is turned on and hence the Bytes received field is also displayed by default. However, if VLAN accounting for a VLAN is turned off, then the Bytes received field is not displayed in the output. For more information on enabling VLAN byte counters, refer to Extended VLAN counters for 8x10G modules on page 112.

To display information for a specific VLAN, enter a VLAN id as shown in the example below.

```
device(config-vlan-13)#show vlan 2001
PORT-VLAN 2001, Name [None], Priority Level 0, Priority Force 0
Topo HW idx    : 0    Topo SW idx: 257    Topo next vlan: 0
L2 protocols   : MRP
Tagged Ports   : ethe 2/1 to 2/6 ethe 2/11 to 2/14 ethe 2/23 to 2/24
Untagged Ports : ethe 1/1 ethe 1/5
Associated Virtual Interface Id: NONE
```

Displaying VLAN information for specific ports

To determine which VLANs a port is a member of, enter the following command.

```
device# show vlan e 4/1
VLANS 1
VLANs 100
```

Syntax: show vlan ethernet slot-number/port-number [| [**begin expression** | **exclude expression** | **include expression**]

The **ethernet** *slot-number/port-number* parameter specifies a port. The command lists all the VLAN memberships for the port.

The output shows the following information.

TABLE 21 Output of show vlan Ethernet

This field...	Displays...
VLANs	The IDs of the VLANs that the port is a member of.

Displaying VLAN status and port types

To display detailed information about the state, port types, port modes, of a VLAN, as well as control protocols configured on the VLAN, enter the following command.

```
device# show vlan detail
Untagged Ports  : ethernet 2/1 to 2/20 ethernet 4/4
Tagged Ports    : None
Dual-mode Ports : ethernet 3/1 to 3/20 ethernet 4/1 to 4/3
Default VLAN    : 1
Control VLAN    : 4095
VLAN Tag-type   : 0x8100
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
2/1   PHYSICAL  UNTAGGED  NONE      DISABLED
2/2   PHYSICAL  UNTAGGED  NONE      DISABLED
2/3   PHYSICAL  UNTAGGED  NONE      DISABLED
2/4   PHYSICAL  UNTAGGED  NONE      DISABLED
2/5   PHYSICAL  UNTAGGED  NONE      DISABLED
.
. (output edited for brevity)
.
4/1   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/2   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/3   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/4   PHYSICAL  UNTAGGED  NONE      DISABLED
PORT-VLAN 100, Name [None], Priority Level0
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
4/1   PHYSICAL  TAGGED    STP       FORWARDING
4/2   PHYSICAL  TAGGED    STP       BLOCKING
```

Syntax: show vlan [**detail** | **brief**] [**begin expression** | **exclude expression** | **include expression**]

The output shows the following information.

TABLE 22 Output of show vlan detail

This field...	Displays...
Untagged Ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as untagged ports in all the VLANs on the device.
Tagged Ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as tagged ports in all the VLANs on the device.
Dual-mode ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as dual-mode ports in all the VLANs on the device.
Default VLAN	ID of the default VLAN
Control VLAN	ID of the control VLAN
PORT-VLAN #, Name, Priority Level	Information for each VLAN in the output begins with the VLAN type and its ID, name and priority level. Then ports that are members of the VLAN are listed, with the following information:
Port	Port slot-number/port-number
Type	Port type: physical or LAG
Tag-Mode	Tag mode of the port: untagged, tagged, or dual-mode
Protocol	Protocol configured on the VLAN.

TABLE 22 Output of show vlan detail (continued)

This field...	Displays...
State	Current state of the port such as disabled, blocking, forwarding, etc.

Displaying VLAN group information

To display information about VLAN groups, enter the following command.

```
device# show vlan-group 10
Configured VLAN-Group entries : 1
Maximum VLAN-Group entries : 32
VLAN-GROUP 10
Number of VLANs: 4
VLANs: 10 to 13
Tagged ports: ethernet 3/1
```

Syntax: show vlan-group [**vlan-group-id**] [**begin** *expression* | **exclude** *expression* | **include** *expression*]

The output shows the following information.

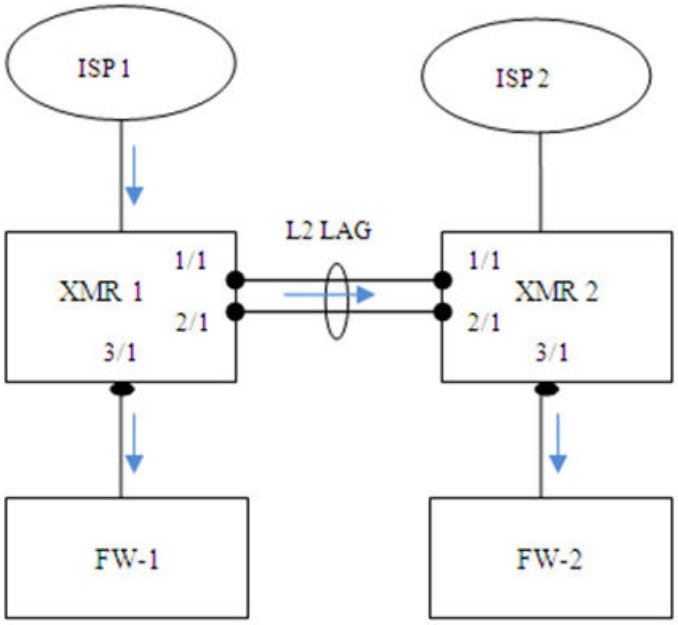
TABLE 23 Output of show vlan Ethernet

This field...	Displays...
Configured VLAN-Group entries	Number of VLAN groups that have been configured on the device.
Maximum VLAN-Group entries	Maximum number of VLAN groups that can be configured on the device.
VLAN-Group #	ID of the VLAN group
VLANs	VLANs that belong to the VLAN group.
Tagged ports:	Type and ID of the tagged ports that are members of the VLAN group

Multi-port static MAC address

The multi-port static MAC address feature enables you to send traffic destined for a particular MAC address on a set of ports of a VLAN instead of flooding the traffic on all ports of the VLAN. This feature enables you to configure a Layer 2 static multicast MAC address. This feature is supported on both the XMR Series, MLX Series and CER 2000 Series, CES 2000 Series series platforms.

FIGURE 16 Multi-port static MAC address example



Consider two XMR Series switches, XMR 1 and XMR 2, connected by a Layer 2 Link Aggregation Group (LAG), with member ports 1/1 and 2/1, as shown in Figure 16. To send traffic to a multicast MAC 'M' to the 3/1 ports on XMR 1 and XMR 2, you can create a static multicast MAC 'M' on the 3/1 ports and Layer 2 LAG (primary port 1/1 of Layer 2 LAG). Traffic sent to multicast MAC 'M' from ISP1 will be sent to ports 1/1 and 3/1 on XMR 1; and XMR 2 will send the traffic received on port 1/1 to FW-2, connected to port 3/1, instead of flooding the traffic on all ports of the VLAN.

Configuring multi-port static MAC address

To configure multi-port static MAC address, enter the following command at the VLAN configuration node level of the CLI. You must configure at least one port.

```
device# vlan 10
static-mac-address 0100.5e42.7f40 multi-ports ethe 1/1 to 1/3
ethe 2/1 to 2/5 priority 5
```

Syntax: [no] static-mac-address *mac-address* **multi-ports ethernet** [*slot1/port1* | [*slot1/port1* **to** *slot1/port#k*] **.. ethernet** [*slot#n/port#n* **to** *slot#n/port#m*] [**priority** *0-7*]

Limitations

The configuration of multi-port static MAC address has the following limitations:

- A maximum number of 400 multi-port MAC addresses and static MAC addresses can be configured in a system; however, the number of configured entries is limited by the number of multicast FIDs available in the system.
- FIDs with the same port mask are shared among multiple multi-port MAC addresses and multi-port ARP entries to conserve the number of multicast FIDs created in the system.
- Multi-port static MAC address cannot be configured on VLAN groups.

- You cannot add any of the interface MAC address as multi-port static MAC address.
- Multi-port static MAC address can be configured for either unicast or multicast addresses.
- Unicast MAC addresses configured as multi-port static MAC addresses will not be learned dynamically in the system or allowed to be dynamically moved to a different port.
- If the multi-port static MAC address being added already exists in the dynamic MAC table, then the dynamic MAC will be deleted and replaced with the configured multi-port static MAC.
- Trunk load-balancing is not supported. The multi-port static MAC address traffic will always be forwarded on the active primary port of the trunk port.
- The multi-port MAC feature is supported in a pure Layer 2 forwarding vlan to forward Layer 2 traffic to multiple ports. This feature should not be configured on a vlan with a virtual router-interface.

Error messages

Error messages are displayed in the following cases:

- You can configure multi-port static MAC addresses only if all the ports in the port list are members of the VLAN.

```
device(config-vlan-100)#static-mac 0001.0001.0003 multi-port eth 2/11
Error - Multiport mac cannot be configured, Port 2/11 is not member of vlan 100
```

- You must provide the complete port mask for deleting the multi-port static MAC address configuration.

```
device(config-vlan-100)#no static-mac 0001.0001.0002 multi-ports eth 2/19 to 2/20
Error - the port list does not match with the configured multiport mac port list
```

- On a trunk port, multi-port static MAC address configuration is allowed only on the primary port of the trunk.

```
device(config-vlan-100)#static-mac-address 0001.0001.0003 multi-ports ethe 2/19 to 2/20 ethe 4/3
Error - Multiport mac cannot be configured with non-primary trunk port  4/3
```

- A port with multi-port static MAC address configuration cannot be a secondary member of the trunk group.

```
device(config-lag-LAG1)#ports eth 4/11
Error: port 4/11 is part of multiport-mac and cannot be added as secondary port of a trunk
```

- When a LAG primary port is part of a multi-port static MAC address, the LAG cannot be undeployed or deleted. Also, when a non-LAG port is part of a multi-port static MAC address, you cannot deploy a LAG with that port. These configurations will be rejected. The following are the sample error messages for each of these cases:
  - Veto check for deploying LAG.

```
device(config-lag-LAG1)#deploy
Error: LAG LAG1 primary port 4/11 is configured as part of a multi-port mac entry, cannot deploy the LAG
```

- – Veto check for undeploying LAG.

```
device(config-lag-LAG2)#no deploy
Error: The primary port 4/15 is configured as part of a multi-port mac entry, cannot undeploy the LAG
```

- – Veto check for deleting LAG.

```
device(config)#no lag LAG2
Error: The primary port 4/15 is configured as part of a multi-port mac address, cannot remove the LAG
```

- A port belonging to a multi-port static MAC address is not allowed to be removed from a VLAN unless it is removed from all the multi-port static MAC addresses.

```
device(config-vlan-100)#no tag e 2/19
Error - port 2/19 is configured as part of a multi-port mac address, cannot remove port from vlan
```

- Module configuration deletion (no module) will be rejected if any of the ports in that module are configured as part of any multi-port static MAC addresses.

```
device(config)#no mod 4 ni-mlx-20-port-1g-100fx
Error - module 4 has ports that are member of the multi-port mac addresses, Cannot remove the module
```

Displaying multi-port static MAC address information

You can display the following information about multi-port static MAC addresses on the device:

- Running configuration
- Changes in the MAC table
- M-port debug information

- LP information

Displaying running configuration

To display the running configuration information of multi-port static MAC addresses, enter the following command.

```
device# show run
vlan 10
tagged ethe 1/1 to 1/20 ethe 2/1 to 2/48
static-mac-address 0100.5e42.7f40 multi-ports ethe 1/1 to 1/3
ethe 2/1 to 2/5 prority 5
```

Displaying changes in the MAC table

You can display the complete MAC table, a specific entry in the MAC table, or a specific MAC entry with M-port details.

To display the complete MAC table, enter the following command.

```
device# show mac
MAC Address         Port          Age          VLAN    Type
0100.5e42.7f40          1/1...         Static           10
Ports  : e 1/1 to 1/3 e 2/1 to 2/5
0000.999d.9996      2/20             Static        10
```

To display a specific MAC address from the MAC table, enter the following command.

```
device# show mac 0100.5e42.7f40
MAC Address         Port          Age          VLAN    Type
0100.5e42.7f40          1/1...       Static       10
Ports: e 1/1 to 1/3 e 2/1 to 2/5
```

To display a specific MAC address with M-port details, enter the following command.

```
device#show mac 0100.5e42.7f40 debug
MAC Address         Port          Age          VLAN    Type
0100.5e42.7f40          1/1...           Static       10
Mport: 30324 FID: 0x00008006 Ports: e 1/1 to 1/3 e 2/1 to 2/5
```

SA and DA learning and aging

Static MAC addresses and multi-port MAC addresses can always be programmed in the CAM of all LP modules with valid VLAN membership. Multi-port static MAC addresses are added or removed from the CAM when a port is added to or deleted from a VLAN. These addresses cannot be dynamically learned or moved to a different port. These addresses will not be removed from the hardware unless the user deletes the multi-port static MAC addresses.

MP switchover and hitless upgrade

The Multi-port static MAC Address feature supports MP switchover and hitless upgrade.

The static MAC table is maintained on both active and standby MPs. The static MAC table is synched to standby MP by CLI configuration commands. The M-port table and FID are also synched from the active MP to the standby MP. After MP switchover, M-port MAC addresses are associated with the FID and, in the case of a missing FID, a new FID will be created and programmed for the multi-port static MAC address.

The static MAC table and FID table will be reprogrammed after LP reload.

Flooding features

User-configured multi-port static MAC addresses will always be programmed on DA CAMs in all PPCR CAMs. So, traffic with this DA MAC address will never be flooded in the VLAN, even when flooding features like transparent VLAN, unknown unicast flooding, multicast flooding, and CPU protection are configured on the system.

ESI overview

An Ethernet Service Instance (ESI) is a provisioning environment for defining VLAN and other layer 2 parameters for creating services, typically across a carrier network.

In a local area network a total of 4K VLANs can be configured across the entire network domain. With a Q-in-Q bridging, VLANs from the set of 4K VLANs can be inter-connected across a provider network. While ESI allows a carrier to provide transport services for different sets of 4K VLANs for different customers, the provider network is still limited to using 4K VLANs across all of the customers connected to a single box, as it is very difficult to configure and manage different sets of 4K VLANs across the different ports within a single system.

Using an ESI, a carrier can create service instances that hold one or more VLANs. Each instance has an alphanumeric name that is locally unique. The purpose of creating an instance is to provide a container to hold VLANs and other layer2 parameters that define properties of all of the elements contained within the instance.

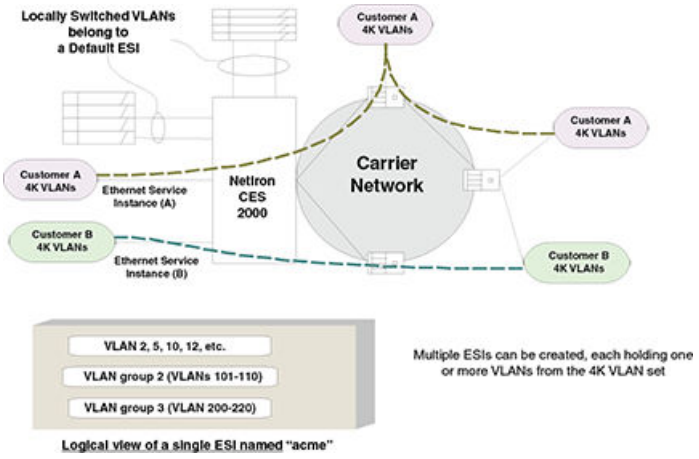
VLANs are added to an ESI using a standard VLAN command for individual VLANs or a VLAN group command for adding a set of VLANs.

In a simplified network shown in ESI overview, customers A and B are connected to a CES 2000 Series device with each customer having a separate set of 4K VLANs. One or more ESIs are created to hold these 4K VLANs.

NOTE

Although theoretically it is possible to add sets of 4K VLANs in ESIs, the actual number of VLANs in an ESI is limited by the use of memory and hardware resources.

Ethernet Service Instance for VLAN configuration



Once an ESI is defined, CES 2000 Series and CER 2000 Series devices operate on rules for configuring VLANs inside an ESI, and check against configuration incompatibilities (such as configuring the same VLAN value from two different ESIs on the same port).

Types of ESI

There are two types of Ethernet Service Instances, as described:

Default ESI

In ESI overview on page 135, VLANs associated with ports in the top left corner of the CES 2000 Series and CER 2000 Series devices aren't being transported over to the carrier network - these VLANs are being locally switched and connected with switches in the local area network. The CES 2000 Series and CER 2000 Series devices support 4K VLANs of this type, without any ESI configured. Internally, these VLANs are associated with a Default ESI, and are referred to as 'Regular VLANs'.

NOTE
On platforms that support the ESI framework: FDP and CDP may be configured in the default ESI. FDP and CDP are not supported under user-defined ESIs.

Customer ESI

ESIs that are configured to hold customer VLANs that need to be transported across a carrier network are usually referred to as customer ESIs. A customer ESI always has a Layer 2 protocol VLAN encapsulation applied to all VLANs in the ESI.

In a carrier network, an incoming customer VLAN packet will usually be configured with successive encapsulations, such as with service VLANs, in-service identifiers, or backbone VLANs. Each encapsulation is associated with a different ESI. To define the encapsulation hierarchy, an ESI for an incoming packet is defined as a client of the ESI for the next encapsulation. The next encapsulation ESI is referred to as a 'provider ESI' and the ESIs that are declared as client ESIs are referred to as 'client ESIs'.

Depending on their association, customer ESIs can be one of the three types:

- Standalone ESI - An ESI that is not linked to any other ESI, and are used only to hold VLANs and define their properties.
- Provider ESI - An ESI with one VLAN, and one or more client ESIs, each holding one or more VLANs.
- Client ESI - An ESI that is defined to be a client of another ESI, and that can have one or more VLANs defined inside it.

Configuration considerations

The following rules apply for CLI operations for Provider Bridge (PB) and related protocols:

- To prevent topology changes at startup, it is recommended that you not use the same ESI-Vlan ID as the Default-Vlan ID.
- An ESI is created for each service (such as a customer CVLANs, SVLANs, PBB, etc.).
- All attributes for an ESI - such as VLAN, port binding, encapsulation, etc., are defined inside an ESI.
- To prevent configuration errors, no parameter overrides are permitted outside of an ESI.
- ESIs can be nested to provide multiple protocol encapsulations for a packet. Restrictions on bindings can be present depending on the actual platform. When nesting is used, inner ESIs are called client ESIs.
- A given VLAN means CVLAN or SVLAN, depending on the encapsulation definition for the ESI:
  - If no encapsulation is defined as "cvlan" the VLAN refers to CVLAN.

  NOTE
  For ESI VLANs, It is mandatory to define an encapsulation.

- - If encapsulation is "svlan", the VLAN refers to SVLAN (PB).
  - If encapsulation is "bvlan", the VLAN refers to B-VLAN (PBB).

- ISID values are treated differently from other VLANs, since ISID parameters have no networking association and are used only for mapping different SVLANs into service identifiers.

Creating an ESI

Create an ESI by naming it and specifying encapsulation type for all the VLANs inside it.

Creating an ESI with CVLAN tagging

To create an ESI with CVLAN tagging named "acme", enter a command such as the following.

```
device (config)# esi acme encapsulation cvlan
```

Syntax: [no] esi *esi-name* **encapsulation cvlan** | **svlan** | **isid** | **bvlan**

Use the **cvlan** parameter to specify the encapsulated customer VLAN (CVLAN).

Use the **svlan** parameter to specify the encapsulated service VLAN (SVLAN).

Use the **isid** parameter to specify the encapsulation for the mapping of SVLANs into service identifiers.

Use the **bvlan** parameter to specify the encapsulated backbone VLAN (BVLAN).

Once an ESI is created, subsequent invocations of the ESI do not require the encapsulation parameter.

Defining CVLANs inside the ESI

To define CVLANs inside the ESI, enter a command such as the following.

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

Configuring the CVLAN to be tagged

To configure CVLAN 10 to be tagged on port 1/1, enter a command such as the following.

```
device(config-esi-acme-vlan-10)# tagged ethernet 1/1
```

Show VLAN commands

The following **show** commands will display custom ESI configurations for VLANs.

Displaying information for a VLAN inside an ESI

To display a VLAN inside an ESI, enter a command such as the following.

```
device(config)#show esi acme vlan 10
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
VLAN 10 details:
----------------
PORT-VLAN 10, Name [None], Priority Level-,Priority Force 0
L2 protocols   : NONE
ESI: bay Encapsulation: cvlan
----------------
No ports associated with VLAN
Arp Inspection: 0
```

```
DHCP Snooping: 0
L2 protocol forwarding mode:Tunnel
Flood domain ID 4176
```

Syntax: show vlan *num*

Displaying information for a VLAN inside an ESI in brief format

The **show vlan brief** command displays VLANs in a tabular format for compactness. This command may be executed from any CLI level.

```
device#show vlan brief
Configured PORT-VLAN entries: 1
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1
VLAN     Name       Encap ESI                              Pri Ports
----     ----       ----- ---                              --- -----
10       [None]     cvlan acme                             -
```

Syntax: show vlan brief

Displaying a single ESI

To display details of a single ESI, enter the following command from any level of the CLI.

```
device#show esi acme
ESI Name         Encap    Number of  Provider         Provider   Provider Client
                          Members    ESI              Encap      VLAN     ESIs
--------         -----    ---------  --------         -----      -------- ------
acme             cvlan    1                                             0
VLAN(s) at this ESI:
-------------------
VLAN     Name       Pri [L2 Protocols]       Ports
10       [None]     cvlan acme                             -   NONE
```

Displaying all ESIs

Use the **show esi** command to display a list of all the ESIs configured in the system. This command can be used at any level of the CLI.

```
device(config)#show esi
ESI Name         Encap    Number of  Provider         Provider   Provider Client
                          Members    ESI              Encap      VLAN     ESIs
--------         -----    ---------  --------         -----      -------- ------
acme             cvlan    1                                             0
```

Syntax: show esi *name*

Tag-type configuration

For the CES 2000 Series and CER 2000 Series, the following two VLAN tag-types are allowed that can be configured globally:

- **tag1** applies to customer edge ports (CVLAN) by default.
- **tag2** applies to provider-network, backbone-edge, and backbone-network port types (SVLAN and BVLAN) by default.

  NOTE
  The **tag1** and **tag2** are independent of port-types, so the system can be configured to use **tag1** for SVLAN, BVLAN and **tag2** for CVLAN.

Configuring tag-types

You can set the ISID value using a separate command similar to the XMR Series and MLX Series command as shown below.

Syntax: [no] tag-value isid *num*

You can configure CVLAN, SVLAN, and BVLAN tag-types as shown below.

```
device(config)# tag-value tag1 8100
device(config)# tag-value tag2 9100
device(config)# tag-type cvlan tag1 svlan tag2 bvlan tag2
```

Syntax: [no] tag-value *num*

Syntax: tag-type *tag-n*

The *num* parameter specifies the value assigned to the tag. The default value for **tag1** is 0x8100 and for **tag2** is 0x88a8.

The *tag-n* parameter can be either **tag1** or **tag2** .

Tag type can be changed from a default value to a specific port as shown in the following example.

```
device(config-if-e1000-1/1)# tag-type tag2 ethernet 1/1
device(config-if-e1000-1/1)# tag-type tag1 ethernet 1/2
```

Syntax: tag-type *tagid* **ethernet** *interface_id*

The *tagid* parameter can be either **tag1** or **tag2** . Possible tagid values are:

- – isid - to set the isid tag-value
  – tag1 - to set the tag-type tag1 value
  – tag2- to set the tag-type tag2 value

The *interface_id* parameter specifies the Ethernet slot and port ID.

Restrictions

The tag-type has the following restrictions:

- CVLAN and SVLAN cannot have the same tag-type but the tag-value can be set to the same.
- SVLAN and BVLAN must have the same tag-type.
- Port-type must be set to the default to configure the port-level tag-type.

Displaying tag types

To display the different tag types, enter a command such as the following.

```
device(config)#show tag-type
Encap      Current VLAN Tags             Default VLAN Tags
-----      -----------------             -----------------
cvlan                 8100                       8100
svlan                 9100                       88A8
isid                  86B5                       88E7
bvlan                 9100                       88A8
```

NOTE

The CES 2000 Series and CER 2000 Series require that the SVLAN and BVLAN tag types be same. The default tag-type for bvlan and svlan is 0x88A8 for the CES 2000 Series and 0x8100 for MLX Series devices. The BVLAN, SVLAN, or CVLAN cannot be configured separately on the MLX Series device as is done on the CES 2000 Series.

Application of a standalone ESI

You can use a standalone ESI to perform VLAN ID translation. For example:

```
device(config)# vlan 5
device(config-vlan-5)#tag eth 1/1
device(config-vlan-5)#exit
device(config)#vlan 6
device(config-vlan-6)#tag eth 1/2
device(config)#vlan 7
device(config-vlan-7)#tag eth 1/3
```

Flood domain and VLAN translation

An ESI consisting of VLANs, can optionally be set up as a flood domain, serving two purposes:

- **Flooding** - Creates a domain where packets received on a port within the flood domain are sent to all other ports in the group with proper VLAN translations.
- **VLAN translation -** This feature can be used for translating between SVLANs across a provider boundary.

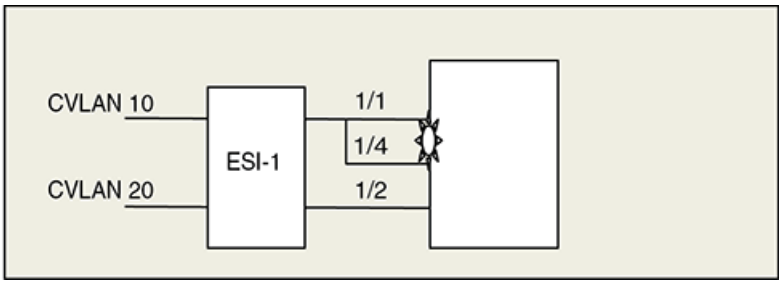
  NOTE
  While the flood domain includes multiple VLANs, spanning tree still works within the scope of each VLAN separately. Therefore, loops spanning across VLANs will not get resolved.

  NOTE
  Multiple VLANs on same port cannot be added in a single flood domain ESI. However multiple VLANs on different ports are allowed to be added.

System operation without flood domain

Figure 18 shows an ESI configuration with a single flood domain.

FIGURE 17 Single flood domain ESI



The following CLI commands create the scenario shown in Figure 18.

```
device(config)# esi ESI_1 encapsulation cvlan
device(config-esi-ESI_1)# vlan 10
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/1
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/4
device(config-esi-ESI_1-vlan-10)# vlan 20
device(config-esi-ESI_1-vlan-20)# tagged ethernet 1/2
```

System operation without a single flood domain

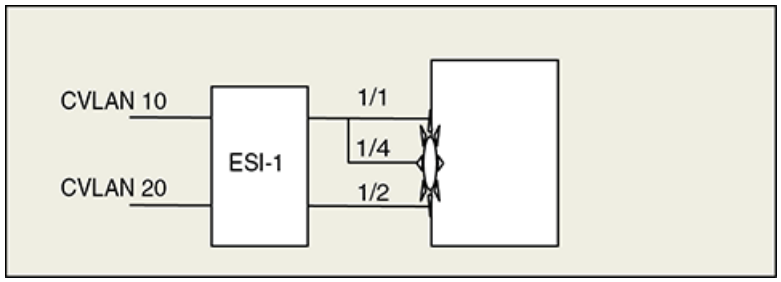
Without a single flood domain configuration, the system operates in the following manner:

- A packet received on 1/1 is sent out on 1/2 with a CVLAN mapping of 20.
- A packet received on 1/2 is sent out on 1/4 with a CVLAN mapping of 10.
- A packet received on 1/4 with a CVLAN mapping of 10 is sent to 1/2 with a CVLAN mapping of 20.

Configuring a flood domain with VLAN translation

You can create a flood domain inside an ESI using the **single-flood-domain** command. A VLAN within an ESI normally defines a flood domain, but when single flood domain is configured, all the VLANs in that ESI become part of one flood domain. In this case every broadcast packet or unknown unicast packet is flooded in all the VLANs in the ESI. When a A C-ESI or S-ESI are configured for single flood domain, they cannot be coupled together.

FIGURE 18 Flood domain



CVLAN translation

Configuring a flood domain with VLAN translation on page 141 shows a configuration for a C-ESI. This combines VLAN 10 and VLAN 20 into one flooding domain.

With this configuration:

- Packets received on 1/1 are sent out on 1/2 with a CVLAN mapping of 20.
- Packets received on 1/2 are sent out on 1/4 with a CVLAN mapping of 10.
- Packets received on 1/4 with a CVLAN mapping of 10 are sent to 1/2 with a CVLAN mapping of 20.

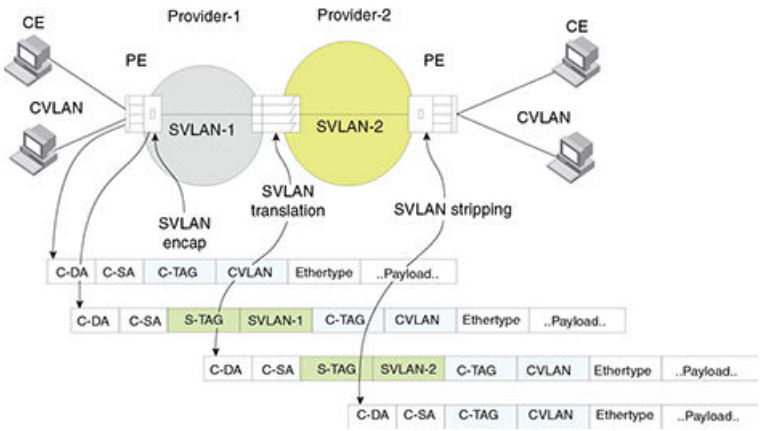
```
device(config)# esi ESI_1 encapsulation cvlan
device(config-esi-ESI_1)# single flood domain
device(config-esi-ESI_1)# vlan 10
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/1
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/4
device(config-esi-ESI_1)# vlan 20
device(config-esi-ESI_1-vlan-20)# tagged ethernet 1/2
```

About IEEE 802.1ad

In a Provider Bridge (PB) network, a provider VLAN is called a Service VLAN (SVLAN), and a customer VLAN is called a Customer VLAN (CVLAN). A CVLAN carries a default tag-type of 0x8100. The range of customer VLANs (CVLANs) can be mapped to an SVLAN, allowing a CVLAN to cross a provider boundary. The SVLAN can be configured to provide service, tunnels, or broadcast domains. The SVLAN and the CVLAN are sent in the same packet so that customer packets with VLAN information are carried to the customer network on the other side.

A Provider Edge (PE) device receives packets with no tags, or packets with CVLAN information, and adds an SVLAN field on the packet before sending to the provider network. The device can be configured to perform SVLAN translation at an inter-provider boundary. Figure 20 provides an example of a PB network.

FIGURE 19 IEEE 802.1ad network



The CVLAN carries a default tag-type of 0x8100. SVLAN encapsulation is similar to CVLAN but with a different tag type (default 0x88a8). A customer's 4K CVLAN domain can be mapped to an SVLAN, allowing the customer VLAN domain to cross a provider boundary. The SVLAN can be configured to provide services, tunnels or broadcast domains.

At an inter-provider boundary, if necessary, the SVLAN value inserted by the first provider may be replaced by a different SVLAN value (this is referred to as SVLAN translation).

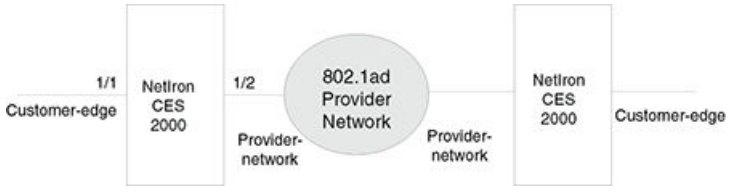
IEEE 802.1ad Provider Bridging limitations

The following provider bridging limitations apply to PB networks:

- An SVLAN can have a value between 1- 4090
- An SVLAN limit of 4K VLANs is typically inadequate in the carrier space.
- As with normal VLAN devices, every PB node must learn all customer MAC addresses, even with SVLAN encapsulation.

Port type configuration for Provider Bridging (PB)

FIGURE 20 Port types in a Provider Bridge (PB)



The CES 2000 Series defines two types of ports for operation in a provider network:

- **Customer-edge** : This port receives packets with CVLAN tagging. These packets are either switched to other customer-edge ports locally, or are encapsulated with SVLAN tags and are sent out on the provider network ports.

- **Provider-network** : This port receives packets with SVLAN tagging, and transmits packets with SVLAN tagging.

There are two additional port types that are defined for the CES 2000 Series: these are **backbone-edge** and **backbone-network** . These port types are defined for IEEE 802.1ah Provider Backbone Bridging (PBB).

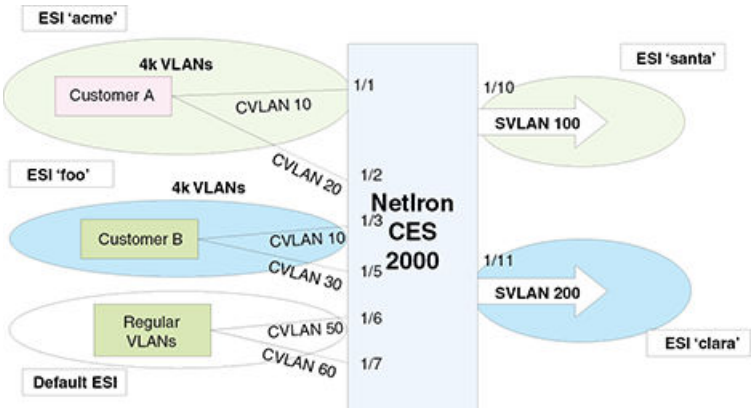
IEEE 802.1ad network configuration example

Configuring a network for IEEE 802.1ad requires the following steps.

1. Configure appropriate port types.
2. Define tag values for CVLAN and SVLAN
3. Define ESIs for CVLAN side and bind VLANs and ports.
4. Define ESIs for SVLAN side and bind VLANs and ports.

Sample configuration

FIGURE 21 IEEE 802.1ad network with ESI definitions



The network architecture for IEEE 802.1ad in Figure 22 shows customer A with two tagged CVLAN ports connected to SVLAN 100, and customer B with two CVLANs connected to SVLAN 200.

To define these configurations and associate them, ESIs are created for each of the configurations. For example, configurations for customer 'A' are defined inside an ESI 'acme' and the carrier-side encapsulation with SVLAN 100 are defined inside an ESI 'santa'.

Once the two ESIs are defined, ESI 'acme' is associated to 'santa' by specifying 'acme' as a Client ESI inside the ESI 'santa'. A similar operation is done for associating customer-side ESI 'foo' with provider-side ESI 'clara' for the customer 'B'.

Configuration steps

The following steps show an example on how to configure an IEEE802.1ad network.

Configure port types for interfaces

Before CVLAN or SVLANs can be provisioned for an interface, the port-type for the interface must be appropriately defined.

The **port-type** command defines a port type for an Ethernet interface. The port-types specify both sides of IEEE 802.1ad and IEEE 802.1ah networks. Enter a command such as the following to set 1/10 and 1/11 to the provider-network port type.

```
device(config)# interface ethernet 1/10
device(config-if-e10000-1/10)# port-type provider-network
device(config-if-e10000-1/10)# exit
```

Syntax: [no] port-type [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the backbone edge port for IEEE 802.1ah PBB.

Use the **backbone-network** parameter to specify the backbone network port for IEEE 802.1ah PBB.

Use the **customer-edge** parameter to specify the customer edge port for IEEE 802.1ad PB.

Use the **provider-network** parameter to specify the provider network port IEEE 802.1ad PB.

Configuring port type values

Four port-type values are specified. For our example, set 1/10 and 1/11 to provider-network port type.

```
device(config)# interface ethernet 1/10
device(config-if-e10000-1/10)# port-type provider-network
device(config-if-e10000-1/10)# exit
```

Use the following commands to set 1/11 to provider-network port type.

```
device(config)#interface ethernet 1/11
device(config-if-e10000-1/11)# port-type provider-network
device(config-if-e10000-1/11)# exit
```

Use the following commands to set 1/1 to customer-edge port type.

```
device(config)#interface ethernet 1/1
device(config-if-e10000-1/1)# port-type customer-edge
device(config-if-e10000-1/1)# exit
```

Syntax: [no] port-type [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the backbone edge port for IEEE 802.1ah PBB.

Use the **backbone-network** parameter to specify the backbone network port for IEEE 802.1ah PBB.

Use the **customer-edge** parameter to specify the customer edge port for IEEE 802.1ad PB.

Use the **provider-network** parameter to specify the provider network port IEEE 802.1ad PB.

Displaying the port type

The **show interfaces** command displays port-type for an interface, as shown below.

```
device(config-if-e10000-1/2)# show interfaces ethernet 1/10
GigabitEthernet1/10 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 0e04.80de.ada0 (bia 0e04.80de.ada0)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Member of VLAN 4096 (untagged), port is in untagged mode, port state is Disabled
  STP configured to ON, Priority is level0, flow control enabled
  arp-inspection-trust configured to OFF
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  Port-type (802.1ad/802.1ah): provider-network
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
```

```
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
NP received 0 packets, Sent to TM 0 packets
NP Ingress dropped 0 packets
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions
NP transmitted 0 packets, Received from TM 0 packets
```

TABLE 24 show interfaces command output

This field...	Displays...
Module type Port# is *State*	The *module type* variable specifies a type of interface module, such as 10GigabitEthernet. The *port#* variable specifies the port number for the interface module. The *state* variable if the interface module is up or down.
Line protocol is *status*	The *status* variable specifies if the line protocol is up or down. If the interface is down due to Link Fault Signaling - Remote Fault Notification (LFS or RFN) the reason is indicated as: "(remote fault)".
STP Root Guard is *status*	The *status* variable specifies if the STP Root Guard is enabled or disabled.
STP BPDU Guard is *status*	The *status* variable specifies if the STP BPDU Guard is enabled or disabled.
Hardware is *module type*	The *module type* variable specifies a type of interface module, such as # Gigabit Ethernet.
Address is *MAC- address*	The *MAC- address* variable specifies the MAC address of the port.
Configured speed and actual speed	The speed that the module has been configured to operate at, and the actual speed it is currently operating at.
Configured port speed and actual duplex	The port capacity that the module has been configured to operate at, and the actual speed it is currently operating at.
Member of *VLAN #* (untagged) *port#* L2 VLANS (tagged) Port is in dual mode/untagged/tagged mode Port state is *status*	The *VLAN#* (untagged) variable specifies a port that is a member of only 1 VLAN. The *port#* L2 VLANS (tagged) variable specifies a port that is a member of multiple ports and untagged. A port is in *dual- mode* specifies member VLAN ports as untagged and tagged. The default mode is dual-mode. The *status* variable identifies the flow of traffic as forwarding or disabled.
STP configured to *status* Priority level Flow control *status*	The *status* variable specifies if the STP is ON or OFF. The priority level assigned to the port-based VLAN. The priority level is on scale from 0-7. The default is 0. The *status* variable is enabled or disabled.
Priority force *status*	The *status* variable specifies if the priority force on a port is disabled on enabled.
Drop precedence level *value*	Identifies the TOS or DSCP value in the IPv4 or IPv6 packet header. The *value* variable specifies the drop precedence on a scale from 0-3. Packets that contain a DSCP value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped. The default value is 0.
Drop precedence force *status*	The *status* variable specifies the drop precedence force as enabled or disabled. Identifies the drop precedence if the force command is configured for a specific ingress port.

TABLE 24 show interfaces command output (continued)

This field...	Displays...
arp-inspection-trust configured to *status*	The *status* variable specifies if arp-inspection-trust feature is configured ON or OFF. The default trust setting for a port is untrusted.
Mirror *status*	The *status* variable specifies if the port mirror command is configured as enabled or disabled.
Monitor *status*	The *status* variable specifies if the port monitor command is configured as enabled or disabled.
Trunk membership	The *Trunk membership* variable identifies the interface module as a member of a primary or secondary port. This specifies members of an active port or not a member of an active port.
Configured trunk membership	The *Configured trunk membership* variable identifies the interface module as a member of any configured trunk or not a member of a configured trunk.
Port name	The *port name* variable identifies the name assigned to the port.
MTU *# bytes* , encapsulation ethernet	Maximum Transmission Unit (MTU) refers to the size of the largest packet or frame that a known layer can pass forward. The *# bytes* variable refers to size of the packet or frame.
#seconds input rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* input rate refers to: • The *value* of bits received per second. • The *value* of packets received per second. • The *%* utilization specifies the port's bandwidth used by received traffic.
# seconds output rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* output rate refers to: • The *value* of bits transmitted per second. • The *value* of packets transmitted per second. • The *%* utilization specifies the port's bandwidth used by transmitted traffic.
value packets input, *value* bytes	• The *value* variable specifies the number of packets received. • The *value* variable specifies the number of bytes received.
Received *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module is receiving on broadcasts, multicasts, and unicast traffic.
value input errors, *value* CRC, *value* frame, *value* ignored	• The *value* variable specifies the number of received packets with errors. • The *value* variable specifies the number of received packets with CRC errors. • The *value* variable specifies the number of received packets with alignment errors. • The *value* variable specifies the number of received packets that are discarded.
value runts, *value* giants	The *value* runts variable specifies the number of small packets that are less than 64 bytes. The *value* giants variable specifies the number of large packets greater than 1518 bytes.
NP received	The number of packets received on the Network Processor (NP).
NP transmitted	The number of packets sent from the Network Processor to the Traffic Manager (TM).
NP Ingress dropped	The number of ingress packets dropped on the Network Processor.
value packets output	• The *value* variable specifies the number of transmitted packets. • The *value* variable specifies the number of transmitted bytes.

TABLE 24 show interfaces command output (continued)

This field...	Displays...
value bytes	
Transmitted *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module transmitted on broadcasts, multicasts, and unicast traffic.
value output errors, *value* collisions	• The *value* variable specifies the number of transmitted packets with errors. • The *value* variable specifies the number of transmitted packets with collision errors.
Network Processor transmitted *value* packets Received from Traffic Manager *value* packets	The *value* variable specifies the number of packets transmitted from the Network Processor. The *value* variable specifies the number of packets received by the Network Processor from the Traffic Manager.

ESIs that are associated to a provider ESI are called client ESIs, and packet encapsulation order follows from client ESI to provider ESI.

The ESI concept as defined here can be used for defining and associating all types of services such as IEEE 802.1ad and IEEE 802.1ah.

Creating an ESI

An ESI is configured by giving a name for the ESI and specifying encapsulation type for all the VLANs inside it.

Creating an ESI with CVLAN tagging

To create an ESI with CVLAN tagging named acme, enter a command such as the following.

```
device(config)# esi acme encapsulation cvlan
```

Syntax: [no] esi *esi-name* **encapsulation cvlan** | **svlan** | **isid** | **bvlan**

Use the **cvlan** parameter to specify the encapsulated Customer VLAN (CVLAN).

Use the **svlan** parameter to specify the encapsulated Service VLAN (SVLAN).

Use the **isid** parameter to specify the encapsulated the mapping of different SVLANs into service identifiers.

Use the **bvlan** parameter to specify the encapsulated Backbone VLAN (BVLAN).

Once an ESI is created, subsequent invocations of the ESI do not require encapsulation parameter.

Steps for provisioning a PB network

1. Create ESI for the customer side.
2. Create an ESI and define one or more CVLANs inside it. In the following command acme is the ESI name.

   ```
   device(config)# esi acme encapsulation cvlan
   ```

3. Define the CVLANs inside the ESI.

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

4. In the following command, CVLAN 10 becomes tagged on port 1/1

```
device(config-esi-acme-vlan-10)# tagged ethernet 1/1
```

5. In the following command, CVLAN 20 becomes tagged on port 1/2

```
device(config-esi-acme)# vlan 20
device(config-esi-acme-vlan-20)# tagged ethernet 1/2
device(config-esi-acme-vlan-20)# exit
```

Create an ESI on provider side

1. Configure santa as the name of the provider IEEE 802.1ad service

```
device(config)# esi santa encapsulation svlan
```

2. Define SVLAN 100 inside this ESI

```
device(config-esi-acme-iptv)# vlan 100
```

3. Associate physical ports to the VLAN

```
device (config-esi-acme-iptv-vlan-100)# tagged ethernet 1/10
```

Display ESI configuration

At this point, all ESIs have been set up. Enter the **show esi** command to display ESIs.

```
device(config)# show esi
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
acme            cvlan    2                                           0
foo             cvlan    2                                           0
santa           svlan    1                                           0
clara           svlan    1                                           0
```

In this display, ESIs are shown with their encapsulations and number of members (VLANs) in each ESI. At this stage, there are no client-provider bindings.

Associate customer ESI with provider ESI

Now that ESIs are defined for both the customer and provider sides, you can bind the customer ESI to the provider ESI using the **esi-client** command for the ESI named acme.

To complete the configuration for a IEEE 802.1ad network as shown in Sample configuration on page 143, associate the customer ESI to the provider ESI.

```
device (config)# esi santa
device (config-esi-santa)# esi-client acme
device(config-esi-santa)# exit
```

Show ESI command for the final configuration

Enter the **show esi** command to display the final configuration.

```
device(config)# show esi
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
```

```
                    Members    ESI        Encap      VLAN       ESIs
--------            -----    ---------  --------   -----      --------   ------
acme                cvlan    2          santa        svlan      100        0
foo                 cvlan    2          clara        svlan      200        0
santa               svlan    1                                             1
clara               svlan    1                                             1
```

In this display, the ESI named acme is an ESI of the encapsulation type CVLAN, which has two members (VLANs). The acme ESI now has the santa ESI as a provider with an SVLAN encapsulation type. The santa provider ESI is configured with VLAN ID 100.

This means that both CVLANs in the acme ESI receive a second SVLAN encapsulation (with a tag-type value of SVLAN as globally configured) and a SVLAN ID of 100.

PB using untagged members

For the configuration shown in Sample configuration on page 143, you added a Customer ESI and a Provider ESI and bound them together to provide PB service. This generic configuration can be used when a particular port is shared between customers. However, if a port is completely owned by one customer, you can use the following simple configuration to provide PB service.

Sample configuration 2

1. In this case, customer port 1/1 maps the customer's 4K VLANs. The provider side port 1/2 uses SVLAN 100. All customer traffic *from* the provider side is appended with an SVLAN 100 tag, and for all traffic going *to* the provider side will have the SVLAN 100 tag removed.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# port-type provider-network
```

2. Configure customer ports as provider network ports instead of customer edge ports.

```
device(config)# interface ethernet 1/2
device (config-if-e1000-1/1)# port-type provider-network
```

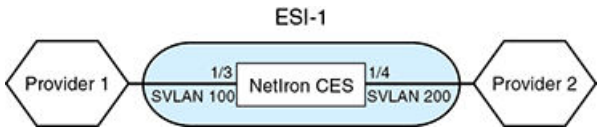
3. Add customer ports as untagged ports so that any traffic tagged with the 8100 tag will also be treated as untagged and will be appended with SVLAN 100.

```
device(config)# esi ESI_1
device(config)# esi ESI_1 encapsulation svlan
device(config-esi-ESI_1)# vlan 100
device(config-esi-ESI_1)# tagged ethernet 1/2
device(config-esi-ESI_1)# untagged ethernet 1/1
device(config-esi-ESI_1)# exit
```

SVLAN translation using flood domain configuration

The **single-flood-domain** command is used for SVLAN translation across provider domains.

FIGURE 22 SVLAN translation at an inter-provider boundary



In the configuration below, packets on port 1/3 with SVLAN=100 are translated to the port 1/4 with SVLAN=200 as the ports are in the same flood domain.

```
device(config)# esi ESI_1 encapsulation svlan
device(config-esi-ESI_1)# single-flood-domain
device(config-esi-ESI_1)# vlan 100
device(config-esi-ESI_1)# tagged ethernet 1/3
device(config-esi-ESI_1)# vlan 200
device(config-esi-ESI_1)# tagged ethernet 1/4
device(config-esi-ESI_1)# exit
```

Untagged ports

Packets from SVLAN port with only SVLAN tag (no CVLAN tag) are sent to only untagged ports in the default ESI.

Default VLAN

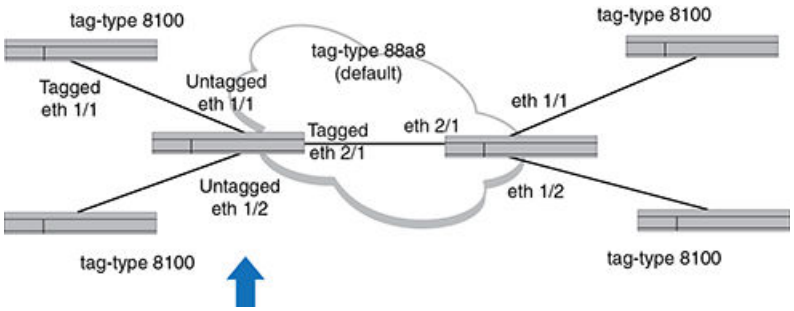
Packets from SVLAN with default VLAN as CVLAN tag are sent to untagged ports in the default ESI.

No Default VLANs are allowed inside non-default ESIs.

Port-based Service Interface Super AggregatedVLANs (SAV)

Using Port-based Service Interfaces is equivalent to using SAV in other Extreme products. Port-based Service Interfaces can be used when mapping a port based interface to a single Service VLAN Tag (S-TAG). When using a port-based service interface, it maps all VLAN-IDs from incoming ports to a SVLAN as shown in Port-based Service Interface Super AggregatedVLANs (SAV).

Port-based service interface



Sample configuration

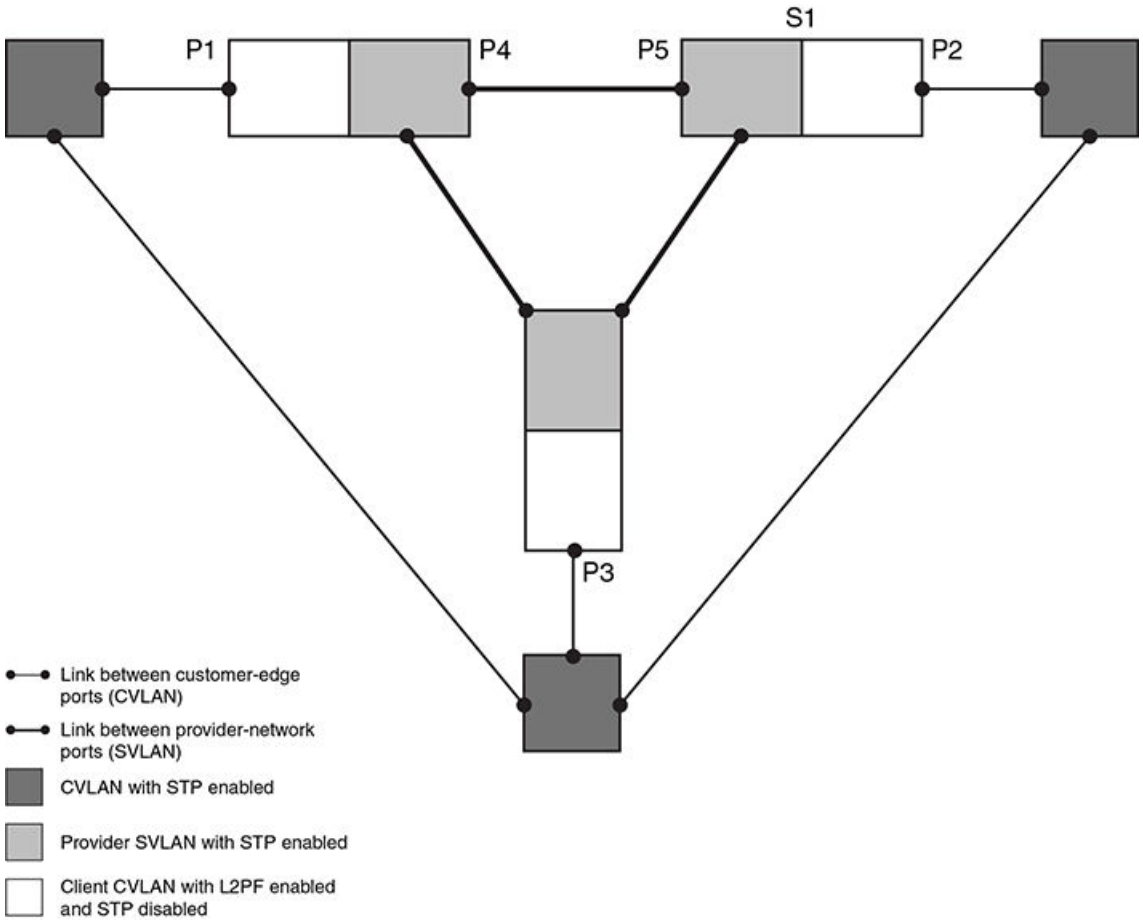
```
device(config)# tag-type tag2 ethernet 2/1
device(config)# vlan 100
device(config-vlan-100)# tagged ethernet 2/1
device(config-vlan-100)# untagged ethernet 1/1
device(config)# vlan 200
device(config-vlan-200)# tagged ethernet 2/1
device(config-vlan-200)#untagged ethernet 1/2
```

Layer 2 Protocol Forwarding (L2PF)

Layer 2 Protocol Forwarding (L2PF) is configured on CVLANs. You can configure the system to forward BPDUs on all CVLANs coming at different edge ports, drop all BPDUs, or selectively enable forwarding on a few CVLANs.

L2PF can transparently extend the STP topology of the CVLAN domain through the SVLAN domain, as if the CVLAN switches were directly hooked together. L2PF can be applied to CVLAN, which is a client of provider SVLAN as shown in "L2PF in a network".

FIGURE 23 L2PF in a network

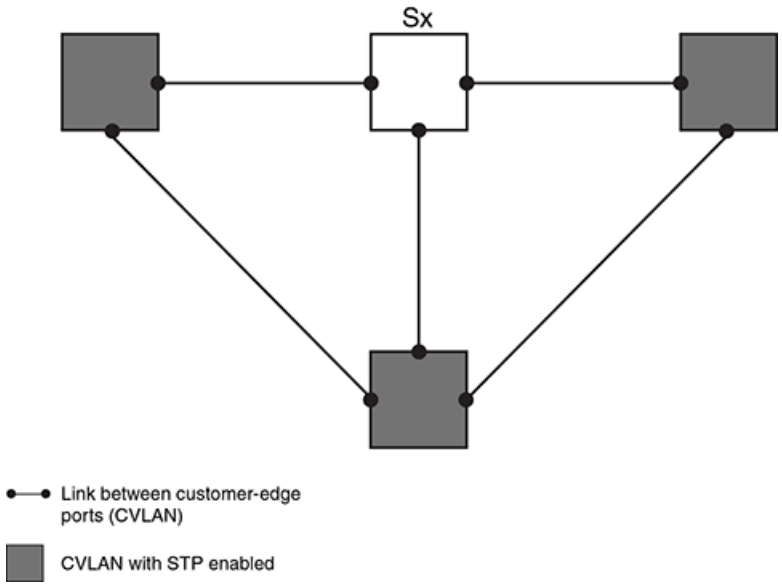


In this network, a CVLAN BPDU ingressing port P1 gets encapsulated and forwarded through the SVLAN domain as an ordinary multicast frame and then it egresses ports P2 and P3. In the SVLAN domain, a SVLAN BPDU originating from P4 and the ingressing port P5 gets terminated and processed in switch S1 for STP calculation. In other words, the final STP topology in the SVLAN domain is completely independent from the final STP topology in the CVLAN domain.

From the CVLAN domain's point of view, the physical network appears as shown in Figure 25.

Below, Sx is a simple bridge without STP.

FIGURE 24 Physical network



NOTE
It is critical for L2PF to disable STP on the client CVLAN to operate in that CVLAN.

STP wins over L2PF when both are enabled on a given client CVLAN. Table 25 displays the Port configuration for IEEE 802.1ah and IEEE 802.1ad.

TABLE 25 Port Configuration

Description	L2PF	STPF
CVLAN BPDU flooded inside CVLAN just like any multicast frame (but not flooded to SVLAN)	Disabled	Disabled
CVLAN BPDU terminated and processed in client CVLAN	Disabled	Enabled
CVLAN BPDU tunneled to provider SVLAN, and flooded in provider SVLAN	Enabled	Disabled
CVLAN BPDU terminated and processed in client CVLAN	Enabled	Enabled

NOTE
The behavior of L2PF is independent on whether the STP in the provider SVLAN is enabled or not.

Global configuration

By default, the device enables Layer 2 protocol forwarding (L2PF). Customer-BPDU packets on all CVLANs are forwarded to provider network ports when they arrive at customer-edge ports.

Since L2PF is globally enabled by default, all CVLANs have L2PF enabled by default. No CLI configuration is needed.

To globally disable Layer 2 protocol forwarding, enter the following command:

device(config)# no l2protocol-forwarding stp

To disable L2PF on a particular CVLAN, enter commands such as the following:

```
device(config)#esi acme encapsulation cvlan
device(config-abc)#vlan 10
device(config-abc-vlan-10)# no l2protocol-forward stp
```

Syntax: [no] l2protocol-forwarding stp

Use the **no** parameter to disable Layer 2 protocol forwarding.

IEEE 802.1ah Provider Backbone Bridging (PBB) Networks for the CES 2000 Series and the CER 2000 Series devices

Overview

The IEEE 802.1ah Provider Backbone Bridges (PBB) standard was developed to address the limitations of Provider Bridges (PB) and to add additional capabilities sought by Service Providers. When compared to a PB network, a PBB network deployment offers simplified operations, lower capital expenditures, and overall better scalability in terms of the number of supported customers. PBB also provides advantages when used in conjunction with VPLS, since PBB reduces the overall MAC address learning requirements. This section provides an overview of PBB, describes its advantages over PB, examines common PBB deployment scenarios, and examines its many benefits when deployed in combination with a core MPLS network supporting VPLS.

Provider Backbone Bridges

The Provider Backbone Bridges (PBB) standard, (IEEE 802.1ah), was developed to address the limitations of the Provider Bridges (PB) standard, (IEEE 802.1ad), and to add additional capabilities sought by Service Providers.

PB allows Service Providers to use a V-LAN identifier (VID) space separate from the customer VID (C-VID) space. PB adds a Service Provider VLAN Tag (S-TAG) containing a Service Provider VID (S-VID) to Ethernet frames (Figure 26). Because PB stacks a second VLAN tag to Ethernet frames, it is also known as "Q-in-Q," as a reference to the standard that originally defined VLAN tags, in other words, IEEE 802.1Q, which is known as defining "Q" frames.

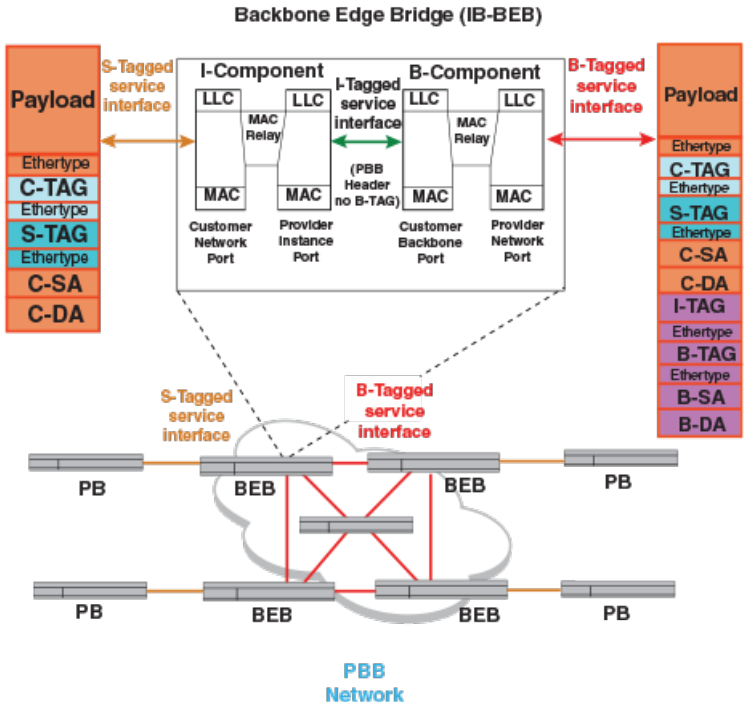
The S-VID field of the S-TAG is 12 bits long, which is the same length of a C-VID field of a customer VLAN Tag (C-TAG). Even though 12 bits can address up to 4096 distinct values, two values have special meaning and are reserved. Therefore, the Service Provider is limited to at most 4090 distinct S-VID values to identify service instances, that is, services or customers in a PB network. Another drawback is that PB frames are addressed by customer Media Access Control (MAC) addresses. This means that core Ethernet switches in a PB network have to learn all the source MAC addresses of all the customer frames traversing the core of the PB network. Thus, the size of the MAC address tables of core PB switches ultimately limits the number of customers that can be supported by a PB network.

To address the above described PB shortcomings, PBB adds a hierarchy view to Ethernet by encapsulating PB frames with a PBB header (which becomes the equivalent of a "Service Provider MAC header") containing a Backbone Destination MAC Address (B-DA), Backbone Source MAC Address (B-SA), and two new tags (Figure 26), which are described later in this document. What makes the B-DA and B-SA "backbone" addresses is the fact that these are MAC addresses of Service Provider's PBB edge switches. An edge PBB switch encapsulates an ingress PB frame with a PBB header containing the destination MAC address of an appropriate egress edge PBB switch. The egress edge PBB switch removes the PBB header and forwards the frame to an attached PB network. Because PBB adds a PBB header containing new destination and source MAC addresses, it is also known as "MAC-in-MAC."

By adding the PBB header, PBB isolates the Service Provider and customer address spaces. This means that Ethernet switches in the core of the Service Provider network will no longer learn customer MAC addresses or use customer MAC addresses to forward customer frames to their destinations. This improves the scaling of the Service Provider network in terms of the number of supported customers, since the number of supported customers is no longer directly tied to the size of the MAC address tables of the core Ethernet switches. In addition, the Service Provider network is now protected from customer network failures, since frame forwarding is now based on its own PBB header. Moreover, customers benefit from added security, since the customer's MAC addresses are no longer learned or used for frame forwarding decisions in the core of the Service Provider network.

As additional benefits to the Service Provider, PBB has the potential to simplify operations, e.g., by separating the customer and Service Provider addressing spaces, and to lower capital expenditures by reducing the cost of Ethernet switches used in the core of the network, since memory and processing power requirements are reduced by limiting MAC address learning to backbone MAC addresses.

FIGURE 25 Customer, PB, and PBB frame formats

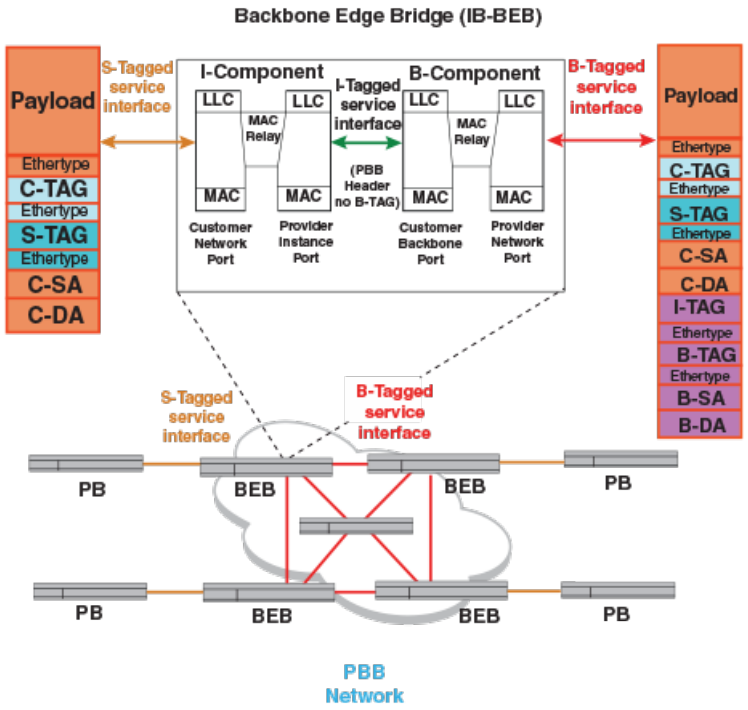


The new Backbone Service Instance Tag (I-TAG) contains a Backbone Service Instance Identifier (I-SID), which is 24 bits long. The I-SID field allows a Service Provider to identify up to 2 to the power of 24, that is, over 16 million service instances. In other words, over 16 million services or customers can be uniquely identified using the I-SID field. Therefore, PBB's I-TAG allows for highly scalable services by eliminating the 4090 service instances limitation of PB.

The semantics and the structure of the Backbone VLAN Tag (B-TAG) are identical to that of the PB S-TAG. The B-TAG was designed this way so that core PBB switches do not need to be aware of PBB. In fact, standard PB switches can be used in the core of a PBB network. Only the switches at the edge of the Service Provider PBB network need to be aware PBB.

A PBB network uses two types of bridges (Figure 27): Backbone Edge Bridges (BEB) and Backbone Core Bridges (BCB). As explained above, the functionality required from a BCB is the same as a standard IEEE 802.1ad PB bridge. A BEB is used at the boundary of a PBB network to add and remove the PBB header.

FIGURE 26 Backbone Edge Bridge operation



As defined in IEEE 802.1ah, a BEB has two main components: an I-Component and a B-Component. From left to right in Figure 27, the I-Component maps S-VIDs to I-SIDs and adds a PBB header without a B-TAG, while the B-Component maps I-SIDs to B-VIDs and adds a B-TAG. These actions are reverted in the opposite direction.

As shown in Figure 27, a BEB containing an I-Component and a B-Component is called an IB-BEB. The B-Component of an IB-BEB forwards frames towards the core of a PBB network based on backbone MAC addresses (that is, it learns backbone MAC addresses), while the I-Component forwards frames towards the PB network based on customer MAC addresses (that is, it learns customer MAC addresses). The terms I-BEB and B-BEB refer to optional BEBs that support a single component type, that is, either I-Component or B-Component, respectively. I-BEBs and B-BEBs expose an I-Tagged service interface, which carries frames with a PBB header, but without a B-TAG.

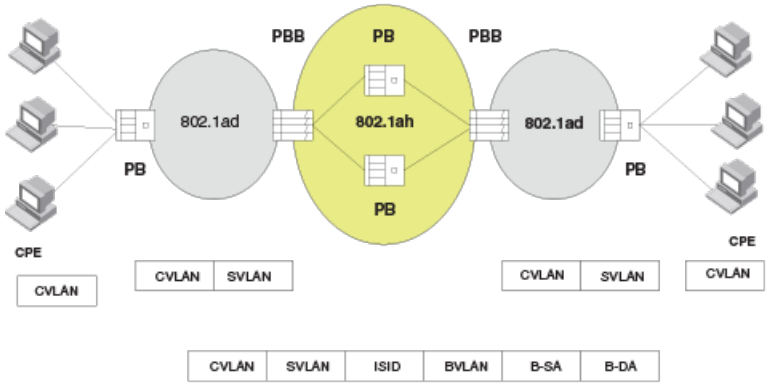
As with PB, PBB networks use a Spanning Tree Protocol (STP), e.g., RSTP or MSTP, to dynamically determine the active topology of the PBB network and MAC address learning to dynamically build a forwarding database. Since an IB-BEB forwards frames to PB and PBB networks, it has to learn customer and backbone MAC addresses. However, since an IB-BEB is at the edge of the Service Provider network, it only learns customer MAC addresses of the local traffic.

NOTE
When using ESI VLANs in the configuration, always configure the protocol in default VLAN.

IEEE 802.1ah Provider Backbone Bridging (PBB)

This section provides details on IEEE 802.1ah Provider Backbone Bridging (PBB), with a description and a configuration example of an integrated PB and PBB network.

FIGURE 27 Integrated IEEE 802.1ad and IEEE 802.1ah network architecture



The Provider Backbone Bridge (PBB) protocol typically resides at the core of a carrier network, interconnecting PB networks. Inside of a PBB network a carrier usually deploys PB systems for layer 2 interconnectivity. The PBB protocol is designed to insulate carrier infrastructure from having to learn customer MAC addresses and provide a separation of service and networking components of an Ethernet service provided to the customers:

- It provides IEEE 802.1ah encapsulation to add a backbone MAC header on the incoming PB packet (which has customer DA or SA) so core carrier switches don't need to learn customer MAC addresses.
- It supports creation of EVC (Ethernet Virtual Circuits) with a concept of an Ethernet Service Instance (ESI), which is enabled by use of a globally unique 24-bit I-component Service Identifier (ISID).
- PB packets are encapsulated with ISID, BVLAN, B-SA, and B-DA as shown in Figure 28.
- Layer 2 switches in the IEEE 802.1ah-encapsulated network are normal PB systems, which process BVLAN header as if it was an SVLAN-encapsulated packet.
- **Tag-types** : BVLAN and SVLAN tag-types are same (default 0x88a8), so the PB systems inside a PBB network can process BVLAN packets as if they were SVLAN encapsulated.
- **ISID operation** : A 24-bit ISID supports the provider bridging operation framework where an EVC setup gives the closed environment for customer packets on the two ends to be limited to the EVC. The EVC is identified by the ISID value.
- Customer MAC address learning is limited to the ISID domains that are part of an EVC. A PBB device doesn't need to learn customer MAC addresses that are not part of this EVC. This provides scalability, as devices are not required to store every customer MAC address that passes through the provider backbone network.
- **BVLAN** : BVLAN provides the network connectivity among PBB nodes. Notice that while in a PB network the SVLAN provides both the networking operation and service interconnection, in a PBB network the service provisioning (ISID) is totally independent on networking framework (BVLAN).

IEEE 802.1ah configuration options

Two types of architectures are possible for providing connectivity to PBB networks:

- Ingress ports receive SVLAN-encapsulated packets. In this case, the ingress port already contains SVLAN mapping and the PBB system provides additional encapsulation for the PBB header.
- Ingress ports directly connect to customer devices and receive CVLAN-mapped traffic. The system then performs SVLAN mapping internally and then performs IEEE 802.1ah encapsulation before sending packets out to PBB network. This model is not supported in CES 2000 Series and CER 2000 Series.

Tag type configuration

Tag Type values for different encapsulations are defined globally for all types of VLANs.

At most, two different external tag-types can be defined for CES 2000 Series. The restriction of two tag types doesn't include an additional tag-type that can be defined for ISID. ISID encapsulation is performed internally and doesn't have a port association.

IEEE-defined tag-type values are as follows:

- CVLAN: 8100
- SVLAN & BVLAN: 88a8 (both PBB and PB bridges have identical outer tag-type so core PB bridges process BVLAN tags and interoperate with PBB bridges)
- ISID: 88e7

When configuring tag types, all tag types must be entered in **a single command line** , as shown below.

```
device(config)# tag-type cvlan tag1 svlan tag2 bvlan tag2
```

Displaying tag types

To display the tag types, enter the following command.

```
device(config)# show tag-type
Encap      Current VLAN Tags             Default VLAN Tags
-----      -----------------             -----------------
cvlan                 8100                       8100
svlan                 9100                       88A8
isid                  86B5                       88E7
bvlan                 9100                       88A8
```

NOTE

On PB networks, SVLAN and BVLAN tag types must be same.

Port configuration for IEEE 802.1ah and IEEE802.1ad at each interface

Table 26 displays the Port configuration for IEEE 802.1ah and IEEE 802.1ad.

TABLE 26 Port configuration for IEEE 802.1ah and IEEE 802.1ad

Port type	Description	Characteristics
PB_CE	Customer Edge Port (IEEE 802.1ad). This is the default port type.	• CVLAN tag.
PB_PN	Provider Network Port (IEEE 802.1ad)	• SVLAN tag • IEEE 802.1ad frame at this interface.
PBB_BE	Backbone-Edge (IEEE 802.1ah)	• SVLAN tag • IEEE 802.1ad frame • This is same encapsulation type as PN, but the provider side of the port (into the carrier network) is an IEEE 802.1ah frame. • A BE port connects to a PB network.
PBB-BN	Backbone-Network (IEEE 802.1ah)	• BVLAN tag

IEEE 802.1ah Provider Backbone Bridging (PBB)network configuration example

The CES 2000 Series and CER 2000 Series sample configuration for PBB functionality is shown in Figure 29. The CES 2000 Series and CER 2000 Series take in SVLAN inputs, map internally to an ISID, and then bind to a BVLAN to provide PBB functionality.

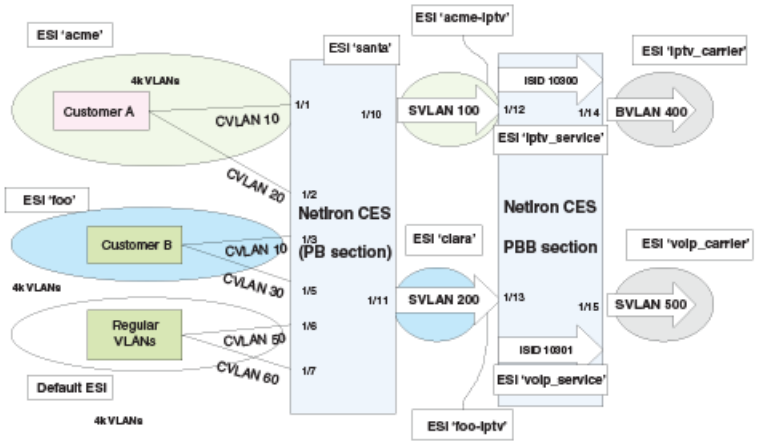
In Figure 29, PB output with SVLAN encapsulation using ESI 'santa' is used to provide input to Ethernet port 1/12 which is configured as a backbone-edge port.

A new ESI 'acme-iptv' is created for the incoming SVLAN (at VLAN ID = 100). This is assigned to a BVLAN (VLAN ID = 400) under ESI 'iptv_carrier' by first mapping it to ISID 10300 under ESI 'iptv_service'.

In this example, PB output with SVLAN encapsulation using ESI 'santa' is used to provide input to Ethernet port 1/12 which is configured as a backbone-edge port.

A new ESI 'acme-iptv' is created for the incoming SVLAN (at VLAN ID = 100). This is assigned to a BVLAN (VLAN ID = 400) under ESI 'iptv_carrier' by first mapping it to ISID 10300 under ESI 'iptv_service'.

FIGURE 28 Provider Backbone Bridging (PBB) functionality



IEEE 802.1ah configurations

The PBB protocol performs IEEE 802.1ah encapsulation on packets that arrive from PB network and are SVLAN-tagged so core carrier switches don't need to learn customer MAC addresses.

Two types of architectures are possible for providing connectivity to PBB networks.

Sequential steps for a IEEE 802.1ah configuration CLI example are shown below.

Define interface types

First step in setting up a PB network configuration is to set interface type properly. Interface type has to match the encapsulation type of the VLAN expected on the interface.

By default, interfaces are of type 'customer-edge' so there is no need to define an interface type for CVLAN ESIs.

1. Set 1/12 and 1/13 to 'backbone-edge' (SVLAN).

```
device(config)# interface ethernet 1/12
device(config-if-e10000-1/10)# port-type backbone-edge
device(config-if-e10000-1/10)# exit
```

2. Configure port 1/14 to be of 'backbone-network' type.

```
device(config)# interface ethernet 1/14
device(config-if-e10000-1/10)# port-type backbone-network
device(config-if-e10000-1/10)# exit
```

Create an SVLAN ESI on IEEE 802.1ad side (PBB ingress)

1. Create an ESI on 801.1ad side. Acme-iptv" is name of the provider IEEE 802.1ad service.

```
device(config)# esi acme-iptv encapsulation svlan
```

2. Define SVLAN 100 as one of the parameters for ESI Acme-iptv.

```
device(config-esi-acme-iptv)# vlan 100
```

3. Associate physical ports to SVLAN 100.

```
device(config-esi-acme-iptv-vlan-100)# tagged ethernet 1/12
device(config-esi-acme-iptv-vlan-100)# exit
```

Create PBB ESI for ISID (PBB ingress - BEB function)

1. Create an ESI on PBB for ISID. Configure "iptv-carrier" as the name of the carrier service.

```
device(config)# esi iptv-service encapsulation isid
```

2. Define any special parameters for this ESI.

```
device(config-esi-iptv-service)# isid 10300
device(config-esi-iptv-service-isid-10300)# exit
```

Create PBB ESI on the carrier side (BVLAN)

1. Create an ESI on PBB. Configure "iptv-carrier" as the name of the carrier service providing IEEE 802.1ah.

```
device(config)# esi iptv-carrier encapsulation bvlan
```

2. Define any special parameters for this ESI.

```
device(config-esi-iptv-carrier)# vlan 400
```

3. Port configuration.

```
device(config-esi-iptv-carrier-vlan-400)# tagged ethernet 1/14
```

Bind ISID to BVLAN

Specify that ISID ESI "iptv-service" is a client of BVLAN ESI "iptv-carrier" and binding is done in two steps. The first command below binds SVLAN ESI 'santa' to ISID 'iptv-service' then puts the ISID inside BVLAN. The second command binds SVLAN ESI 'clara' to ISID 'iptv-service' then puts the ISID inside BVLAN.

1. Bind SVLAN ESI 'santa' to ISID 'iptv-service' then puts the ISID inside BVLAN.

```
device(config-esi-iptv-service)# esi-client acme-iptv isid iptv-service
```

2. Bind SVLAN ESI 'clara' to ISID 'iptv-service' then puts the ISID inside BVLAN.

```
device(config-esi-iptv-service)# esi-client clara isid voip-service
```

ESI configuration display after mappings

To display the ESI configurations, enter the **show esi** command.

```
device(config)#show esi
ESI Name      Encap    Number of  Provider   Provider   Provider   Client
                       Members    ESI        Encap      VLAN       ESIs
--------      -----    ---------  --------   -----      --------   ------
acme          cvlan    2          santa        svlan      100      0
foo           cvlan    2          clara        svlan      200      0
santa         svlan    1          iptv-service isid       10300    1
clara         svlan    1          iptv-service isid       10300    1
iptv-service  isid     1          iptv-carrier bvlan      400      2
iptv-carrier  bvlan    1          None         None       None     1
```

Syntax: show esi

```
device(config-esi-foo)#
ESI: acme Encapsulation: cvlan
PORT-VLAN 42, Name [None], Priority Level0
L2 protocols   : NONE
ESI: acme Encapsulation: cvlan
PORT-VLAN 43, Name [None], Priority Level0
L2 protocols   : NONE
ESI: acme Encapsulation: cvlan
PORT-VLAN 10, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/3
ESI: foo Encapsulation: cvlan
PORT-VLAN 30, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/5
ESI: foo Encapsulation: cvlanv
PORT-VLAN 100, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/10
ESI: santa Encapsulation: svlan
PORT-VLAN 200, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/11
```

Integrated IEEE 802.1ad and IEEE 802.1ah

The Provider Backbone Bridge (PBB) protocol usually resides at the core of the network, interconnecting Provider Bridging networks. Integrated IEEE 802.1ad and IEEE 802.1ah provides the following:

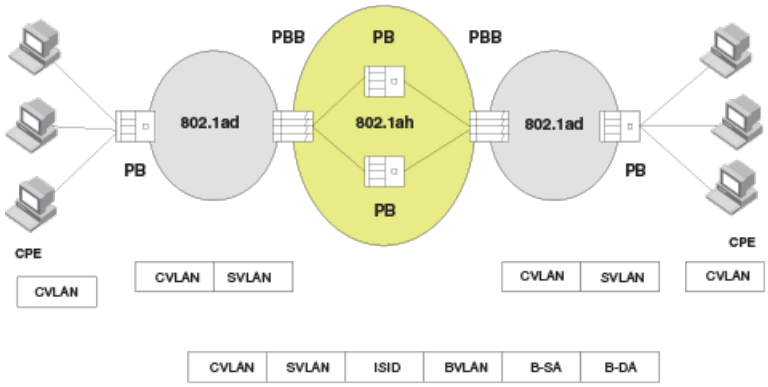
- Provides IEEE 802.1ah encapsulation to add a backbone MAC header on the incoming Provider Bridging packet so the carrier switches will not need to learn the customer MAC address.

- Supports the creation of Ethernet Virtual Circuits (EVC) with a concept of an Ethernet Service Instance (ESI), which is enabled by use of a globally unique 24-bit 1-component Service Identifier (ISID).
- Layer 2 switches in the IEEE 802.1ah-encapsulated network are normal Provider Bridging systems, which process BVLAN header as if it was an SVLAN-encapsulated packet.
- **Tag-types** - BVLAN and SVLAN tag-types are the same (default 0x88a8), so the Provider Bridging system inside the PBB network can process BVLAN packets as they were SVLAN encapsulated.
- ISID Operation:
  - 24-bit ISID supports the provider bridging operation framework where an Ethernet Virtual Circuits (EVC) setup gives the closed environment for customer packets on the two ends to be limited to the EVC.
  - Customer MAC address learning is limited to the ISID domains that are part of an EVC. A PBB device does not need to learn customer MAC addresses that are not part of this EVC. This provides scalability, as devices are not required to store every customer MAC address that passes through the provider backbone network.
- **BVLAN:** BVLAN provides the network connectivity among PBB nodes.

  NOTE
  While in a Provider Bridging network the SVLAN provides both the networking operation and service interconnection, in a PBB network the service provisioning (ISID) is totally independent on networking framework (BVLAN).

Figure 30 provides an example of a Provider Backbone Bridged (PBB) network interconnecting two Provider Bridged (PB) networks.

FIGURE 29 Provider Backbone Bridged network interconnecting two Provider Bridged networks

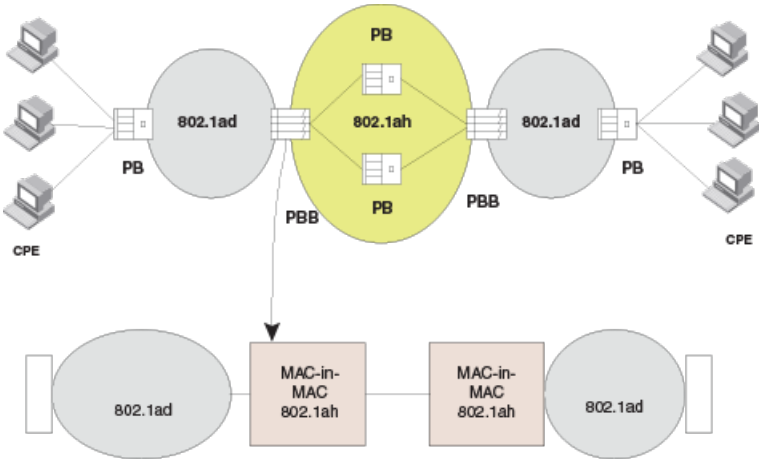


IEEE 802.1ah (PBB) configurations

Ingress ports receive SVLAN-encapsulated packets. In this case, the ingress port already contains SVLAN tagged frames and the PBB switch provides additional encapsulation by adding the PBB header. An SVLAN can be mapped to one unique ISID or alternatively multiple SVLANs can be mapped to the same ISID.

In Figure 31, the PBB device expects tagged Ethernet packets coming in with SVLAN encapsulation. This is the most common configuration for PBB as a provider of carriers' backbone infrastructure.

FIGURE 30 IEEE 802.1ah PBB configuration



Interface configuration for Provider Bridge and Provider Backbone Bridge (PBB) networks

When configuring the Extreme device, a port is configured to be one of the one of the interface types:

- – Customer-edge (CE)
  – Provider-network (PN)
  – Backbone-edge (BE)
  – Backbone-network (BN)

Configuring the port-type for an interface

Before a VLAN can be provisioned for an interface, the port-type for the interface must be defined. This command defines a port type for an Ethernet interface. The port-types specify both sides of IEEE 802.1ad and IEEE 802.1ah networks. To define the port-type of the interface, enter commands such as the following.

```
device(config)# interface ethernet 5/1
device(config-if-e10000-5/1)# port-type provider-network
device(config-if-e10000-5/1)#
```

Syntax: port-type [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the Backbone Edge Port for IEEE 802.1ah PBB

Use the **backbone-network** parameter to specify the Backbone Network Port for IEEE 802.1ah PBB

Use the **customer-edge** parameter to specify the Customer Edge Port for IEEE 802.1ad PB

Use the **provider-network** parameter to specify the Provider Network Port IEEE 802.1ad PB

Displaying port- types

The **show interfaces** command displays port-type for an interface, as shown below.

```
device(config-if-e10000-5/1)# show interfaces
10GigabitEthernet5/1 is empty, line protocol is down
  Hardware is 10GigabitEthernet, address is 0e04.80de.ada0 (bia 0e04.80de.ada0)
  Configured speed 10Gbit, actual unknown, configured duplex fdx, actual unknown
```

```
Member of VLAN 1 (untagged), port is in untagged mode, port state is Disabled
STP configured to ON, Priority is level0, flow control enabled
arp-inspection-trust configured to OFF
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
Port-type (802.1ad/802.1ah): provider-network
300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants
NP received 0 packets, Sent to TM 0 packets
NP Ingress dropped 0 packets
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions
NP transmitted 0 packets, Received from TM 0 packets
```

Syntax: show interfaces

TABLE 27 Display of show interfaces command

This field...	Displays...
Module type Port# is *State*	The *module type* variable specifies a type of interface module, such as 10GigabitEthernet. The *port#* variable specifies the port number for the interface module. The *state* variable if the interface module is up or down.
Line protocol is *status*	The *status* variable specifies if the line protocol is up or down. If the interface is down due to Link Fault Signaling - Remote Fault Notification (LFS or RFN) the reason is indicated as: "(remote fault)".
STP Root Guard is *status*	The *status* variable specifies if the STP Root Guard is enabled or disabled.
STP BPDU Guard is *status*	The *status* variable specifies if the STP BPDU Guard is enabled or disabled.
Hardware is *module type*	The *module type* variable specifies a type of interface module, such as # Gigabit Ethernet.
Address is *MAC- address*	The *MAC- address* variable specifies the MAC address of the port.
Configured speed and actual speed	The speed that the module has been configured to operate at, and the actual speed at which it is operating.
Configured port speed and actual duplex	The port capacity that the module has been configured to operate at, and the actual speed at which it is operating.
Member of *VLAN #* (untagged) *port#* L2 VLANS (tagged) Port is in dual mode/untagged/tagged mode Port state is *status*	The *VLAN#* (untagged) variable specifies a port that is a member of only 1 VLAN. The *port#* Layer 2 VLANS (tagged) variable specifies a port that is a member of multiple ports and untagged. A port is in *dual- mode* specifies member VLAN ports as untagged and tagged. The default mode is dual-mode. The *status* variable identifies the flow of traffic as forwarding or disabled.
STP configured to *status* Priority level Flow control *status*	The *status* variable specifies if the STP is ON or OFF. The priority level assigned to the port-based VLAN. The priority level is on scale from 0-7. The default is 0. The *status* variable is enabled or disabled.
Priority force *status*	The *status* variable specifies if the priority force on a port is disabled on enabled.

TABLE 27 Display of show interfaces command (continued)

This field...	Displays...
Drop precedence level *value*	Identifies the TOS or DSCP value in the IPv4 or IPv6 packet header. The *value* variable specifies the drop precedence on a scale from 0-3. Packets that contain a DSCP value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped. The default value is 0.
Drop precedence force *status*	The *status* variable specifies the drop precedence force as enabled or disabled. Identifies the drop precedence if the force command is configured for a specific ingress port.
arp-inspection-trust configured to *status*	The *status* variable specifies if arp-inspection-trust feature is configured ON or OFF. The default trust setting for a port is untrusted.
Mirror *status*	The *status* variable specifies if the port mirror command is configured as enabled or disabled.
Monitor *status*	The *status* variable specifies if the port monitor command is configured as enabled or disabled.
Trunk membership	The *Trunk membership* variable identifies the interface module as a member of a primary or secondary port. This specifies members of an active port or not a member of an active port.
Configured trunk membership	The *Configured trunk membership* variable identifies the interface module as a member of any configured trunk or not a member of a configured trunk.
Port name	The *port name* variable identifies the name assigned to the port.
MTU *# bytes* , encapsulation ethernet	Maximum Transmission Unit (MTU) refers to the size of the largest packet or frame that a known layer can pass forward. The *# bytes* variable refers to size of the packet or frame.
#seconds input rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* input rate refers to: • The *value* of bits received per second. • The *value* of packets received per second. • The *%* utilization specifies the port's bandwidth used by received traffic.
# seconds output rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* output rate refers to: • The *value* of bits transmitted per second. • The *value* of packets transmitted per second. • The *%* utilization specifies the port's bandwidth used by transmitted traffic.
value packets input, *value* bytes	• The *value* variable specifies the number of packets received. • The *value* variable specifies the number of bytes received.
Received *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module is receiving on broadcasts, multicasts, and unicast traffic.
value input errors, *value* CRC, *value* frame, *value* ignored	• The *value* variable specifies the number of received packets with errors. • The *value* variable specifies the number of received packets with CRC errors. • The *value* variable specifies the number of received packets with alignment errors. • The *value* variable specifies the number of received packets that are discarded.
value runts, *value* giants	The *value* runts variable specifies the number of small packets that are less than 64 bytes.

TABLE 27 Display of show interfaces command (continued)

This field...	Displays...
	The *value* giants variable specifies the number of large packets greater than 1518 bytes.
NP received	The number of packets received on the Network Processor (NP).
NP transmitted	The number of packets sent from the Network Processor to the Traffic Manager (TM).
NP Ingress dropped	The number of ingress packets dropped on the Network Processor.
value packets output *value* bytes	• The *value* variable specifies the number of transmitted packets. • The *value* variable specifies the number of transmitted bytes.
Transmitted *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module transmitted on broadcasts, multicasts, and unicast traffic.
value output errors, *value* collisions	• The *value* variable specifies the number of transmitted packets with errors. • The *value* variable specifies the number of transmitted packets with collision errors.
Network Processor transmitted *value* packets Received from Traffic Manager *value* packets	The *value* variable specifies the number of packets transmitted from the Network Processor. The *value* variable specifies the number of packets received by the Network Processor from the Traffic Manager.

Point to Point PBB

Point to Point PBB (P2P PBB) provides the ability to turn off MAC address learning on a per service instance basis (ISID). This provides support for point-to-point services, such as EVPL, which do not require MAC address learning. Point to Point PBB is designed to flood the traffic to a specific remote BEB in the core PBB network, instead of flooding across all the BEBs.

Limitations

When P2P PBB is enabled, the MACs learned on S-VLANs are duplicated in the new flood domain. This will reduce the maximum number of supported MACs from 131072 VPLS MACs to 65536 VPLS MACs.

Configuring Point to Point PBB

Use the **p2p-mac** command to enable this feature. Since the P2P PBB feature is specific to a service instance it has to be executed for each ISID.

```
device(config)#esi A-isid encapsulation isid
device(config-esi-A-isid)#isid 18001
device(config-esi-A-isid-isid-18001)#p2p-mac 001b.edb4.5ac1
```

Syntax: [no] p2p-mac *mac-addr*

The *mac-addr* parameter requires the MAC address of the remote BEB.

Use the **no** command to disable this feature.

Show commands

Use the **show flood-domain** command to display the extra flood-domain (shadow flood domain). This extra flood domain information is only shown for P2P PBB instances.

```
device# show flood-domain
FDID                Type              NumMem             VLAN Owner      VLAN Owner ESI
----                ----              ---------      ----------      ---------
4357                PBB                2                    1234               i-isid
4358                B_VLAN         1                        500               b-vlan
4359                S_LOOP         1                        100               s-vlan
4360                S_FWD          2                       100               s-vlan
4369, 4370      PBB                          1                2345               A-isid
```

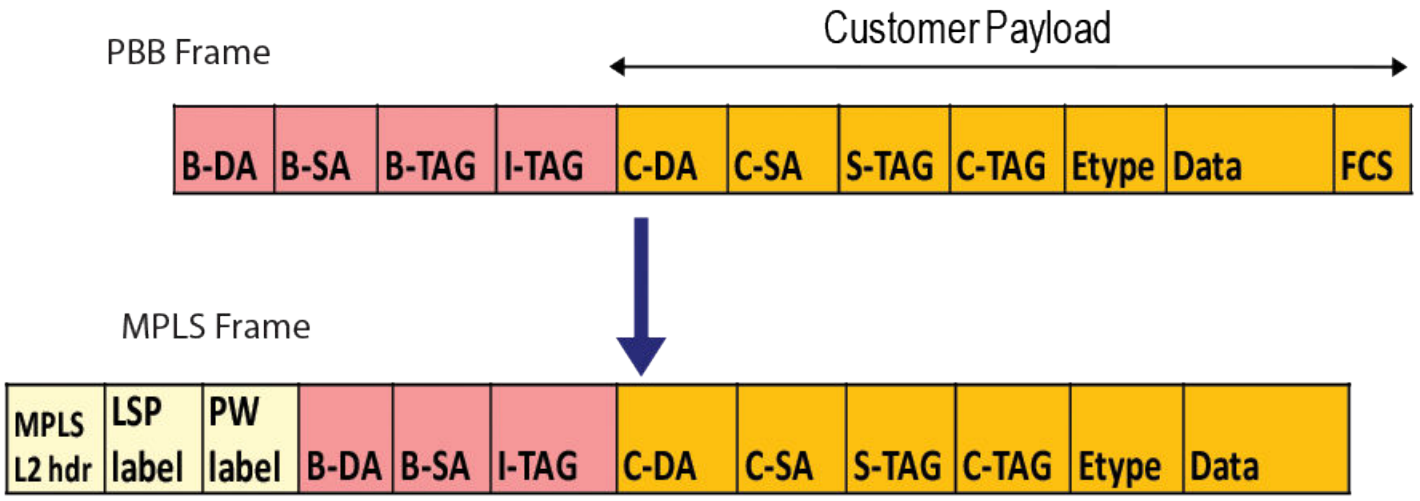
You can use the **show flood-domain** command with the *fd_id* variable to filter the flood domain. The **show flood-domain** command with the *shadow_fd_id* variable can also be used to show the identical information.

```
device#show flood-domain 4369
device#show flood-domain 4370
FDID                Type           NumMem              VLAN Owner      VLAN Owner ESI
----                ----          ----------      ----------      ---------
4369, 4370      PBB                    1               2345               A-isid
```

ISID mapping to VPLS

The ISID mapping to VPLS feature allows the service instance to be identified end-to-end across the Ethernet and VPLS networks using the same value without modifying how the MPLS network operates. When the PBB packet is sent out on the MPLS cloud, the ISID is always preserved in the packet as the payload tag. A VPLS instance can recognize PBB packets only when it is configured in tagged-mode. Figure 32 illustrates this feature.

FIGURE 31 ISID mapping

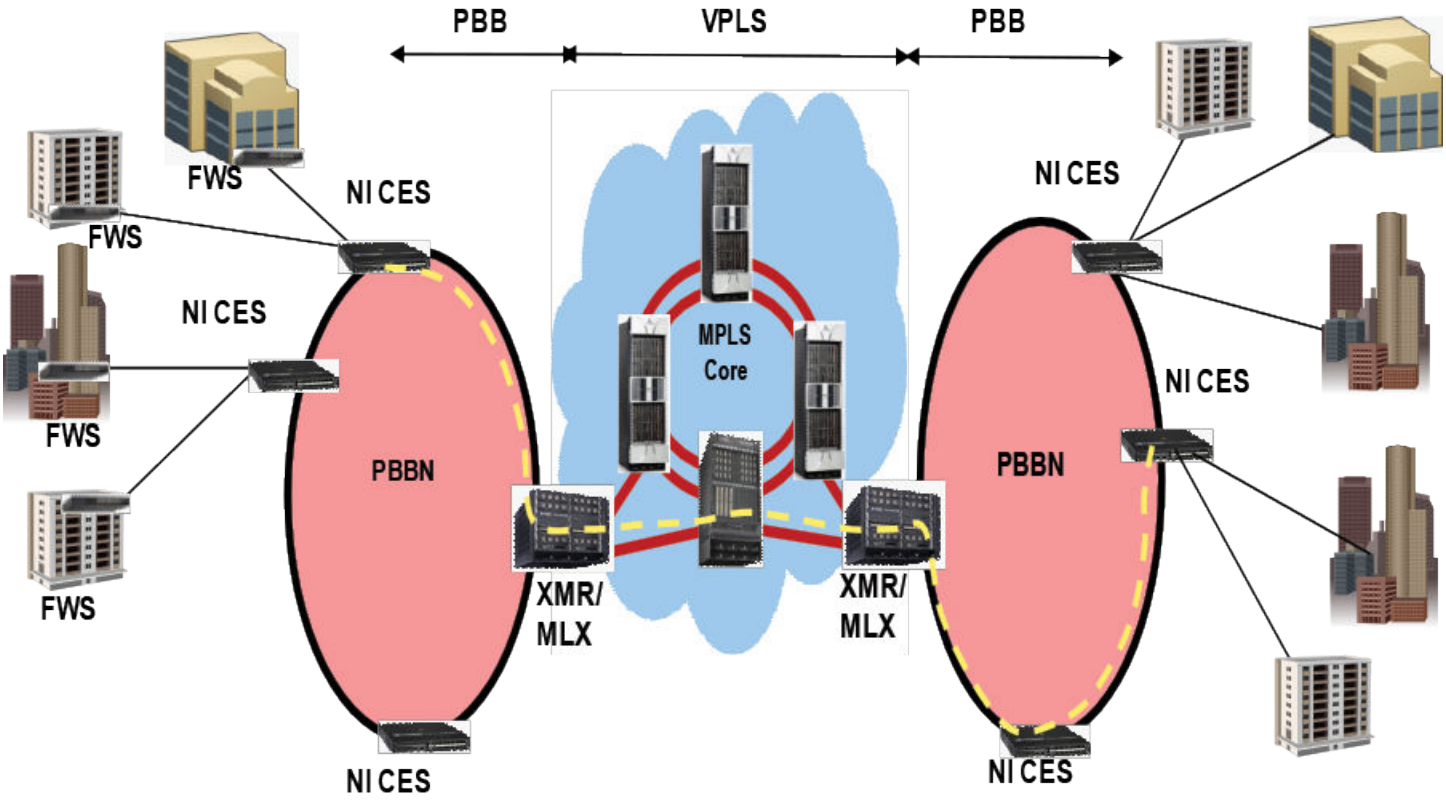


ISID endpoint configuration considerations

- ISID range is between 256 (0x100) and 16777214 (0xFFFFFE).
- ISID endpoints can be configured only if the VPLS instance is using the tagged mode.

- You cannot mix ISID with non-ISID endpoints in the same VPLS instance.
- All endpoints within a VPLS instance should have the same ISID.
- For remote switching, you must configure the same ISID on both the ingress and egress routers.
- You cannot configure more than one ISID endpoint on the same port in the same VPLS instance.
- Multicast snooping will not be allowed on a VPLS instance which contains ISID endpoints.
- The **default-max-frame-size** configured must be sufficient to carry the entire packet across the MPLS cloud.

FIGURE 32 ISID endpoint configuration example



Configuring the ISID endpoints

The existing VLAN CLI under the VPLS configuration mode has been enhanced to configure ISID endpoints.

To configure ISID endpoints, enter commands such as the following:

```
device(config-mpls)# vpls test 100
device(config-mpls-vpls-test)# vc-mode tagged
device(config-mpls-vpls-test)# vlan 100 isid 450
```

Syntax: [no] vlan 1-4094 [**inner-vlan 1-4095** | **isid 256 - 16777214**]

The outer vlan can be from 1 to 4094, but it excludes the default VLAN ID.

The **inner-vlan** can be in the range from 1 to 4095.

The **ISID** can be in the range from 256 to 16777214. The ISIDs from 0 to 255 and 16777215 are reserve.

Tag type and ether type

For a packet to be classified as a Dual Tag packet, the Ether type of the inner-vlan tag in the packet should always be 0x8100. The expected Ether type for the outer vlan (B-TAG/S-TAG) can be changed using the global **tag-type** command. The default is 0x8100. Different Ether types can be configured per-port.

The existing tag-type CLI has been enhanced to specify the expected Etype for I-TAG. The default is 0x88e7 (this Etype configuration is global and cannot be specified per-port). The configuration will take effect immediately, and you do not need to reload the box for the new Ether type to take effect.

```
device(config)# tag-type)
device(config)# tag-type isid 88e8
```

Syntax: [no] tag-type *HEX-VALUE* [*port-type slot/port* [**to** *slot/port*]

Use the *HEX-VALUE* parameter to specify the etype in hex (Default: 0x8100).

Syntax: [no] tag-type isid *HEX-VALUE*

Use the **isid** parameter to specify the etype for I-Tagged packets.

Use the *HEX-VALUE* parameter to specify the etype in hex (Default: 0x88e7).

Topology Groups

For topology groups, the member VPLS VLAN commands have been enhanced to specify the ISID level. To configure member VPLS VLAN at the ISID level, enter commands such as the following.

```
device(config)# topo 1
device(config-topo-group-1)# master-vlan 100
device(config-topo-group-1)# member-vlan vpls id 100 vlan 100 isid 450
```

Syntax: [no] member-vlan vpls id *vpls-id* | **name** *name* **vlan** *vlan-id* [**to** *vlan-id*] **isid** *isid*

OR

Syntax: [no] member-vlan vpls id *vpls-id* | **name** *name* **vlan** *vlan-id* [**inner-vlan** *inner-vlan-id* [**to** *inner-vlan-id*] **...** | **isid** *isid*]

Show commands

Use the **show mpls vplsb id** command to display ISID information.

```
device# show mpls vpls id 3
VPLS name_raw, Id 3, Max macentries: 8192
Total vlans: 1, Tagged ports: 3 (3 Up), Untagged ports 0 (0 Up)
IFL-ID: 4097
Vlan300 inner-vlan500
Tagged: ethe3/1 ethe3/11 ethe3/13
Total VC labels allocated: 16 (983072-983087)
VC-Mode: Raw
Total VPLS peers: 1 (1 Operational)
Peer address: 200.200.200.200, State: Operational, Uptime: 1 hr 10 min
Tnnlin use: tnl1(4)
LDP session: Up, Local VC lbl: 983072, Remote VC lbl: 983072
Local VC MTU: 1500, Remote VC MTU: 1500
LOCAL VC-Type: Ethernet (0x05), Remote VC-Type: Ethernet (0x05)
CPU-Protection: OFF
Local Switching: Enabled
```

To display the ISID endpoints in a topology group, use the following command.

```
device# show topology
Topology Group 1
==================
Topo HW Index   : 65535
Master VLAN     : 100
VPLS VLAN exist : TRUE
Member VLAN     : None
Member Group  : None
M
ember VPLSs  : vpls id 100 vlan 100 isid 450

Control Ports : None
Free Ports :
VPLS ID 100 VLAN 100 ISID 450 - ethe 2/1
```

Syntax: show topology

Load balancing traffic

You can enable the device to mask out different PBB header fields. When a PBB packet is received on the device, by default the network processor checks the packet (including the Bvlan SA/DA) and includes the PBB header fields in hash calculations even when an ISID endpoint is not configured on the device.

To mask out the PBB header fields during hash calculations, use the **load-balance** command.

```
device(config)# load-balance mask pbb cust-l2-hdr all
device(config)# load-balance mask pbb cust-ip4-ip6-hdr 1
device(config)# load-balance mask ethernet isid 2 1
```

Syntax: [no] load-balance mask ethernet *sa-mac* | *da-mac* | *vlan* | *etype* | *inner-vlan* | *isid* **all** | *slot* [*np-id*]

Syntax: [no] load-balance mask pbb *cust-l2-hdr* | *cust-ip4-ip6-hdr* **all** | *slot* [*np-id*]

By default all options are off.

When using the **isid** option for the **load-balance mask ethernet** command, the ISID field in the PBB packet on the endpoint will be masked out.

When using the **cust-l2-hdr** option for the **load-balance mask pbb** command, the destination and source MAC addresses in the customer Layer 2 header will be masked out during hash-calculations. This applies only to the ingress device, for packets received on the endpoint.

When using the **cust-ip4-ip6-hdr** option for the **load-balance mask pbb** command, the IPv4 and IPv6 protocol field and destination and source addresses in the customer Layer 3 header will be masked out. This applies only to the ingress device, for packets received on the endpoint.

> NOTE
> During the hash calculations, cust-l2-hdr and cust-ip4-ip6-hdr will not be considered.

When using the **all** option, it will turn on the masking on all cards and packet processors.

The **slot** option will turn on the specific slot, and when combined with the *np-id*, it will be enabled on the specific packet processor of a given slot.

Syntax: [no] load-balance mask ethernet *sa-mac* | *da-mac* | *vlan* | *etype* | *inner-vlan* | *isid* **all** | *slot* [*np-id*]

Syntax: [no] load-balance mask pbb *cust-l2-hdr* | *cust-ip4-ip6-hdr* **all** | *slot* [*np-id*]

Show commands

Use the **show load-balance mask-option** command to view the mask sub-options individually.

```
device# show load-balance mask-options ethernet
Mask Ethernet options -
Mask Source MAC is enabled on -
No Slots
Mask Destination MAC is enabled on -
No Slots
Mask Vlan is enabled on -
No Slots
Mask Inner-Vlan is enabled on -
No Slots
Mask ISID is enabled on
-
```

Use the **show load-balance mask-option pbb** command to view the pbb mask sub-options.

```
device# show load-balance mask-options pbb
Mask PBB options -
Mask PBB Customer L2 Header is enabled on -
All Slots
Mask PBB Customer IPv4/IPv6 Header is enabled on -
Slot 1
Slot 2 -   NPID 1
```

Sample configurations

You can configure a dual tagged and an ISID endpoint on the same port in different VPLS instances as shown below.

```
device(config)# router mpls
device(config-mpls)# vpls test5 105
device(config-mpls-vpls-test5)# vc-mode tagged
device(config-mpls-vpls-test5)# vlan 105 isid 1050
device(config-mpls-vpls-test5-vlan-105-isid-1050)# tag e 2/1
device(config-mpls-vpls-test5-vlan-105-isid-1050)# vpls test6 106
device(config-mpls-vpls-test6)# vlan 106 inner-vlan 1060
device(config-mpls-vpls-test6-vlan-106-inner-vlan-1060)# tag e 2/1
device(config-mpls-vpls-test6-vlan-106-inner-vlan-1060)# vpls test7 107
device(config-mpls-vpls-test7)# vc-mode tagged
device(config-mpls-vpls-test7)# vlan 106 isid 1060
device(config-mpls-vpls-test7-vlan-106-isid-1060)# tag e 2/1
device(config-mpls-vpls-test7-vlan-106-isid-1060)#
```

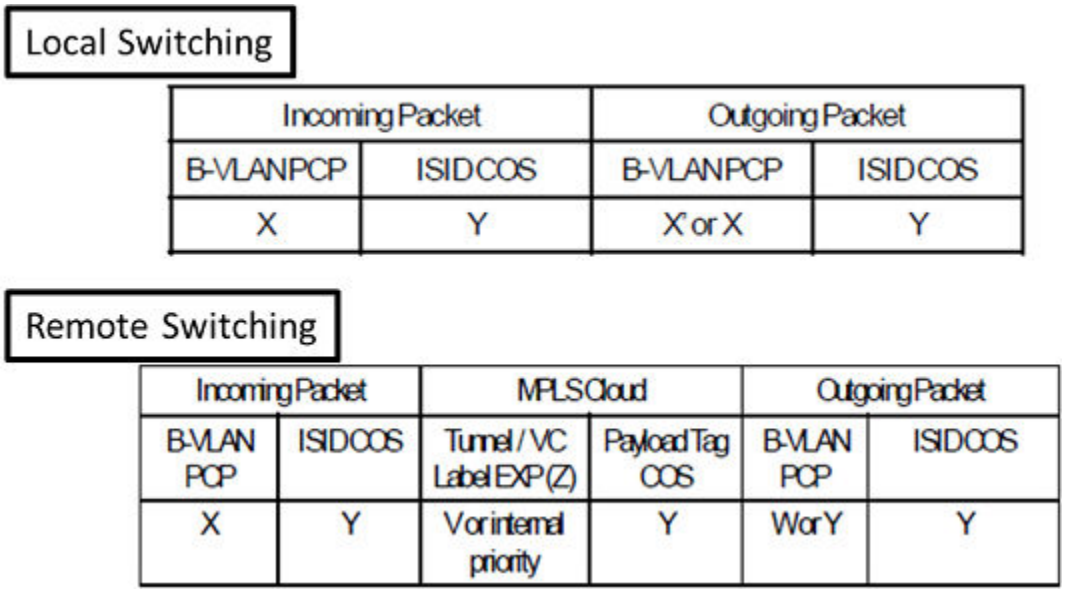
You can have two or more ISID endpoints in the same VPLS instance with different BVIDs on different ports as shown below.

```
device(config)# router mpls
device(config-mpls)# vpls test8 108
device(config-mpls-vpls-test8)# vc-mode tagged
device(config-mpls-vpls-test8)# vlan 108 isid 1080
device(config-mpls-vpls-test8-vlan-108-isid-1080)# tag e 2/1
device(config-mpls-vpls-test8-vlan-108-isid-1080)# vlan 109 isid 1080
device(config-mpls-vpls-test8-vlan-109-isid-1080)# tag e 2/2
device(config-mpls-vpls-test8-vlan-109-isid-1080)#
```

CoS with ISID to ISID endpoints

Figure 34 illustrates the behavior when using CoS with ISID to ISID endpoints.

FIGURE 33 CoS Treatment



LEGEND:

X - original B-VLAN PCP.

Y - original ISID COS.

X' - mapped PCP bits from internal priority (X contributes to internal priority) using PCP encode table.

V - mapped EXP bits from internal priority (X contributes to internal priority) using EXP encode table.

Z - incoming EXP bits as described by Tunnel/VC label column = V or internal priority.

W - mapped PCP from internal priority (Z contributes to internal priority) using PCP encode table.

The 'or' option in the Tunnel/VC label column is to differentiate when 'qos exp encode policy' is on (default) or off.

The 'or' option in the Outgoing B-VLAN column is to differentiate when 'qos pcp encode policy' is on (default) or off.

Local switching

The following configuration guidelines should be considered for local switching when using CoS with ISID to ISID endpoints.

- The Internal priority is mapped from the outer VLAN CoS in the incoming packet or incoming port's priority using the decode map.
- The outgoing outer VLAN CoS is mapped from internal priority using egress encoding map by default. The internal priority does not affect the outgoing ISID CoS.
- The outgoing outer VLAN CoS is same as the incoming packet's outer VLAN CoS if the **qos pcp encode-policy off** command is configured on the outgoing interface.
- The outgoing ISID CoS is the same as incoming packet's ISID CoS. This cannot be changed by any configuration.

Remote switching

The following configuration guidelines should be considered for remote switching when using CoS with ISID endpoints.

Local end point to remote peer (MPLS cloud)

The following configuration guidelines should be considered for remote switching when using CoS with ISID endpoints to a MPLS Cloud.

- The internal priority is mapped from outer VLAN CoS in the incoming packet or incoming port priority using the decode map.
- If VPLS or LSP CoS is configured, this value overrides internal priority.
- The outgoing tunnel or VC label EXP bits are mapped from internal priority using the egress encoding map by default.
- The outgoing tunnel or VC label EXP bits are set to the internal priority if the **qos exp encode-policy off** command is configured on the outgoing interface.
- The CoS of the payload tag is the same as the incoming packet ISID CoS and cannot be overridden by any other configuration.

Remote peer (MPLS cloud) to local end point

The following configuration guidelines should be considered for remote switching when using CoS with a MPLS Cloud to ISID endpoint.

- The internal priority is mapped from the tunnel or VC label EXP bits in the incoming packet or incoming port priority using the decode map.
- VPLS CoS will not override internal priority.
- The outgoing outer VLAN CoS is mapped from internal priority using egress encoding map by default. The internal priority does not affect the outgoing inner VLAN CoS.
- The outgoing outer VLAN CoS is the CoS in the payload tag if the **qos pcp encode-policy off** command is configured on the outgoing interface.
- The outgoing ISID CoS is the CoS in the payload tag.

End-to-end behavior

The following configuration guidelines should be considered for remote switching when using end-to-end behavior for CoS.

- The incoming outer VLAN CoS on the ingress router can affect the outgoing outer VLAN CoS on the egress router by default. This can be overridden by the VPLS or LSP CoS.
- The incoming ISID CoS on the ingress router is preserved in the outgoing outer VLAN COoS if the **qos pcp encode-policy off** command is configured on the outgoing interface of the egress router.
- The incoming ISID CoS on the ingress router is always preserved in the outgoing ISID COS of the egress router.

Configuration mismatch - forwarding behavior

The same ISID value must be configured on ingress and egress devices. Typically, when the MPLS egress device receives the first packet, the CAMs are programmed so that forwarding can be done in hardware. If there is a configuration mismatch, if the MPLS egress device has received a packet with an ISID that is not the same as the configured ISID, the software can detect the mismatch and not program the CAMs. This will cause the packets to be discarded.

NOTE

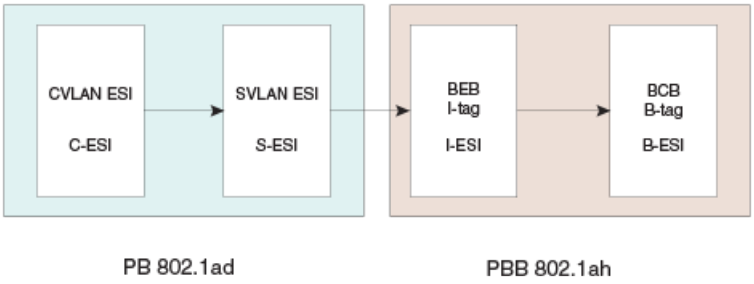
If the first packet arrived with the right ISID, the CAMs will be programmed as expected. Subsequently, if the packet arrives with an incorrect ISID, it will not be possible for the hardware to detect the mismatch and packets will be forwarded to the VPLS endpoint with the incorrect ISID.

When CPU protection is enabled, broadcast or unknown unicast packets will continue to be forwarded even when they arrive with an incorrect ISID.

Adding and removing VLANs and ESIs

Use Figure 35 and the tables in this section for information about adding and removing VLANs and ESIs. Refer to the Figure 35chapter for additional information.

FIGURE 34 Association of ESI and VLAN for different stages



Adding a VLAN to an ESI

TABLE 28 Adding a VLAN to an ESI

Elementn	ESI encapsulation	Configuration	Duplicates	Decision
Add CVLAN	CVLAN		• No duplicate within same ESI • No duplicates among other client ESIs of this ESI's parent	OK
Add SVLAN	SVLAN	No client ESI defined, No I-ESI provider ESI defined	• No duplicate within same ESI • No duplicates across all provider ESIs	OK
		No client ESI defined, No I-ESI provider ESI defined	• Duplicates among all S-ESI with no provider ESI	NOT OK
		I-ESI provider ESI defined	Duplicates among all S-ESI belonging to the I-ESI	NOT OK
Add SVLAN	SVLAN	Client ESI defined		
		Number of SVLANs in the ESI == 0	• No duplicates across all provider ESIs	OK
		Number of SVLANS in the ESI >= 1		NOT OK

Adding a source ESI to a target ESI

TABLE 29 Adding a source ESI to a target ESI

	Source ESI encapsulation	Target ESI encapsulation	Condition	Duplicates check	Decision
Add ESI (PB)	CVLAN	CVLAN			NOT OK
	CVLAN	SVLAN	Number of SVLANS in the ESI <= 1 1	1. No duplicates across CVLANs 2. No duplicates among other client ESIs	OK
	SVLAN	SVLAN			NOT OK
	SVLAN	CVLAN			NOT OK
PBB	SVLAN	ISID	Number of ISIDs in the ESI <= 1	Subject to duplicate check	OK
	SVLAN	SVLAN			NOT OK
	SVLAN	BVLAN			NOT OK
	ISID	BVLAN	Number of BVLAN in the ESI <= 1	Subject to duplicate check	OK
	ISID	ISID			NOT OK
	ISID	SVLAN			NOT OK

Deleting a VLAN

TABLE 30 Deleting a VLAN

VLAN type (ESI-encapsulation)	Actions
CVLAN	Delete CVLAN from ESI
SVLAN	Delete SVLAN from ESI
BVLAN	Delete BVLAN from ESI
ISID	Delete ISID from ESI

Deleting an ESI

TABLE 31 Deleting an ESI

	ESI-encapsulation	Actions
PB	CVLAN	• Remove ESI from any associated provider ESI's client ESI list • Delete all VLANs bound to the ESI • Delete ESI and return to free pool
PB	SVLAN	• Delete SVLANs bound to the ESI • Remove link with any client ESI and sever links of all client ESIs

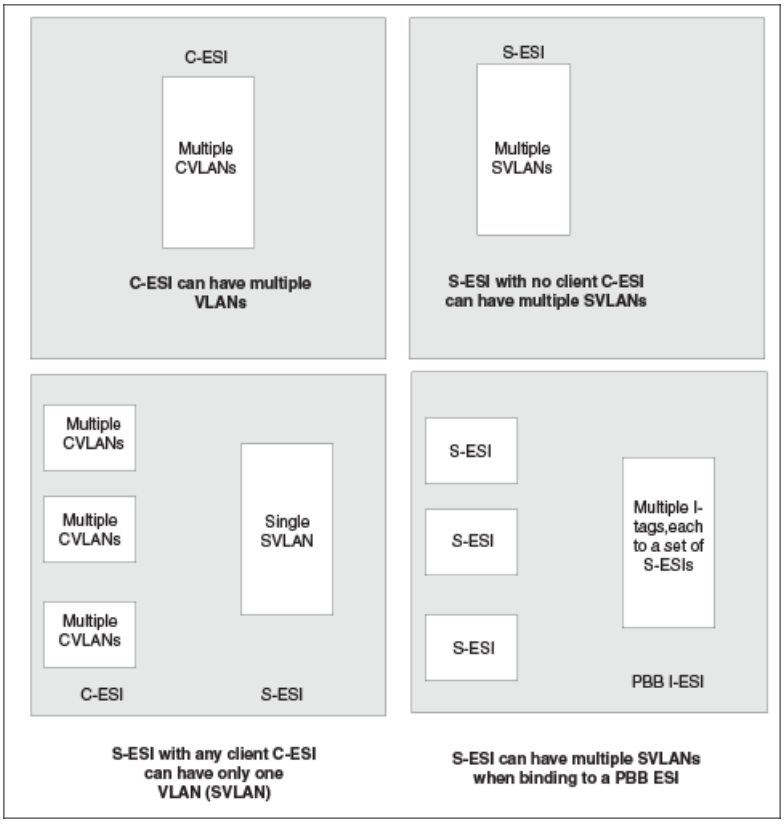
TABLE 31 Deleting an ESI (continued)

	ESI-encapsulation	Actions
		• Remove ESI from its provider ESI's (such as PBB) client ESI list • Delete ESI and return to free pool
PBB	SVLAN	• Remove ESI from any associated provider ESI's client ESI list • Delete all VLANs bound to the ESI • Delete ESI and return to free pool
PBB	ISID	• Delete ISID bound to the ESI • Remove link with any client ESI and sever links of all client ESIs • Remove ESI from its provider ESI's (such as PBB) client ESI list • Delete ESI and return to free pool
PBB	BVLAN	• Delete BVLANs bound to the ESI • Remove link with any client ESI and sever links of all client ESIs • Remove ESI from its provider ESI's (such as PBB) client ESI list • Delete ESI and return to free pool

Valid ESI configuration and interconnection modes

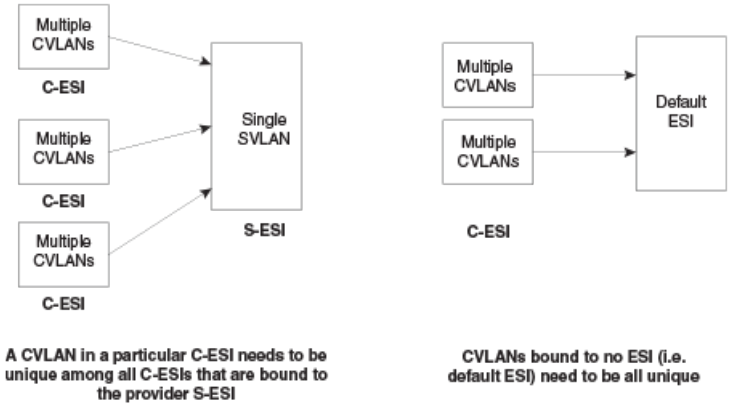
Figure 36 shows the allowable configurations for the ESI elements.

FIGURE 35 Allowable configurations for different ESI elements



Uniqueness requirements for VLANs

FIGURE 36 CVLAN uniqueness requirement



A CVLAN value to be added to a C-ESI must be unique:

- Among all CVLANs within the C-ESI

- If the C-ESI has a provider S-ESI (with or without a SVLAN), the CVLAN needs to be unique across all C-ESI that are clients of the provider S-ESI.

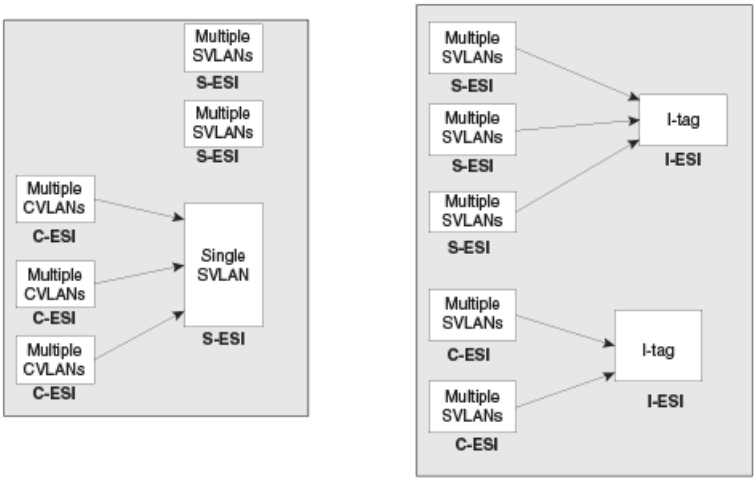
Default ESI

If CVLANs are not bound to a particular C-ESI, that is not entered with any ESI name they are assigned to a default ESI.

- CVLANs need to be unique among all CVLANs that are not bound to a particular C-ESI.

SVLAN uniqueness

FIGURE 37 SVLAN uniqueness for IEEE 802.1ad and IEEE 802.1ah configurations



PB 802.1ad: SVLANs bound to different provider S-ESIs need to be unique, whether or not connected with C-ESIs

802.1ah PBB: A SVLAN in a particular I-ESI needs to be unique among all S-ESIs that are bound to the provider I-ESI

IEEE 802.1ad (PB)

For IEEE 802.1ad (PB), because the provider only has 4K of SVLANs, the SVLANs must be unique among all S-ESIs

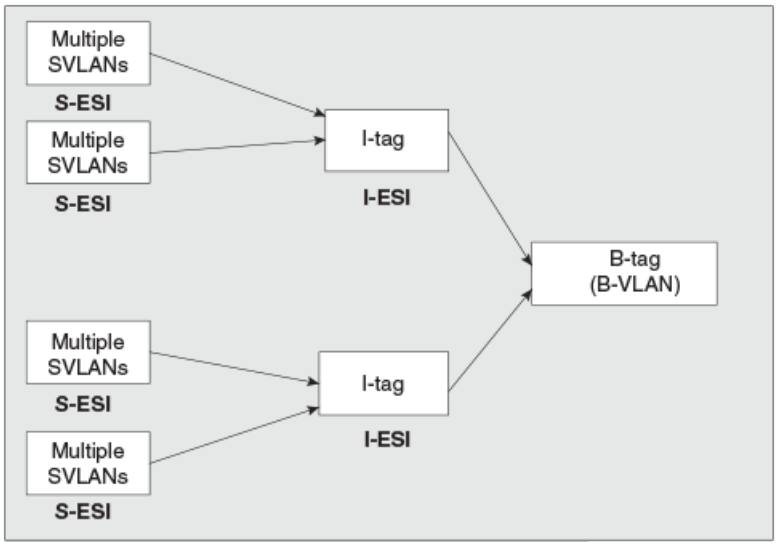
IEEE 802.1ah (PBB)

With IEEE 802.1ah (PBB), multiple SVLANs are mapped to I-tags. These SVLANs may belong to different providers and should be unique only among the SVLANs for the S-ESI to which the SVLAN belongs, and among all the S-ESIs which are clients for the I-ESI for the I-tag encapsulation.

I-tag uniqueness

I-tags must be unique across all I-ESIs that are associated with a single B-ESI provider ESI, as shown in Figure 39.

FIGURE 38 I-tag mappings



Provider Backbone Bridging (PBB) Networks for the XMR Series and MLX Series devices

Overview

The IEEE 802.1ah Provider Backbone Bridges (PBB) standard was developed to address the limitations of Provider Bridges (PB) and to add additional capabilities sought by Service Providers. When compared to a PB network, a PBB network deployment offers simplified operations, lower capital expenditures, and overall better scalability in terms of the number of supported customers. This section provides an overview of PBB for XMR Series and MLX Series, describing its advantages and examines common PBB deployment scenarios.

Provider Backbone Bridges

The Provider Backbone Bridges (PBB) standard, (IEEE 802.1ah), was developed to address the limitations of the Provider Bridges (PB) standard, (IEEE 802.1ad), and to add additional capabilities sought by Service Providers.

PB allows Service Providers to use a V-LAN identifier (VID) space separate from the customer VID (C-VID) space. PB adds a Service Provider VLAN Tag (S-TAG) containing a Service Provider VID (S-VID) to Ethernet frames (Figure 40). Because PB stacks a second VLAN tag to Ethernet frames, it is also known as "Q-in-Q," as a reference to the standard that originally defined VLAN tags, that is, IEEE 802.1Q, which is known as defining "Q" frames.

The S-VID field of the S-TAG is 12 bits long, which is the same length of a C-VID field of a customer VLAN Tag (C-TAG). Even though 12 bits can address up to 4096 distinct values, two values have special meaning and are reserved. Therefore, the Service Provider is limited to at most 4090 distinct S-VID values to identify service instances, that is, services or customers in a PB network. Another drawback is that PB frames are addressed by customer Media Access Control (MAC) addresses. This means that core Ethernet switches in a PB network have to learn all the source MAC addresses of all the customer frames traversing the core of the PB network. Thus, the size of the MAC address tables of core PB switches ultimately limits the number of customers that can be supported by a PB network.

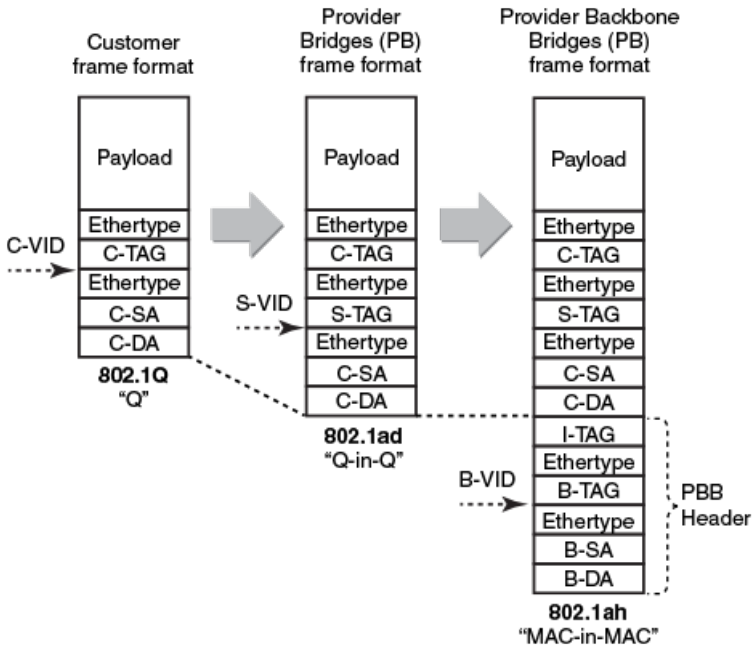
To address the above described PB shortcomings, PBB adds a hierarchy view to Ethernet by encapsulating PB frames with a PBB header (which becomes the equivalent of a "Service Provider MAC header") containing a Backbone Destination MAC Address (B-DA), Backbone Source MAC Address (B-SA), and two new tags (Figure 40), which are described later in this document. What makes the B-DA and B-SA "backbone" addresses is the fact that these are MAC addresses of Service Provider's PBB edge switches. An edge PBB switch encapsulates an ingress PB frame with a PBB header containing the destination MAC address of an appropriate egress edge PBB switch. The egress edge PBB switch removes the PBB header and forwards the frame to an attached PB network. Because PBB adds a PBB header containing new destination and source MAC addresses, it is also known as "MAC-in-MAC."

By adding the PBB header, PBB isolates the Service Provider and customer address spaces. This means that Ethernet switches in the core of the Service Provider network will no longer learn customer MAC addresses or use customer MAC addresses to forward customer frames to their destinations. This improves the scaling of the Service Provider network in terms of the number of supported customers, since the number of supported customers is no longer directly tied to the size of the MAC address tables of the core Ethernet switches.

In addition, the Service Provider network is now protected from customer network failures, since frame forwarding is now based on its own PBB header. Moreover, customers benefit from added security, since the customer's MAC addresses are no longer learned or used for frame forwarding decisions in the core of the Service Provider network.

As additional benefits to the Service Provider, PBB has the potential to simplify operations, e.g., by separating the customer and Service Provider addressing spaces, and to lower capital expenditures by reducing the cost of Ethernet switches used in the core of the network, since memory and processing power requirements are reduced by limiting MAC address learning to backbone MAC addresses.

FIGURE 39 Customer, PB, and PBB frame formats



The Backbone Service Instance Tag (I-TAG) contains a Backbone Service Instance Identifier (I-SID), which is 24 bits long. The I-SID field allows a Service Provider to identify up to 2 to the power of 24, that is, over 16 million, service instances. In other words, over 16 million services or customers can be uniquely identified using the I-SID field. Therefore, PBB's I-TAG allows for highly scalable services by eliminating the 4090 service instances limitation of PB.

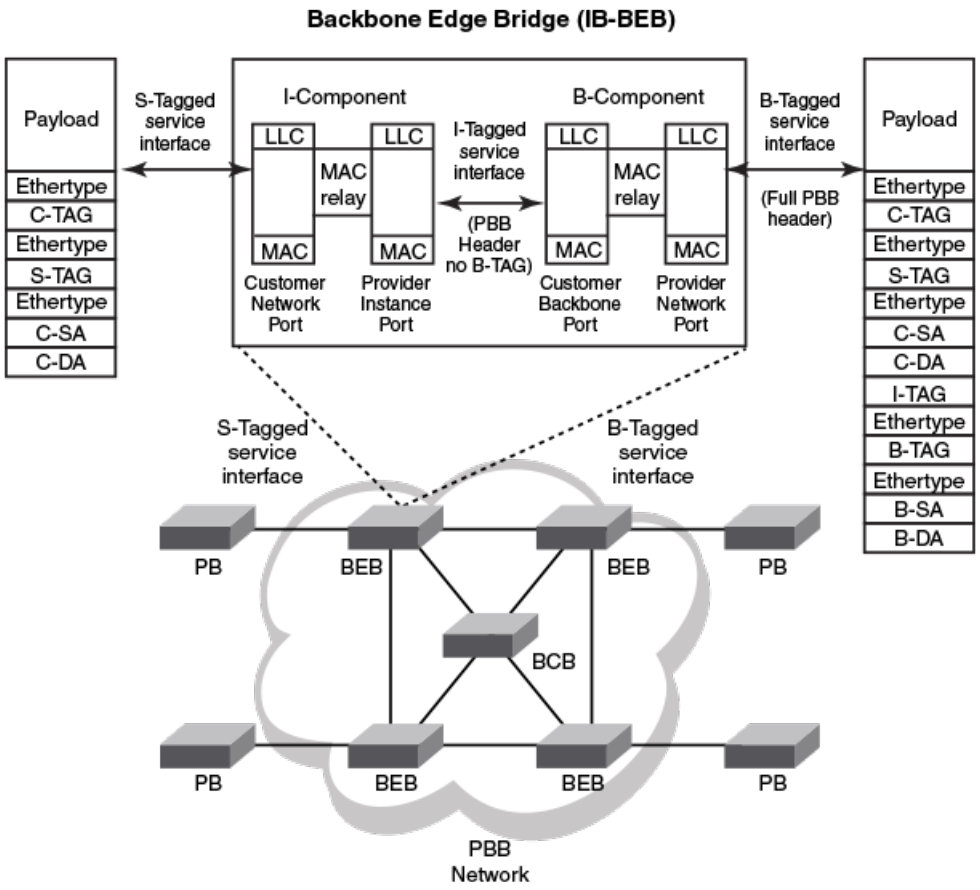
The semantics and the structure of the Backbone VLAN Tag (B-TAG) are identical to that of the PB S-TAG. The B-TAG was designed this way so that core PBB switches do not need to be aware of PBB. In fact, standard PB switches can be used in the core of a PBB network. Only the switches at the edge of the Service Provider PBB network need to be aware PBB.

A PBB network uses two types of bridges (Backbone Edge Bridge (BEB) operation on page 182): Backbone Edge Bridges (BEB) and Backbone Core Bridges (BCB). As explained above, the functionality required from a BCB is the same as a standard IEEE 802.1ad PB bridge. A BEB is used at the boundary of a PBB network to add and remove the PBB header.

Backbone Edge Bridge (BEB) operation

A BEB containing an I-Component and a B-Component is called an IB-BEB. The B-Component of an IB-BEB forwards packets towards the PBB network based on backbone MAC addresses (that is, it learns backbone MAC addresses), while the I-Component forwards packets towards the PB network based on the customer MAC addresses (that is, it learns customer MAC addresses). XMR Series and MLX Series only support IB-BEB. Neither support just the I-component BEB or just the B-component BEB.

FIGURE 40 Backbone Edge Bridge operation



Service instance

A Service Instance (SI) is also known as a bridging domain. It is defined as a single flooding domain where any traffic that requires flooding is always flooded to all endpoints under the same SI. When an unknown destination packet is received at a PBB endpoint it will cause flooding of this packet to all the endpoints (source port suppressed). The following types of endpoints defined here are supported by PBB.

Untagged endpoint

By default a port is configured with a Ethernet type (TPID) of 0x8100. If an untagged packet is received by the port, it is accepted and switched within the PBB instance to the desired destination MAC. If a 0x8100 tagged packet is received and the port is operating as untagged mode with the default TPID of 0x8100, the packet will be dropped as an invalid packet.

Port-based untagged endpoint

Port-based untagged endpoint is defined to always accept whatever packet is received by the given port whether there is any tag present in the packet received. As explained earlier, if the packet received on an untagged mode interface where the received packet contains the tag that matches the configured port Ethernet type value, it is dropped as an invalid packet. To achieve the desired behavior of treating the entire packet as an untagged packet (no tag stripping/insertion), the user must configure the Ethernet type of the port to a value that

will not match the Ethernet type of any packets sent to this port. This way, although the packet received may contain a tag its value will never match to the configured TPID value of the port (avoid setting it to a known protocol Ethernet type value) thus the tag will be treated as part of the customer payload. A suggested value of 0x9FFF can be used for this purpose as this value is not listed in the IEEE Ethernet Type Listing as a reserved value for a known protocol at the time of this document creation. Please always refer to IEEE Ethernet Type field public listing to avoid setting to a value that may conflict with a known protocol.

C-Tagged endpoint

C-tagged endpoint is defined as an endpoint that considers all packets received with the top most tag being the Customer Tag (C-Tag). This is achieved by configuring a tagged endpoint under the PBB instance. The user must configure the Ethernet type of the port to match the C-Tag TPID value of the received packets. Packets sent out through the C-tagged endpoint will have a C-tag with C-VID 100 added at egress to the endpoint.

Example 1:

If the packet received has a C-VID of 100 and the destination is another C-Tagged endpoint with C-VID 200, when the packet exits the destination C-Tagged endpoint, it will have the C-VID of 200. The original C-VID 100 is "translated" in this case.

Example 2:

If the packet is destined to a port based untagged endpoint, the packet will be sent out without the original C-VID 100 tag because it was removed at the ingress of the C-tagged endpoint.

Example 3:

If the packet is destined to an IB-Tagged endpoint, the original tag that contains the C-VID of 100 is stripped at ingress and a PBB header (contains the B-MACs, B-Tag and I-Tag) is inserted when the packet exits through the IB-Tagged endpoint. Optionally, if S-VLAN keep mode is configured, an S-Tag will also be inserted. More details about S-VLAN keep mode is discussed in a later section.

S-Tagged endpoint

S-Tagged endpoint is defined as an endpoint that considers all packets received with the top most tag being the Service provider Tag (S-Tag). The user must configure the Ethernet Type value of the port to match the S-Tag TPID value of the packets received on the specified port. The packet processing operation whether it is S-Tagged or C-Tagged is equivalent. When flooding is performed, a valid packet received through the S-Tagged endpoint will be flooded to all endpoints configured under the same SI, regardless of the egress endpoint tag type.

Dual-tagged endpoint

Dual-tagged endpoint is defined as an endpoint that expects two tags (Q-in-Q) where the top most tag must match the configured Ethernet Type value of the port and the inner tag must match the default TPID of 0x8100. The dual-tagged endpoint configuration is NOT supported with PBB on XMR Series and MLX Series.

IB-Tagged Endpoint

An IB-Tagged endpoint represents all frames on a physical port with a particular B-VID and a particular I-SID. When such a packet is received from the IB-endpoint destined to another local endpoint of the same PBB instance, the PBB header will be stripped and a destination tag may be inserted accordingly based on the exiting endpoint configuration.

Example 1:

If the packet is destined to a C-Tagged endpoint with C-VID 100, the original PBB header is stripped and a C-Tag with C-VID 100 is inserted as the packet exits the C-tagged endpoint. The TPID value will be configured for the Ethernet type value of the exit port.

Example 2:

If the packet is destined to an S-Tagged endpoint with S-VID 300, the original PBB header is stripped and an S-Tag with S-VID 300 is inserted as the packet exit the S-tagged endpoint. The TPID value will be configured for the Ethernet type value of the exit port.

Example 3:

If the packet is destined to a port based untagged endpoint, the original PBB header is stripped. No additional tag will be inserted as the packet exit the untagged endpoint.

Example 4:

If the packet is destined to another IB-Tagged endpoint with the same B-VID and ISID value of the received PBB frame, the original PBB frame is NOT touched (the PBB header remains unchanged) and will be send out towards the specified outgoing interface as determined by the BEB's switching logic.

IB-Tagged Endpoint with S-VLAN Keep mode

The system can be configured with S-VLAN keep mode which indicates that packets received at the IB-Tagged endpoint from the PBB network will contain an S-Tag after the I-Tag field (see Backbone Edge Bridge (BEB) operation on page 182). When such a packet is destined to a local endpoint, in addition to stripping the PBB header the S-Tag is also removed.

Example 1:

If the packet is destined to a C-Tagged endpoint with C-VID 100, the original PBB header and the S-Tag is stripped and a C-Tag with C-VID 100 is inserted as the packet exit the C-tagged endpoint. The TPID value will be what is configured for the exiting port's Ethernet type value.

Example 2:

If the packet is destined to an S-Tagged endpoint with S-VID 300, the original PBB header and the S-Tag is stripped and an S-Tag with S-VID 300 is inserted as the packet exit the S-tagged endpoint. The TPID value will be what was configured for the exiting port's Ethernet type value. The S-VID of the S-Tag in the original PBB frame is "translated" into the S-VID of the destination S-Tagged endpoint. The PCP/DEI is preserved or modified depends on the PCP encode and decode setting of the receiving/destination ports.

Example 3:

If the packet is destined to another IB-Tagged endpoint with the same B-VID and ISID value of the received PBB frame, the original PBB frame is NOT touched (the PBB header and S-Tag remains unchanged) and will be send out towards the specified outgoing interface as determined by the BEB's switching logic.

S-VLAN VID Value setting when packet exit IB-Tagged Endpoint with S-VLAN Keep mode

If a packet is received from a C-Tagged or S-Tagged endpoint destined towards an IB-Tagged endpoint under S-VLAN keep mode, the S-VLAN VID value will depend on the original VLAN ID received. If the packet was received from an untagged endpoint, then the untagged VLAN value of the port will be used as the S-VLAN VID as it goes out of the IB-Tagged endpoint.

Example 1

If the packet is received from a C-Tagged endpoint with C-VID 100 and destined to an IB-tagged endpoint, the original tag that contains the C-VID of 100 is stripped and a PBB header (contains the B-MACs, B-Tag, I-Tag, and S-Tag) is inserted when the packet exits through the IB-Tagged endpoint. The S-Tag S-VLAN VID in the PBB header will have a value of 100.

Example 2

If the packet is received from a S-Tagged endpoint with S-VID 200 and destined to an IB-tagged endpoint, the original tag that contains the S-VID of 200 is stripped and a PBB header (contains the B-Macs, B-Tag, I-Tag, and S-Tag) is inserted when the packet exits through the IB-Tagged endpoint. The S-Tag S-VLAN VID in the PBB header will have a value of 200.

Example 3

If the packet is received from an Untagged endpoint with default port VLAN set as 150 and destined to a IB-tagged endpoint, a PBB header (contains the B-Macs, B-Tag, I-Tag, and S-Tag) is inserted when the packet exits through the IB-Tagged endpoint. The S-Tag S-VLAN VID in the PBB header will have a value of 150.

Customer to ISID mapping

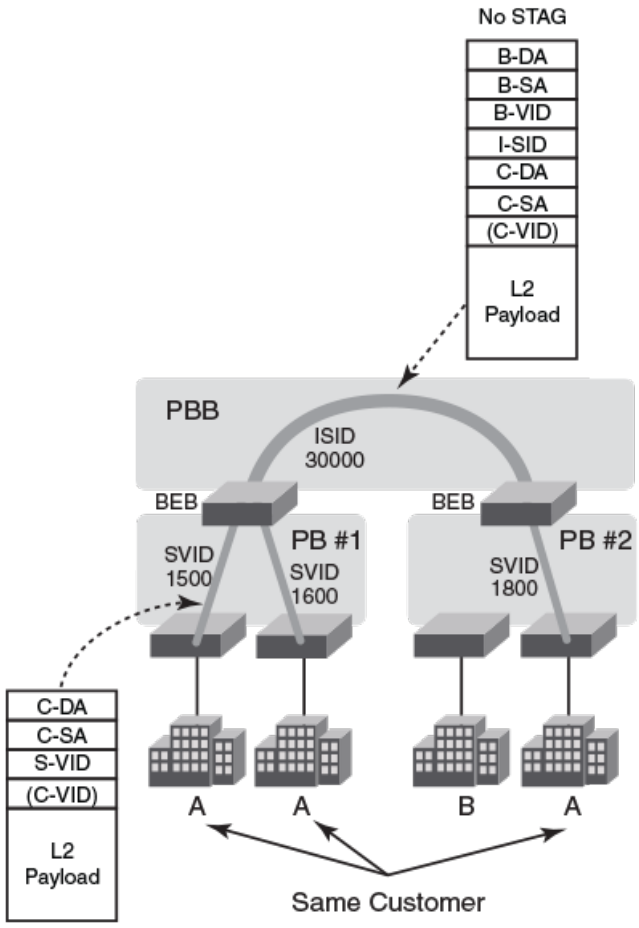
There are two types of Customer to ISID Mapping. 1:1 Customer to ISID Mapping and N:1 Customer to ISID Mapping. XMR Series and MLX Series only support one single flooding domain per SI regardless of which mapping is chosen. 1:1 Customer to ISID mapping is supported by default. XMR Series and MLX Series will also support S-VLAN Keep Mode to inter-operate with CES 2000 Series and CER 2000 Series and other vendors on the same PBB network that supports the N:1 Customer to ISID mapping.

Regardless of which type of mapping is chosen, XMR Series and MLX Series will always strip the incoming S-Tag and add the S/C-Tag as it goes out on the BEB's S/C-Tag endpoint based on what is configured and not based on what is received. Optionally the S-Tag TPID value can also be "translated" based on the exiting port's configured Ethernet Type value.

1:1 Customer to ISID mapping

With 1:1 customer to ISID mapping, a single customer is mapped to an ISID value. There may be many S-Tag endpoints configured under the same SI, but they are all considered as belonging to one customer. The BEB strips the S-Tag at the ingress of the BEB and inserts the S-Tag at the egress of the BEB. There is no S-TAG in the PBB frame when it traverses the PBB network. Any PBB instance configured in the system is considered as one SI which implies one single flooding domain. 1:1 Customer to ISID mapping shows an example of a 1:1 customer to ISID mapping.

FIGURE 41 1:1 customer to ISID mapping



N:1 Customer to ISID Mapping Interop (S-VLAN Keep Mode)

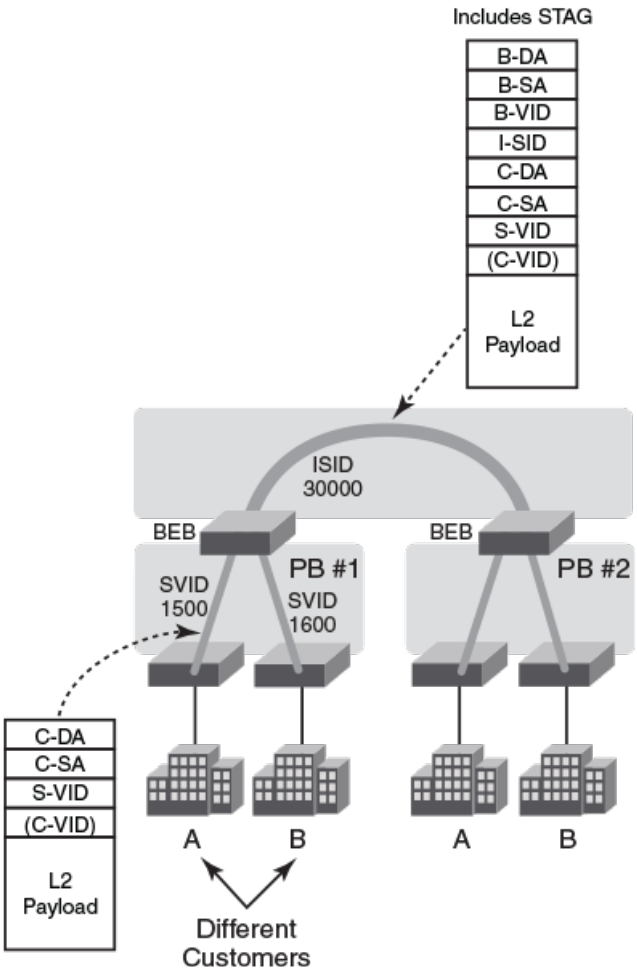
The benefits of N:1 customer to ISID mapping include the following:

- Multiple customers are mapped to the same SI to use the same ISID.
- The S-VLAN at the ingress is carried all the way to the egress side of the PBB network.
- When flooding of a packet is required, it only floods to the endpoints that have the same S-VID.

XMR Series and MLX Series supports S-VLAN Keep Mode. This deviates from true N:1 Customer to ISID mapping in that it will NOT have different flooding domains based on the S-VLAN, but continue to use one single flooding domain per VPLS instance. When flooding of packets is required, it will always flood to all the endpoints (source port suppressed) within the same VPLS instance (SI).

If the S-VLAN keep mode is desired, you must ensure that all the XMR Series and MLX Series nodes on the same PBB network are all configured with the S-VLAN keep mode. Once a BEB is configured with S-VLAN keep mode any PBB frames received without the proper S-Tag present in the PBB frame will be treated as invalid packet and discarded.

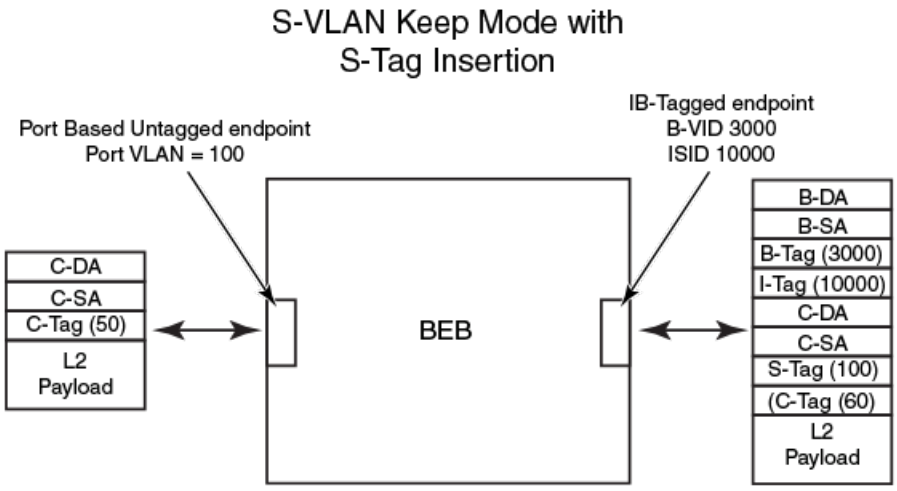
FIGURE 42 S-VLAN keep mode



Port Based Untagged Frame with S-Tag Insertion

With S-VLAN keep mode it is also possible to insert an S-Tag on top of whatever tag(s) the original packet may already have when it goes out into the PBB network. This is done by configuring a port-based untagged endpoint with the desired S-VID as the Port's default VID and send port based untagged packets into this port. When the corresponding packet (shown in Figure 44 as a C-Tagged packet) is switched towards an IB-endpoint into the PBB network, the BEB will insert the PBB header as well as inserting an S-Tag with the S-VID set to the ingress port's configured VLAN value. The S-Tag TPID will be set to whatever was configured as the system-wise S-Tag Ether Type value.

FIGURE 43 S-VLAN keep mode with S-Tag insertion



PBB packet switching

There are several scenarios on how a packet is switched within the BEB.

- Packet may be switched between a local endpoint and an IB endpoint of the SI.
- Packet may be switched between two local endpoints of the SI.
- Packet may be switched between two IB endpoints where the BEB is acting as a BCB.
- Unknown C-DA packets flooding handling.

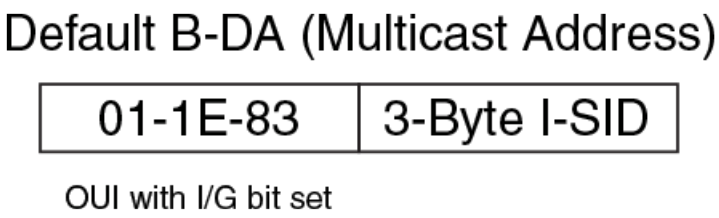
PBB Packet Received at the IB endpoint

When the sender does not know where the C-DA is located, the packet is flooded with the configured flooding B-DA over the B-VLAN configured for the respective SI. If the sender knows which BEB owns the intended C-DA (C-DA was learnt and associated with a particular BEB's B-MAC), it will send the packet to that destination BEB.

Unknown Destination PBB Packet Handling

When a BEB receives a PBB packet with the default backbone multicast destination address as its B-DA and the corresponding ISID is of interest (it has configuration of SI that has matching BVLAN and ISID), it will examine the packet to determine if the associated C-DA in the PBB packet is a known MAC address. If it is known, it will forward the packet to the port that learned the C-MAC. If the C-DA is also unknown to the BEB that received this PBB packet, it will need to flood this packet to all its local endpoints as well as to all the IB-endpoints of the corresponding SI. When such flooding occurs, the packet destined towards the PBB networks will retain the original received PBB header without any modification. The default backbone multicast destination MAC address is constructed by concatenating the three octet OUI 00-1E-83 with the three octet I-SID (and asserting the I/G bit to signify a group MAC address). The resulting B-DA is shown in Figure 45.

FIGURE 44 Unknown Destination PBB Packet Handling



If the PBB packet received was for an ISID that the BEB does not care about (no SI configured that matches the B-VLAN and ISID value of the PBB packet), then the packet is forwarded based on the B-Tag and the B-MACs as a regular Layer 2 packet. No attempt will be made to look into the inner MAC of the PBB packet.

NOTE
No validation is made between the 3-byte I-SID value within the default B-DA multicast address to the PBB header's ISID value. XMR Series and MLX Series will always use the PBB header ISID value when a switching decision is made.

Known Destination PBB Packet handling

When ingress BEB receives a packet with known C-DA that it learnt from one of the IB-endpoint, it will forward the received packet towards the IB-endpoint that it learnt the C-DA. The PBB header will contain the B-VID and ISID that correspond to what was configured for the corresponding SI. The B-DA will be the B-MAC associated with the learnt C-DA. The B-SA will be set to the chassis base MAC address.

Unknown Destination PB packet Handling

When a packet is received at a local endpoint (S/C-Tagged or Port based untagged), if the C-DA is unknown, it will also require flooding of this packet towards all local endpoints as well as towards the PBB network via the IB-endpoints.

If a flooding B-DA was configured for this SI, when it floods the packet to the IB-endpoints, it will use the configured flooding B-DA. Otherwise, it will use the default backbone multicast destination MAC address when it floods the packet towards the PBB network.

BEB Acting as BCB

When a BEB receives a PBB packet with the B-DA and it is not its own chassis MAC, then the packet must be destined for some other PBB node within the PBB network. In this case, the BEB will act as a BCB and Layer 2 switch the received packet based on the B-MACs and the B-VLAN towards the next PBB node. The ISID portion of the packet is ignored and treated as part of the payload.

PBB MAC Learning

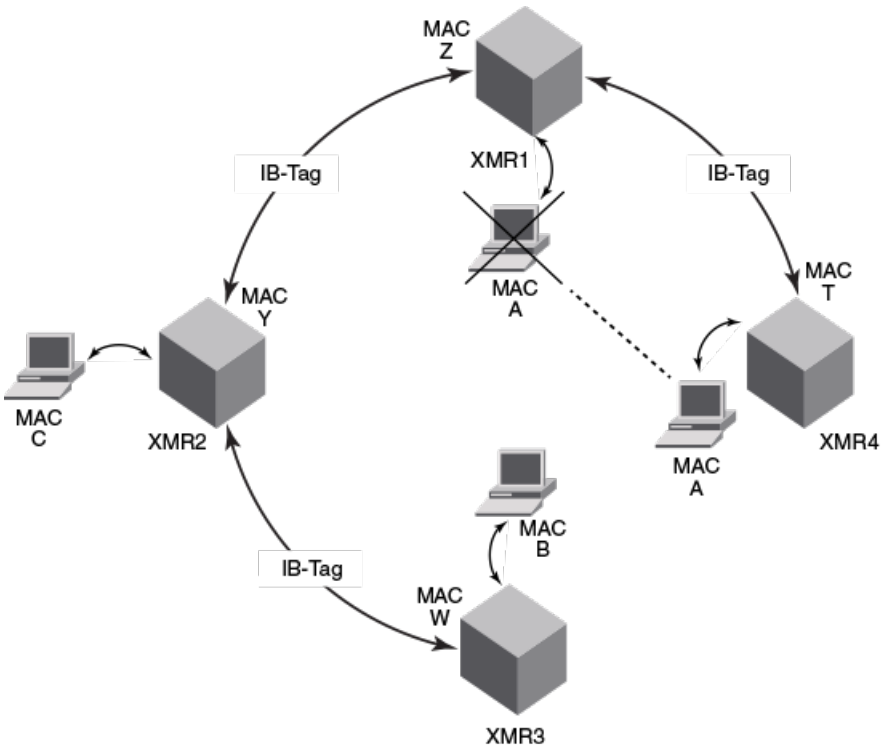
With the introduction of PBB, there will be some MAC interactions for the B-MACs involved between Layer 2 (for B-VLAN) and PBB.

MAC Learning for PBB Packets

PBB packets are MAC-in-MAC packets. There are two types of MAC learning involved depending on whether the ISID involved in the received PBB packet is of any interest. If there is no such SI configured for the PBB packet's B-VLAN and ISID, only the outer MAC (B-SA) is learned via the regular Layer 2 MAC management for the corresponding B-VID. A regular Layer 2 endpoint must be configured for each B-VLAN used as an IB-endpoint in a PBB instance.

If the PBB packet received has an SI configured that matches the B-VLAN and ISID of the packet's PBB header, the inner MAC (C-SA) will be learned by the corresponding PBB instance and an association is made between the C-SA to the B-SA of the received PBB packet. The B-SA association under PBB is primarily used to program the hardware so that it knows how to set up the PBB header for packets destined to the C-MAC that was associated with a particular B-MAC. The B-DA of such packet forwarding will be set up according to this association. Another usage of this B-SA association under PBB is to enable detection of B-MAC movement where the C-MAC association to B-MAC may have changed from B-MAC-Z to B-MAC-T in Figure 46.

FIGURE 45 MAC Learning for PBB packets



L2 MAC Installment by PBB

When packet is being switched by the hardware based on the inner MACs (C-DA and C-SA), the outer MACs (B-DA and B-SA) will stop hitting the regular Layer 2 CAMs programmed for the associated B-SA. So without any intervention from PBB, these B-MACs once learned by the corresponding Layer 2 B-VLAN, will start to age out. Once aged out, if another PBB packets is received that is being switched based on the outer B-MACs, will end up causing re-learning those B-MACs again and unnecessary flooding even though there may exist a constant traffic for some packet flow that is being switched based on the inner MACs that associate with these aged out B-MACs.

In order to address this Layer 2 B-MAC aging out issue, PBB installs the associated B-MACs to the appropriate Layer 2 VLAN MAC space and marking them similar to static MACs where aging is disabled. These installed B-MACs are dynamically installed based on the C-MAC to B-MAC association. These installed B-MACs will be flush-able by the following methods:

- Explicit CLI command such as "clear mac".
- Topology Changes Notification (TCN) that caused flushing such B-MACs.

Once the B-MAC is cleared by the CLI method or TCN method, the B-MAC will cease to exist in the Layer 2 VLAN space until either PBB re-install it again due to C-MAC relearned and re-associated with the B-MAC or B-MAC is learnt by Layer 2 B-VLAN based on BCB forwarding action.

When the last C-MAC associated with the corresponding B-MAC is aged out, the corresponding B-MAC that was previously installed will be re-programmed to allow aging and if there were no Layer 2 traffic hitting this B-MAC, it will be aged out by Layer 2 naturally.

Temporarily Learning of C-MACs on Transit BEB

Although one of the main purposes of deploying the PBB network is to reduce the learning of C-MACs to a smaller set, the BEB may momentarily learn the C-MACs that does not belong to the BEB due to unknown flooding traffic using the default backbone multicast destination MAC address. Once the destination becomes known, the sender will stop using the default mcast flood MAC and use the learned B-MAC, those C-MACs that were temporarily learned on the transit BEB will start to age out.

PBB PCP/DEI Setting

Behaviors of the B-Tag PCP/DEI, I-Tag PCP/DEI and S-Tag PCP/DEI settings are described in the following sections.

B-Tag PCP Setting

The B-Tag PCP setting has two options:

1. Encode PCP On - This is the default case where the B-Tag PCP is derived based on the internal priority modified by the selected PCP encode table.
2. Encode PCP Off - When this is configured on the egress IB endpoint interface, the B-Tag PCP value will be set to a fixed value depends on the "Forced-PBB-PCP" configuration on the corresponding B-VLAN:
   a) **Forced-PBB-PCP** is set on B-VLAN - This allows user to force the B-Tag PCP setting. The PBB header B-Tag PCP value will be set to the specified forced-PBB-PCP value.
   b) **Forced-PBB-PCP** is NOT configured for the B-VLAN - This will cause the PBB header B-Tag PCP value set to "0".

NOTE
If a packet ingress through a C-Tagged endpoint is destined to a remote PBB node, the C-Tag PCP value will be used the same way as described above.The Internal priority setting is based on the original packet's PCP value operated by the port's PCP decode table, as explained in the *Extreme NetIron QoS and Traffic Management Configuration Guide*.The **forced-PBB-PCP** configuration option has no effect on the regular Layer 2 traffic using the same B-VID. For pass through PBB packets (BEB acting as a BCB case), the B-Tag PCP value is not modified at all. The existing "priority" configuration on the B-VLAN operates independently from the **forced-PBB-PCP** . It governs on the PCP setting of the regular Layer 2 traffic as well as pass through PBB traffic.

B-Tag DEI Setting

The B-Tag DEI setting has the following options:

1. Encode PCP On - This is the default case where the B-Tag DEI is derived based on the internal drop precedence and the **qos use-dei** setting configured on the egress IB interface.
   a) "qos use-dei" On - The B-Tag DEI setting is based on the internal drop precedence set up at the ingress.
   b) "qos use-dei" Off - The B-Tag DEI setting is set to "0" regardless of what the internal drop precedence may be.

2. Encode PCP Off - When this is configured on the egress IB endpoint interface, the B-Tag DEI value will always be "0".

NOTE
There is no force option for the DEI setting.

NOTE
If the packet ingress is through a C-Tagged endpoint destined to a remote PBB node, the C-Tag CFI value will be used as described above.

NOTE
Internal drop precedence on the ingress is derived based on the original packet's PCP value operated by the PCP decode table and optionally merged with the packet's DEI/CFI bit if the "qos use-dei" is set on the ingress port.

I-Tag PCP Setting

I-Tag PCP is always taken from the PCP value that is received from the ingress. In most cases this is the SVLAN PCP value.

NOTE
If the packet ingress is through a C-Tagged endpoint destined to a remote PBB node, the C-Tag PCP value will be used as the I-Tag PCP value in the PBB header.

NOTE
If the packet ingress is through an untagged endpoint destined to a remote PBB node, the PCP value of "0" will be used as the I-Tag PCP value in the PBB header.

I-Tag I-DEI Setting

Packets ingress through an S-Tagged endpoint destined to a remote PBB node will have the DEI bit preserved from the received packet copied unto the PBB header I-Tag I-DEI field.

NOTE
If the packet ingress is through a C-Tagged endpoint destined to a remote PBB node, the C-Tag CFI value will be used as the I-Tag I-DEI value in the PBB header.

NOTE
If the packet ingress is through an untagged endpoint destined to a remote PBB node, the I-Tag I-DEI value in the PBB header will be "0".

S-Tag PCP/DEI Setting

When a packet is received from an IB endpoint, there are PCP/DEI in the B-Tag, PCP/DEI in the I-Tag and optionally PCP/DEI in the S-Tag if S-VLAN keep mode is configured. When this packet is switched to a local S-Tagged endpoint, the PCP/DEI setting will be set according to the rules documented in the following sections.

S-Tag PCP Setting

The packet internal priority for an ingress IB endpoint is derived based on the ingress PCP decode table.

1. S-VLAN keep mode is set - The PBB packet's original S-VLAN's PCP value.

2. S-VLAN Keep mode is not set - The original packet's I-Tag PCP value.

   Depending on the encode PCP policy configured on the egress S-Tagged endpoint, the PCP value will be derived as following:

S-Tag DEI Setting

The packet internal drop precedence for an ingress IB endpoint is derived based on ingress PCP decode table operated on:

1. S-VLAN keep mode is set - The PBB packet's original S-VLAN's PCP value and merged with S-VLAN's DEI bit if "qos use-dei" is also configured at the ingress IB endpoint.
2. S-VLAN Keep mode is not set - The original packet's I-Tag PCP value and merged with I-Tag's I-DEI bit if "qos use-dei" is also configured at the ingress IB endpoint.

   Depending on the encode PCP policy configured on the egress S-Tagged endpoint, the DEI value will be derived as following:

Configuring PBB

This section discusses the limitations and configuration commands required to configure PBB.

Limitations

- Disabling PBB - When PBB is configured, and you want to disable PBB, you will have to remove all endpoint and b-dest-mac configurations. The alternative is to delete the VPLS instance before disabling PBB on the instance.
- Ensure the BVLAN configured for the vlan-isid endpoint is already configured as part of the Layer 2 VLAN for the given interface.
- If PBB is not configured before the endpoint is configured, the user will need to delete all endpoints before configuring PBB.
- A dual tagged endpoint configuration is not allowed under the PBB configuration.
- Auto-discovery configuration is not allowed under the PBB configuration.
- Multicast configuration is not allowed under the PBB configuration.
- VPLS-MTU configuration is not allowed under the PBB configuration.
- VPLS peer configuration is not allowed under the PBB configuration.
- VPLS-local-switching is turned on when PBB is configured. Since PBB support is internally implemented using VPLS local switching, the local switching feature cannot be turned off when PBB is configured for the given VPLS instance.
- VC-mode configuration is not allowed under the PBB configuration.
- ISIDs cannot be reused across VLANs under the PBB configuration.

Configuring PBB

Note that the configuration of PBB on the MLX Series and XMR Series is done under the VPLS CLI constructs, since PBB on the MLX Series and XMR Series is internally supported similar to Local VPLS. However, PBB on the MLX Series and XMR Series does not actually use MPLS. PBB on the MLX Series and XMR Series does not generate any MPLS signaling and does not use any MPLS protocols. All PBB traffic conforms to the PBB standard.

PBB configuration is not enabled by default and must be enabled per VPLS instance. The optional Backbone Destination MAC Address configuration under PBB is used for PBB connections, which do not require flooding in the PBB core. The customer may use this capability to set the default B-DA to the address of the destination BEB of the point-to-point connection.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)# b-dest-mac 2010.0345.4232
vlan 10
tagged ethe 4/1
vlan 200 isid 20000
tagged ethe 2/1
```

Enabling a new PBB configuration

The following command enables a PBB instance.

```
device(config)#router mpls
device(config-mpls)#vpls vinst 2000
device(config-mpls-vpls-vinst)# pbb
```

Syntax: [no] pbb

PBB Backbone Destination MAC Address Configuration

Configuring a Backbone destination MAC address for a PBB instance allows the flood traffic to be directed towards the specified BEB node instead of using the default backbone multicast destination MAC address. This is used to create point-to-point connections (using a Unicast MAC address).

```
device(config)#router mpls
device(config-mpls)#vpls vinst 2000
device(config-mpls-vpls-vinst)# pbb
device(config-mpls-vpls-vinst-pbb)#b-dest-mac 0001.0001.0003
```

Syntax: [no] b-dest-mac *Ethernet MAC address used for point topoint*

A multicast MAC address will be allowed only if it has the well-known PBB MAC prefix of 01-1E-83.

The following MAC addresses are not allowed:

- A broadcast MAC address is disallowed.
- A zero MAC address is disallowed.
- The own chassis mac in disallowed.

Configuring PBB policy

Any global PBB specific configuration is configured using this sub-mode.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)#
```

Syntax: [no] pbb

System-wide SVLAN Tag Type

The **stag-type** command specifies the Etype used for S-tagged packets in SVLAN-keep mode. This Etype is global.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)#
device(config-mpls-vpls-policy-pbb)#stag-type 8888
```

Syntax: [no] stag-type *hex*

The default Stag-type is 0x88A8.

SVLAN Keep Mode Configuration

To configure SVLAN keep mode, use the **svlan-keep** command. Once the SVLAN keep mode is configured, the system configured SVLAN Etype value will be used to enforce that the SVLAN is present in the IB packets received.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)#
device(config-mpls-vpls-policy-pbb)#svlan-keep
```

Syntax: [no] svlan-keep

```
Before configuring the SVLAN keep mode, ensure that the system-wide SVLAN tag type is already set to the
desired value. If there is a mismatch, the packet will be discarded.
```

Vlan force-pbb-pcp option

The **force-pbb-pcp** command configures the PCP value per VLAN. This command applies only to IB tagged endpoints, whereas the VLAN priority applies to Layer 2 VLANs.

```
device(XMR12)#conf t
device(XMR12)(config)#vlan 60
device(XMR12)(config-vlan-60)#force-pbb-pcp 5
```

Syntax: [no] force-pbb-pcp *value*

The *value* variable set the PCP value per VLAN and can be configured in the range from 1 to 7.

Show Commands

The following section discusses available show commands.

Show mpls vpls detail

The **show mpls vpls detail** command displays information about the operation state of the VPLS instance in regard to the local endpoints.

```
device(XMR12)#show mpls vpls detail
VPLS 1, Id 1, Max mac entries: 8192
 PBB
 Total vlans: 2, Tagged ports: 2 (2 Up), Untagged ports 0 (0 Up)
 IFL-ID: 4096
  Vlan 300
   Tagged: ethe 1/5
```

```
  Vlan 3000 isid 40000
   Tagged: ethe 4/1
CPU-Protection: ON,  MVID: 0x001,  VPLS FID: 0x0000a00c
 Local Switching: Enabled
 Extended Counter: ON
```

Show mac

The **show mac** command displays the MACs learned by Layer 2.

```
device(XMR12)#show mac
Total active entries from all ports = 1
Type Code - ST:Static SEC:Secure 1x:Dot1x NA: NotAvail A:Allow D:Deny IB: Installed B-MAC
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
MAC Address     Port      Age      VLAN    Type
0012.f2f7.3b00  4/19         0         10           IB
```

Syntax: show mac

Show mac vpls

The **show mac vpls** command displays the MACs learnt under all VPLS instances.

```
device(XMR12)#show mac vpls
Total VPLS mac entries in the table: 2 (Local: 2, Remote: 0)
VPLS       MAC Address    L/R/IB Port  Vlan(In-Tag)/Peer ISID     Age
====       ===========    ====== ====  ================= ====     ===
1          0000.0404.0000 IB     1/2   300               30000    0
1          0000.0403.0000 L      1/1   200               NA       0
```

Show mac vpls X HHHH.HHHH.HHHH

To view the detail of the B-MAC info, you must enter the command **show mac vpls x** , where *hhhh.hhhh.hhhh* is the C-MAC that has a B-MAC association.

```
device(XMR12)#show mac vpls 1 0000.0404.0000
    VPLS: 1         MAC:  0000.0404.0000     Age:     0
    Local MAC       Port: 1/2                VLAN:    300 ISID:    30000
    Associated B-MAC: 0000.B000.0201
    Trunk slot mask: 0x00000000
```

Syntax: show mac vpls x *hhhh.hhhh.hhhh*

Show mac vpls X b-mac HHHH.HHHH.HHHH

The **show mac vpls X b-mac** command is used to display the C-MACs learned per B-MAC per VPLS instance.

```
device(XMR12)#show mac vpls 1 b-mac 0000.B000.0201
Total VPLS mac entries associated with b-mac 0000.B0000.0201: 2
MAC Address    Port  Vlan(In-Tag)/Peer ISID      Age
===========    ====  ================= ====      ===
0000.0404.0000 1/1   300               30000     10
0000.0408.0000 1/1   200               30000     20
```

Syntax: show mac vpls X b-mac *hhhh.hhhh.hhhh*

hhhh.hhhh.hhhh is the C-MAC that has a B-MAC association.

Show mac vpls pbb-ib x

The **show mac vpls pbb-ib** command displays the C-MACs associated with any B-MAC. Otherwise, the output refers to a specific instance. Here is a sample output:

```
device(XMR12)#show mac vpls pbb-ib 1
Total VPLS mac entries associated with instance 1 pbb-ib: 3
MAC Address     BMAC Address    Port  Vlan(In-Tag)/Peer ISID      Age
============    ============    ====  ================= ====      ===
0000.0404.0000  0000.B000.0201  1/2   300               30000     10
0000.0414.0000  0000.B000.0201  1/2   300               30000     12
0000.0418.0000  0000.B418.0302  1/1   300               30000     20
device(XMR12)#show mac vpls pbb-ib
Total VPLS mac entries: 5
VPLS ID MAC Address     BMAC Address    Port  Vlan(In-Tag)/Peer ISID      Age
======= ============    ============    ====  ================= ====      ===
    3     0000.0000.0200  0000.B020.A101  4/4   120               33000                          30
    4      0000.0003.0220  0000.B004.B001  4/1
121                                                                      2000
22
    1      0000.0404.0000  0000.B000.0201  1/2   300               30000                      10
    1      0000.0414.0000  0000.B000.0201  1/2
300                                                                      30000                      12
    1      0000.0418.0000  0000.B418.0302  1/1   300               30000     20
```

Syntax: show mac vpls pbb-ib *x*

x refers to a VPLS PBB instance. If no instance is specified, the MACs for all instances are displayed, sorted by CMAC.

Show nht

The **show nht** command will show those entries that are MAC based. For MAC based entry many of the fields displayed are not applicable and will have the N.A. value displayed.

```
device(XMR12)#show nht
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP          Index      MAC Address                    VLAN      Out I/F Out Port TNL CNT XC CNT LABEL
EXP/PCP
10.20.20.1          0               000c.dbf5.c773    1           1/20           1/20
0                       1
N.A.              1           000c.dbf4.0000    50   N.A.      N.A.
0                         3                                  0
N.A.              2           000c.dbf5.0000    50   N.A.      N.A.
0                         2                                  0
N.A.              3           000c.dbf6.0000    60   N.A.      N.A.
0                         1                                  5
```

Syntax: show nht

Show nht vlan vlan_id

The **show nht vlan** command allows the nht entry to be displayed based with those that have the matching VLAN ID.

```
device(XMR12)#show nht vlan 50
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP Index MAC Address              VLAN Out I/F  Out Port  TNL CNT XC CNT  LABEL   EXP/PCP
N.A.           1 000c.dbf4.0000 50   N.A.     N.A.      0       3               0
N.A.           2 000c.dbf5.0000 50   N.A.     N.A.      0       2               0
```

Syntax: show nht vlan *vlan_id*

Show nht mac-based [vlan vlan id]

The **show nht mac-based** command displays the MAC-based NHT entries as well as optionally filtered to a specified VLAN ID.

```
device(XMR12)#show nht mac-based
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP Index     MAC Address          VLAN Out I/F  Out Port  TNL CNT XC CNT  LABEL   EXP/PCP
N.A.        1      000c.dbf4.0000 50   N.A.     N.A.      0       3               0
N.A.        2      000c.dbf5.0000 50   N.A.     N.A.      0       2               0
N.A.        3      000c.dbf6.0000 60   N.A.     N.A.      0       1               5
device(XMR12)#show nht mac-based vlan 60
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP Index     MAC Address          VLAN Out I/F  Out Port  TNL CNT XC CNT  LABEL   EXP/PCP
N.A.        3      000c.dbf6.0000 60   N.A.     N.A.      0       1               5
```

Syntax: show nht mac-based [**vlan** *vlanid*]

Show cam ifl-isid slot/port output changes

The output of the **show cam ifl-isid** command displays the Service-Type programmed in the service PRAM.

```
device(XMR12)#show cam ifl-isid 1/2
Slot Index   Port  Outer VLAN Itag ISID  PRAM   IFL ID IPV4/V6  Service
     (Hex)                              (Hex)         Routing  Type
1    00c1ffb 1/2   300        2000       181ffb 4097   0/0      5
1    00c1ffc 1/1   300        2000       181ffc 4097   0/0      5
```

Syntax: show cam ifl-isid *slot/port*

Show service-type-table output changes

The output of **show service-type-table** command displays the Forced VLAN Action field programmed in the service type table entry.

```
device#show service-type-table port 1/1
ST   Service   Service  VLAN   RX QOS  TCI    TC    VLAN  IPv4   IPv6   Mcast Forced
entry   type                  ID    valid    ID   valid  Index   ID    Rtg    Rtg   bidir  VL ACT
0x00000  0 (LEGACY) 000000    1     00000   0      -    0000    N      N     N      N
0x00032  1 (VPLS  ) 000005    1     04095   0      -    0050    Y      Y     N      N
0x000c8  1 (VPLS  ) 000003    1     04095   0      -    0200    Y      Y     N      Y
```

Syntax: show service-type-table *slot/port*

802.1ag over PBB OAM

Connectivity Fault Management (802.1ag) is for end-to-end connectivity monitoring. The following functionality has been added.

- CFM monitoring for C-SVLAN/S-VLAN
- CFM monitoring for ISID and B-VLAN
- CFM monitoring for Link MA
- Port status TLV
- Remote Defect Indication

  NOTE
  PBB-OAM is not supported in the 24x10G card. Only Plain VLAN CFM is supported in the 24x10G card.

Configuration scenarios

Within a PBBN (see MAC Learning for PBB Packets on page 190), the encapsulation performed by Provider Instance Ports (PIPs) also encapsulates CFM frames sourced by customers attached to Customer Network Ports (CNPs). The encapsulation of S-VLAN and C-VLAN CFM frames hides them from the PBBN. All eight levels of CFM frames generated in customer networks are carried over the backbone as encapsulated data and may be used by customer networks.

FIGURE 46 Backbone Edge Bridge Operation

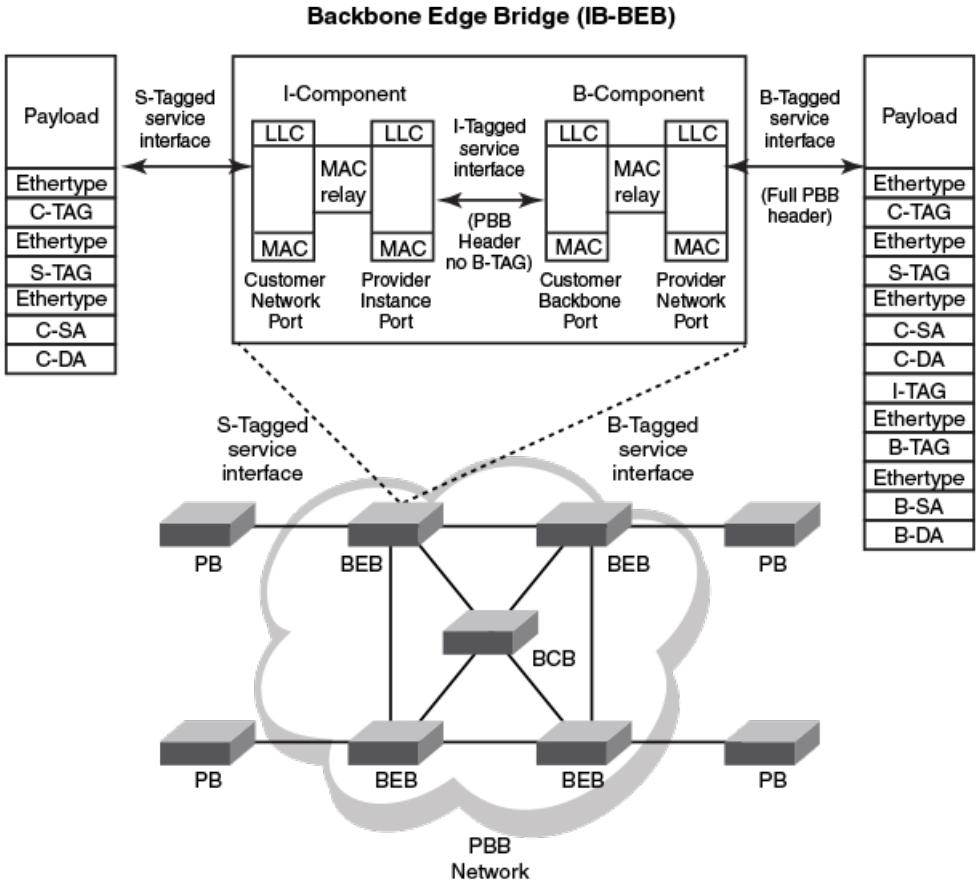
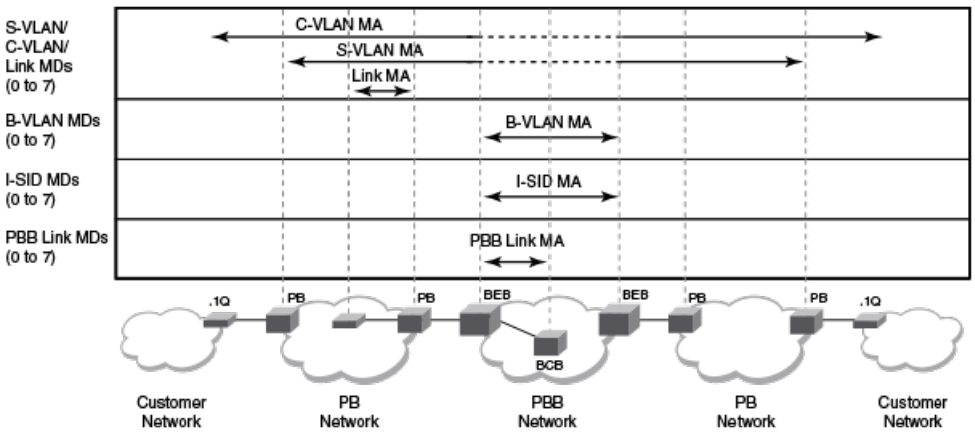


Figure 48 describes the four types of Maintenance Associations (MA) defined by IEEE 802.1ag CFM.

- A full set of eight MD levels exists within each PBBN for use by I-SIDs MAs.
- A full set of eight maintenance levels exists within each PBBN for use by B-VLAN CFM frames.
- An additional eight maintenance levels exists for the LAN link segments.

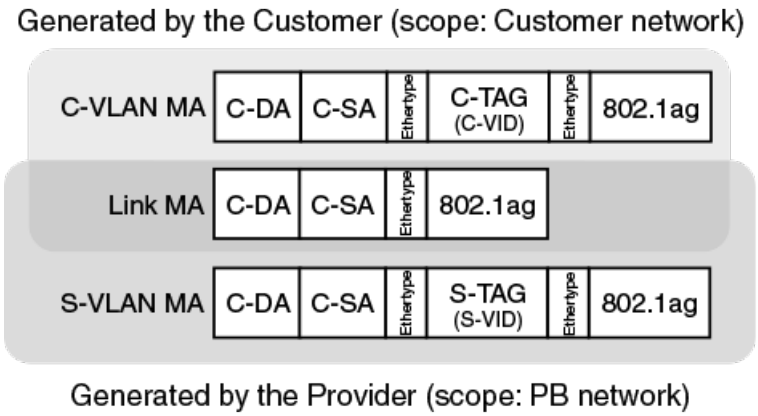
FIGURE 47 Maintenance Association (MA) Categories



S-VLAN, C-VLAN, and Link MAs (Figure 49)

- Customers and Provider (PB) share the same MD level space.
- CFM processing scope within Customer/PB network is determined by the MD level selection.
- The PB network normally only processes provider MD level OAM frames.
- Customer C-VLAN OAM frames use customer MD level and are normally not processed by the PB network.
- S-VLAN CFM frames are only visible where S-TAGs are processed.
- C-VLAN CFM frames are only visible where C-TAGs are processed.
- Once the S-VLAN CFM frames are encapsulated by a BEB, they appear just like any data frame within the PBBN. That is, they do not activate any CFM functions within the PBBN past the BEB.

FIGURE 48 S-VLAN, C-VLAN, and Link MA Frames



B-VLAN MAs (Figure 50)

- B-VLAN MAs manage the B-VLANs within a single PBBN.
- The scope of these 802.1ag frames is the PBB network only. These frames do not leave the PBB network.
- These CFM frames are only visible within the PBBN where the B-TAG is being processed.

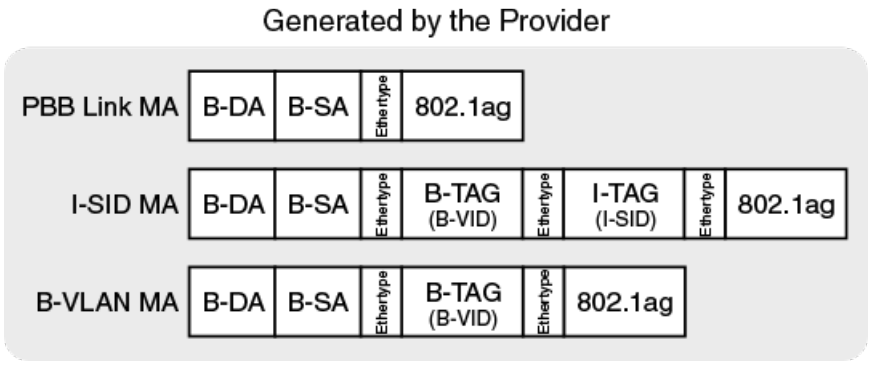
I-SID MAs (Figure 50)

- Backbone service instance CFM frames are only visible within the PBBN where the I-TAG is being processed.

PBB LAN Link Segment MAs (Figure 50)

- PBB LAN link MAs optionally manage links within a PBBN.

FIGURE 49 PBB MA Frames



Types of MEPs and MIPs

Associated with the MAs described in the previous section, there are MEPs and MIPs. Support is provided for the following types of MIPs and MEPs on the appropriate interfaces:

- MIPs and up/down MEPs for C-VIDs: Where C-Tags are processed
- MIPs and up/down MEPs for S-VIDs: Where S-Tags are processed
- Up/down MEPs for I-SIDs: Where I-Tags are processed
- MIPs and up/down MEPs for B-VIDs: Where B-Tags are processed
- Down MEPs for Link MAs

Hierarchical Fault Detection Operation

Ethernet OAM uses a hierarchical fault detection scheme.

- Customer scope: Uses C-VLAN MA and Link MA
- PB scope: Uses S-VLAN MA and Link MA
- PBB scope: Uses B-VLAN MA, I-SID MA, and Link MA

Faults can be detected using Continuity Check messages at any of the four MA categories. Once a fault is detected, Link Trace can be used to narrow down the location of the fault. Depending on the location of the fault, a different MA category may need to be used to further isolate the location of the fault.

802.1ag for Link MA

A new MA type is supported for Link MA.

```
device(config- dot1ag -DONAME) #ma-name MANAME link-MA priority 4
device(config- dot1ag -DONAME-MANAME) # mep 2 down port eth 1/1
```

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | [**vll** [*vll-name*] | [*vll-id*] | **vll-local** [*vll-name*] | **link-MA** } **priority** *priority*

Only the **down MEP** configuration is allowed for a link-MA.

Syntax: mep *mep-id* [**up** | **down**] [**vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

Error messages

If an **up MEP** command is issued the following error message is displayed.

```
device(config- dot1ag -DONAME-MANAME) # mep 2 up port eth 1/1
Error : UP MEP cannot be configured on LINK-MA
```

MEP configuration for link MA is rejected if more than one port is entered in the configuration.

```
device(config- dot1ag -DONAME-MANAME) # mep 2 down port eth 1/1 to 1/3
Error : MEP cannot be configured on multiple ports for Link MA
```

MIP operation is not allowed for link MA. An error is displayed if the **MIP-POLICY** command is issued.

```
device(config- dot1ag -DONAME-MANAME)#mip-policy explicit
Error: MIP cannot be configured on LINK-MA
```

Link MA capabilities and limitations

- MIP functionality is not supported for link-MA.
- Link trace functionality is not supported for link-MA.
- Loopback and delay measurement functionality is supported on link-MA.
- 802.1ag sub-second-timer functionality is supported for link-MA.
- 802.1ag functionality is supported during hitless upgrade and switchover.
- 802.1ag functionality is supported for LAG.
- When MEP is configured on a VPLS instance with sub-second CCM timer, the MIP flooding should be turned off. When there is no MEP and it is a sub-second-timer, then dot1ag flooding should be turned on.

```
device#show cfm
Domain: D10
Index: 3
Level: 7
  Maintenance association: MA
  Ma Index: 1
  CCM interval: 10000 ms
  LINK MA ID: 0
  Priority: 3
  MEP                  Direction                    MAC                              PORT          PORT-
STATUS-TLV
  ====                 ========                  =========                         =====
===============
    10      DOWN                  0012.f2f7.3900             ethe 1/1        N
device#show cfm connectivity
Domain: D10 Index: 3
Level: 7
  Maintenance association: MA
  MA Index: 1
  CCM interval: 10000 ms
  LINK MA ID: 0
  Priority: 3
RMEP            MAC                          VLAN/PEER        AGE
PORT                    SLOTS                     STATE
====      ==========                            ===========     =====
=====                 =====                     =======
1         0012.f2f7.3861                             0                   54020
1/1               1           OK
```

802.1ag for CVLAN and SVLAN

Extreme supports MEP and MIP for the regular VLAN and MEP for VPLS VLAN.

You will need to create a new MA for PBB-VPLS. SVLAN functionality is supported in a similar manner as it is with CVLAN functionality. The only difference is the tag-type configuration for the VPLS end-points.

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | **pbb-vpls** *vpls-id* | [**vll** [*vll-name*] | [vll-id]|**vll-local**[*vll-name*]]}**priority***priority*

Example

```
device(config- dot1ag -DONAME)# ma-name MANAME pbb-vpls 10 priority 4
```

Both down MEP and UP MEP configuration are accepted for CVLAN and SVLAN.

Syntax: mep *mep-id* [**up** | **down**] [**vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

- UP MEP for CVLAN and SVLAN will transmit CCM packets on C-tagged, S-tagged and IB-tagged end-points on the service instance.
- MIP will take care of CVLAN/SVLAN translation and MIP is created on C-tagged and S-tagged end-points.
- 802.1ag sub-second-timer functionality is supported for PBB CVLAN/SVLAN.
- 802.1ag functionality for PBB CVLAN/SVLAN is supported during switch-over.
- 802.1ag functionality is not supported during hitless upgrade for PBB CVLAN/SVLAN.
- 802.1ag functionality is supported if the PBB CVLAN/SVLAN end-point is a LAG.
- Link-trace, loopback and delay-measurement for PBB CVLAN/SVLAN is supported.
- 802.1ag for CVLAN/SVLAN is supported on untagged end-points, port based untagged end-points and C-tagged end-points.
- 802.1ag for CVLAN/SVLAN is not supported on dual-tagged end-points.
- SVLAN keep-mode is supported.
- When MEP is configured on a VPLS instance with sub-second CCM timer, the MIP flooding should be turned off. When there is no MEP and it is a sub-second-timer, then dot1ag flooding should be turned on.

```
device# show cfm
Domain: D1
Index: 1
Level: 3
  Maintenance association: MA
  Ma Index: 1
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  MEP   Direction                    MAC                          PORT       PORT-STATUS-TLV
  ====  ========    =========                          =====       ==============
    1          UP          0012.f2f7.3900           ethe 1/1                 N
  MIP      VLAN/VC                          Port                         Level                    MAC
  ====    ========                              =====                        =====
==========
               200                   1/1               3               0012.f2f7.3901
device# show cfm connectivity
Domain: D1 Index: 1
Level: 3
  Maintenance association: MA
  MA Index: 1
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  RMEP              MAC                    VLAN/ISID                    AGE
PORT               SLOTS           STATE
  ====               =====                     =======                      ======
```

```
=======                 ====                ==========
    10          0012.f2f7.3860          200                              73460
1/2               1           OK
```

802.1ag for BVLAN

BVLAN functionality is supported by the use of regular VLANs only.

802.1ag for ISID

PBB-VPLS functionality supports IB-tagged end-points. 802.1ag for ISID is supported in a similar manner as it is supported for CVLAN and SVLAN end-points for local-VPLS. It requires the creation of a new MA for PBB-VPLS.

```
device(config- dot1ag -DONAME)# ma-name MANAME pbb-vpls 10 priority 4
```

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | **pbb-vpls** *vpls-id* | [**vll** [*vll-name*] | [*vll-id*] | **vll-local** [*vll-name*] } **priority** *priority* **mep** *mep-id* [**up** | **down**] [**vlan** *vlan-id* **ISID** *isid* **port** *port-id* | **vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

- Both down MEP and UP MEP configuration is accepted for ISID.
- UP MEP for ISID will transmit CCM packets out ONLY on the BVLAN ports carrying the same ISID.
- MIP functionality is not supported for ISID.
- 802.1ag sub-second-timer functionality for ISID is supported.
- 802.1ag functionality for ISID during switchover is supported.
- 802.1ag functionality for ISID during hitless upgrade is not supported.
- 802.1ag functionality for ISID is supported where IB-tagged end-points are LAG.
- Link-trace, loopback and delay-measurement for ISID is supported.
- SVLAN keep-mode is supported.
- When MEP is configured on a VPLS instance with sub-second CCM timer, the MIP flooding should be turned off. When there is no MEP and it is a sub-second-timer, then dot1ag flooding should be turned on.

```
device#show cfm
Domain: D2
Index: 2
Level: 1
  Maintenance association: MA
  Ma Index: 1
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  MEP           Direction                       MAC                   PORT                     PORT-
STATUS-TLV
  ====      ========                        =========                       ======
==============
    10          DOWN              0012.f2f7.3901               ethe
1/2                                   N

  MIP     VLAN/VC                    Port                    Level                    MAC
  ====    ========               =====                 =====                    ==========
               100               1/1               1              0012.f2f7.3901
               200               1/2               1              0012.f2f7.3901
```

In the following example, MIP is created for BVLAN not for ISID.

```
device# show cfm connectivity
Domain: D2 Index: 2
Level: 1
  Maintenance association: MA
  MA Index: 1
```

```
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  RMEP           MAC                          VLAN/ISID                    AGE
PORT               SLOTS                STATE
  ====  ==========                          =========                      =====
=====                ======                 ======
     1      0012.f2f7.3861                        1000                        72320
1/2                1               OK
```

802.1ag Port Status TLV

The Port Status TLV indicates the ability of the Bridge Port on which the transmitting MEP resides to pass ordinary data, regardless of the status of the MAC.

> **NOTE**
> Port Status TLV is not be supported for LINK MA.

MEP configuration provides support for port status TLV.

Syntax: mep *mep-id* [**up** | **down**] [**tlv-type** *tlv-type-value*] [**vlan** *vlan-id* **ISID** *isid* **port** *port-id* | **vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

The *tlv-type-value* should be set to the *port-status-tlv*.

Sample configuration output

Syntax: show cfm domain *domainName* **ma** *maName* **mep-id** *mepId*

```
device# show cfm domain D2 ma 1 mep-id 5000
Domain: customer
Index: 1
Level: 7
  Maintenance association: admin
  Ma Index: 2
  CCM interval: 10000 ms
  ESI aaa VLAN ID: 100
  Priority: 7
  MEP   Direction           MAC                                        PORT         PORT-STATUS-TLV
  ====  =========  =========                                       =======        ================
    1      DOWN                0024.3863.7741            ethe 1/1            Y
Y - Means port status tlv is enabled for the MEP
N - Means port status tlv is not enabled for the MEP
device #show cfm connectivity do bvlan ma bvlan rmep-id 5000
Domain: bvlan Level: 5
Maintenance association: bvlan VLAN VLAN/VPLS/VLL ID: 250 Priority: 5
CCM interval: 1000 ms
RMEP           MAC                  PORT      Oper      Age          CCM              RDI
Port      Intf      Intvl Seq          core
                           State     Val           Cnt      Status Status      Error Error Fault level
====  ============== ==== =====  ==== ====== =======  ==== ====== ====== === ==
5000  0024.3898.da20           3/1      OK               2156 216030      N
2      0     N     N           N
Value              Port Status

1              Port Blocked ( Not Forwarding)

2              Port Forwarding
```

802.1ag RDI

The Remote Defect Indication (RDI), a single bit, is carried in CCM packets. The absence of RDI in a CCM indicates that the transmitting MEP is receiving CCMs from all RMEPs. The presence of RDI indicates that transmitting MEP is not receiving CCM from one or more RMEPs.

```
device#show cfm connectivity domain customer ma admin rmep-id 1
Output:
Domain: customer Level: 7
Maintenance association: admin VLAN VLAN/VPLS/VLL ID: 100 Priority: 7
CCM interval: 100ms
                              PORT                Oper          Age
RMEP          MAC                       State          Val                 Cnt                 CCM
RDI
====           =========                    ======                 =====       =====           ======= ==========
  1           0024.3863.7745      1/1                                OK            8180
13                 Y
Port                 Intf                Intvl           Seq
Status                 Status                Error           Error
======                 ======                =====           =====
0                0               N           N

Sample Output:
CES-2#show cfm connectivity domain customer ma admin rmep-id 1
Output:
Domain: customer Level: 7
Maintenance association: admin VLAN VLAN/VPLS/VLL ID: 100 Priority: 7
CCM interval: 100ms
RMEP   MAC               PORT    Oper     Age       CCM           RDI
                                                    State    Val
Cnt
====  =========   ====== =====   =====  ======= ==========
  1  0024.3863.7745    1/1    OK          8180         13              Y
Port        Intf          Intvl     Seq
Status    Status     Error    Error
====== ======   =====  =====
   0               0          N         N
```

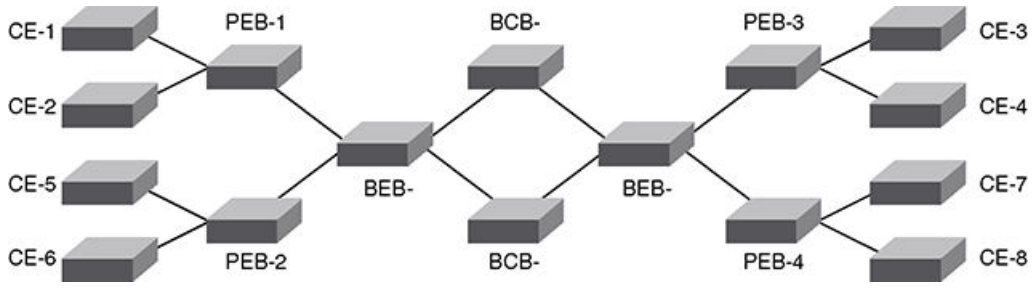
Deployment Scenarios and CLI Configuration

NOTE

RDI is not supported for LINK MA.

Deployment Scenario 1 (Down MEPs on CEs and MIP on PE)

FIGURE 50 Deployment scenario 1



Configuration for CE Devices

Configuration for CE devices is the same as for 802.1ag over VLAN.

Configuring CE-1

VLAN configuration

```
device(config)#vlan 10
device(config-vlan-10)#tagged ethernet 1/1
```

CFM configuration:

1. To enable CFM, enter the following command:

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

   ```
   device(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain of vlan-id 10 with a priority 3.

   ```
   device(config-cfm-md-CUST_1)#ma-name ma_5 vlan-id 10 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages. The default is 10-seconds.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

5. Configure a MEP on port 1/1 and vlan 10.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#mep 1 down vlan 10 port ethe 1/1
   ```

6. Configure a remote-mep.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#remote-mep 2 to 2
   ```

7. Continue the configuration for other CE devices as needed.

Configuration for PE Devices:

Configuring PEB-1

Tag-type configuration

Assume CVLAN tag-type is 0x8100 and S-VLAN tag-type is 0x900.

```
device(config)#tag-type 8100 eth 1/1
device(config)#tag-type 8100 eth 1/2
device(config)#tag-type 9100 eth 1/3
```

VPLS local configuration:

Assume CE-1 is connected to PEB-1 port 1/1 on VLAN 10 and CE-2 is connected to PEB-2 port 1/2 on VLAN 20 and PEB-1 is connected to BEB-1 on port 1/3 VLAN 30.

```
device(config)#router mpls
device(config-mpls)#vpls PB-VPLS 20
device(config-mpls-vpls-PB-VPLS)#pbb
device(config-mpls-vpls-PB-VPLS-pbb)#exit
device(config-mpls-vpls-PB-VPLS)#
device(config-mpls-vpls-PB-VPLS)#vlan 10
device(config-mpls-vpls-PB-VPLS-vlan-10)#tagged eth 1/1
```

```
device(config-mpls-vpls-PB-VPLS-vlan-10)#
device(config-mpls-vpls-PB-VPLS-vlan-10)#vlan 20
device(config-mpls-vpls-PB-VPLS-vlan-20)#tagged eth 1/2
device(config-mpls-vpls-PB-VPLS-vlan-20)#
device(config-mpls-vpls-PB-VPLS-vlan-20)#vlan 30
device(config-mpls-vpls-PB-VPLS-vlan-30)#tagged eth 1/3
```

CFM configuration:

If the VPLS local configuration is not done prior to configuring maintenance association. The MA configuration is not allowed. PEB-1 will work as a MIP.

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | **pbb-vpls** *vpls-id* | [**vll** [*vll-name*] | [*vll-id*] | **vll-local** [*vll-name*] } **priority** *priority*

1. Enable CFM using the **cfm-enable** command.

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

   ```
   device(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain for vpls-id 20 with a priority 3.

   ```
   device(config-cfm-md-CUST_1)#ma-name ma_5 pbb-vpls 20 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10 seconds.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

   In the above configuration, MIP gets created by default on the VPLS end-points. You can also configure **explicit-mip** on PEB-1. In that case, MIP will be created on the VPLS end-points if a MEP is created on the port at some lower MD Level.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)# mip-creation explicit
   ```

   To change back to default use the following command.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)# mip-creation default
   ```

Verifying Connectivity Using 802.1ag

Once you configure CE-1, CE-2, PEB-1, and PEB-2 you can determine the end-to-end connectivity by looking at the remote-mep status using the following show commands:

Syntax: show cfm connectivity [**domain NAME**] [**ma MA NAME**]

```
device#show cfm connectivity
Domain: CUST_1 Level: 7
Maintenance association: ma_5
CCM interval: 10000 ms
VLAN ID: 30
Priority: 3
RMEP              MAC                VLAN/PEER           AGE              PORT           SLOTS
====          =============          =======                    ====   ====           ====
2             000c.dbe2.8a00                    30             879          1/2              1
device#show cfm connectivity domain CUST_1 ma ma_5 rmep-id 2
Domain: CUST_1 Level: 7
Maintenance association: ma_5 VLAN ID: 30 Priority: 3
CCM interval: 10
RMEP            MAC                PORT           Oper          Age      CCM            RDI
Port               Intf            Intvl Seq
                                        State           Val          Cnt      Status Status
Error           Error
```

```
==== =============== === === ===== ===  ===            ====            ======      ======           ====
2     000c.dbe2.8a00                1/1 OK  26000 2600 N
0                                    0      N                       N
Verifying Connectivity Using 802.1ag Loopback/Linktrace
```

Use the **cfm linktrace domain NAME ma MA-NAME src- mep** and the **cfm loopback domain NAME ma MA-NAME scr-mep** commands to manually monitor the status of peer, as shown below:

Syntax: cfm linktrace domain NAME ma MA-NAME src- mep *mep-id* **target-mip HH:HH:HH:HH:HH:HH** | **target-mep** *mep-id* } [**timeout** *timeout*] [**ttl** *TTL*]

```
device#cfm linktrace domain CUST_1 ma ma_5 src-mep 1 target-mep 2
Linktrace to 000c.dbe2.8a00 on Domain CUST_1, level 4: timeout 10ms, 8 hops
------------------------------------------------------------
Hops         MAC          Ingress  Ingress Action  Relay Action
Forwarded  Egress   Egress Action    Nexthop
------------------------------------------------------------
Type Control-c to abort
1   000c.dbe2.8a00 10.1.1.1   IgrOK          RLY_HIT
Not Forwarded
Destination 000c.dbe2.8a00 reached
```

Syntax: cfm loopback domain NAME ma MA-NAME scr-mep *mep-id* { **target-mip HH:HH:HH:HH:HH:HH** | **targetmep** *mep-id* } [**number** *number*] [**timeout** *timeout*]

```
device#cfm loopback domain CUST_1 ma ma_5 src-mep 1 target-mep 2
DOT1AG: Sending 10 Loopback to 000c.dbe2.8a00, timeout 10000 msec
Type Control-c to abort
Reply from 000c.dbe2.8a00: time=3ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time=38ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
A total of 10 loopback replies received.
Success rate is 100 percent (10/10), round-trip min/avg/max=0/4/38 ms.
```

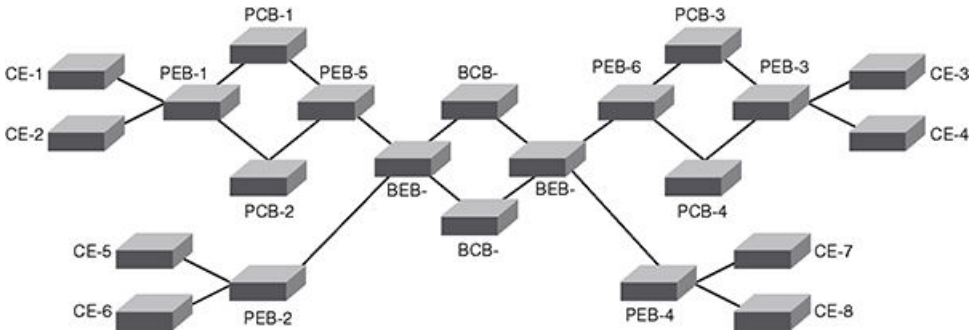
If the linktrace and loopback to target-mep 2 fails, then linktrace can be done on the MIPs on PEB-1 and PEB-2 to know the exact failure.

Deployment Scenario-2 (UP MEPs and MIPs on PEs)

If you have a deployment scenario where PEB-1 is not directly connected to BEB-1, but it is connected to a PB network, then you can use MIPs configured over the intermediary nodes, assuming the PB network is managed by the same administrator.

If the network is managed by a separate administrator, then you can have UP-MEPs configured on the PE ports connected to CE devices. The intermediate devices will have MIPs configured.

FIGURE 51 Deployment scenario-2and3



Configuration for PE Devices

Configuring PEB-1

The local VPLS configuration will be the same as shown in the previous deployment scenario. If the local VPLS configuration is not done prior to configuring maintenance association, the MA configuration is not allowed. Also, the port and vlan in the MEP configuration should exist in local VPLS configuration prior to MEP configuration. Otherwise, it is not allowed. The port in the MEP configuration can be either a tagged or untagged port already present in the local-VPLS configuration.

CFM configuration steps for PEB-1

1. Enter the **cfm-enable** command to enable CFM.

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

   ```
   device(config-cfm)#domain-name PROVIDER_1 level 4
   ```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

   ```
   device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10-seconds.

   ```
   device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
   ```

5. Configure a MEP on port 1/1 and vlan 10.

   ```
   device(config-cfm-md-PROVIDER_1-ma-ma_8)#mep 6 up vlan 10 port ethe 1/1
   ```

   A similar configuration will need to be done on PEB-3. MIP should be configured on PEB-5, PEB-6, PCB-1, PCB-2, PCB-3, and PCB-4.

CFM configuration steps for PEB-5.

1. Enter the **cfm-enable** command to enable CFM.

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

```
device(config-cfm)#domain-name PROVIDER_1 level 4
```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

```
device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10-seconds.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
```

To monitor the connectivity between PEB-1 and PEB-3, you can use the **show cfm connectivity** command as mentioned in the previous scenario. Also, you can use either loopback or linktrace on PEB-1 or PEB-3.

Deployment Scenario-3 (MIPs on BEBs)

In Deployment Scenario-2 (UP MEPs and MIPs on PEs) on page 210 BEBs can work as MIP for the UP MEPs configured on PEs in the previous deployment scenario.

Configuring BEB-1

Tag-type configuration

Assume S-VLAN tag-type is 0x9100 and B-VLAN tag-type is 0x88e8

```
device(config)#tag-type 9100 eth 1/1
device(config)#tag-type 9100 eth 1/2
device(config)#tag-type 88e8 eth 1/3
```

SVLAN Configuration

```
device(config)#vlan 10
device(config-vlan-10)#tagged ethernet 1/1
```

SVLAN Configuration

```
device(config)#vlan 20
device(config-vlan-20)#tagged ethernet 1/2
```

BVLAN Configuration

```
device(config)#vlan 100
device(config-vlan-100)#tagged ethernet 1/3
```

VPLS local configuration

Assume port 1/1 and 1/2 on BEB are connected to PE devices (S-tagged end-point) and 1/3 is an IB-tagged end-point.

```
device(config)#router mpls
device(config-mpls)#vpls PBB-VPLS 20
device(config-mpls-vpls-PBB-VPLS)#pbb
device(config-mpls-vpls-PBB-VPLS-pbb)#exit
device(config-mpls-vpls-PBB-VPLS)#
device(config-mpls-vpls-PBB-VPLS)#vlan 10
device(config-mpls-vpls-PBB-VPLS-vlan-10)#tagged eth 1/1
device(config-mpls-vpls-PBB-VPLS-vlan-10)#
device(config-mpls-vpls-PBB-VPLS-vlan-10)#vlan 20
device(config-mpls-vpls-PBB-VPLS-vlan-20)#tagged eth 1/2
device(config-mpls-vpls-PBB-VPLS-vlan-20)#
device(config-mpls-vpls-PBB-VPLS-vlan-20)#vlan 100 isid 200000
```

```
device(config-mpls-vpls-PBB-VPLS-vlan-100-isid-200000)#tagged eth 1/3
device(config-mpls-vpls-PBB-VPLS-vlan-100-isid-200000)#
```

CFM configuration

If the VPLS local configuration is not done prior to configuring the maintenance association, the MA configuration is not allowed. BEB-1 will work as an MIP.

1. Enter the **cfm-enable** command to enable CFM.

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

   ```
   device(config-cfm)#domain-name PROVIDER_1 level 4
   ```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

   ```
   device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
   ```

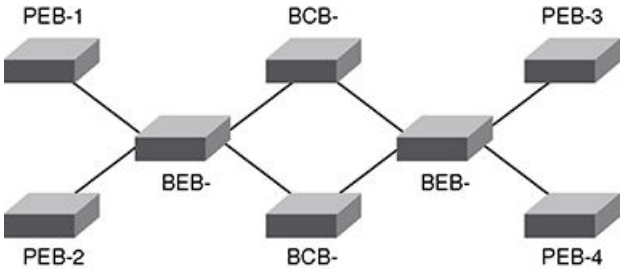
4. Set the time interval between successive Continuity Check Messages. The default is 10-seconds.

   ```
   device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
   ```

   The above configuration will create SVID MIPs on BEBs. You can use linktrace to BEB MIPs from PE MEPs to further isolate the fault location.

Deployment Scenario-4 (ISID MEPs on BEBs)

FIGURE 52 Deployment scenario-4and5



The Local VPLS configuration is similar to the previous scenario.

CFM configuration steps for BEB-1.

1. Enter the **cfm-enable** command to enable CFM

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name PBB_1 and level 4.

   ```
   device(config-cfm)#domain-name PBB_1 level 3
   ```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

   ```
   device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10-seconds.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
```

5. Configure a MEP on port 1/3 and vlan 10.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#mep 6 up ISID 200000 vlan 10 port ethe 1/3
```

To monitor the connectivity between BEB-1 and BEB-2, you can use the **show cfm connectivity** commands as mentioned in the previous scenario. Also, you can use either loopback or linktrace on BEB-1 or BEB-2.

Deployment Scenario-5 (BVLAN MEPs on BEBs and MIP on BCBs)

BVLAN CFM configuration will be similar to regular VLAN and it will support both MEP and MIP functionality.

Show Commands

The **show cfm** command provides the following output.

```
device#show cfm
Domain: md2
Level: 6
  Maintenance association: ma2
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 4
  MEP   Direction       MAC                                          PORT
  ====  =========           =========                                ====
    2    UP                     000c.dbf3.fa02           ethe 1/3
```

The **show cfmconnectivity** command provides the following output.

```
device#show cfm connectivity
Domain: md2 Level: 6
  Maintenance association: ma2
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 4
  RMEP               MAC                     VLAN/PEER                         AGE
PORT              SLOTS
  ====               =====                     ============          =======
========                  ====
    3      000c.dbf3.fb02                            10.2.2.2                    320              1
```

The **show cfm connectivity domain** command provides the following output.

```
device#show cfm co domain md2 ma ma2 rmep 3
Domain: md2 Level: 6
Maintenance association: ma2 PBB-VPLS ID: 100 Priority: 4
CCM interval: 10000 ms
  RMEP   MAC            PORT  Oper     Age      CCM      RDI Port   Intf   Intvl Seq
                             State    Val      Cnt          Status Status Error Error
  ====  =============== ===== =====  ======= ==========  ==== ====== ====== ===== =====
     3  000c.dbf3.fb02  00c35    OK       20          39   N     0      0     N     Y
```

The **show cfm domain** command with the *domain-name***ma***ma-name* parameters, provides the following output.

```
device#show cfm do md2 ma ma2
Domain: md2
Level: 6
  Maintenance association: ma2
  CCM interval: 10000 ms
  VLL ID: 100
  Priority: 4
  MEP   Direction  MAC            PORT/TX PORT
```

```
  ====  =========  =========      ====
     2    UP      000c.dbf3.fa02  ethe 1/3
  REMOTE MEP id 3 MAC 000c.dbf3.fb02    OK 2
CFM port (VL100) PBB-VPLS  100
CFM port (VL800000) PBB-VPLS 100
device#
```

Syntax: show cfm domain *domain-name* **ma** *ma-name*

Spanning Tree Protocol

Spanning Tree Protocol overview

The CES 2000 Series and CER 2000 Series devices support the Ethernet Service Instance (ESI) framework. A user can configure ESIs in the process of configuring Provider Bridging and Provider Backbone Bridging. By default, a device has a "default ESI" configured in which VLANs 1- 4090 exist. This chapter refers to configuration and use of Spanning Tree Protocols under the default ESI.

IEEE 802.1D Spanning Tree Protocol (STP)

The Extreme device supports Spanning Tree Protocol (STP) as described in the IEEE 802.10-1998 specification. STP eliminates Layer 2 loops in networks, by selectively blocking some ports and allowing other ports to forward traffic, based on configurable bridge and port parameters. STP also ensures that the least cost path is taken when multiple paths exist between ports or VLANs. If the selected path fails, STP searches for and then establishes an alternate path to prevent or limit retransmission of data.

Enabling or disabling STP

STP is disabled by default on the Extreme device. Thus, new VLANs you configure on the Extreme device have STP disabled by default. Table 32 lists the default STP states for the Extreme device.

TABLE 32 Default STP states

Device type	Default STP type	Default STP state	Default STP state of new VLANs
Extreme	Extreme's multiple instances of spanning tree	Disabled	Disabled

By default, each VLAN on the Extreme device runs a separate spanning tree instance. Each Extreme device has one VLAN (VLAN 1) by default that contains all of its ports. However, if you configure additional port-based VLANs on the Extreme device, then each of those VLANs on which STP is enabled and VLAN 1 all run separate spanning trees.

You can enable or disable STP on the following levels:

- **Globally** - Affects all VLANs on the Extreme device.
- **Individual VLAN** - Affects all ports within the specified VLAN. When you enable or disable STP within a VLAN, the setting overrides the global setting. Thus, you can enable STP for the ports within a VLAN even when STP is globally disabled, or disable the ports within a port-based VLAN when STP is globally enabled.
- **Individual port** - Affects only the individual port. However, if you change the STP state of the primary port in a LAG group, the change affects all ports in the LAG group.

Enabling or disabling STP globally

Use the following methods to enable or disable STP on the Extreme device on which you have not configured VLANs.

> NOTE
> When you configure a VLAN, the VLAN inherits the global STP settings. However, once you begin to define a VLAN, you can no longer configure standard STP parameters globally using the CLI. From that point on, you can configure STP only within individual VLANs.

> NOTE
> Reloading the Extreme device with global STP enabled can display error on boot-up (error - no more stp instances available) if the number of vlans in the configuration are more than configured **system-max** for STP instances. The error message has no effect on the functionality.

When configuring spanning- tree at the global CLI level, the following message will prompt you to enter "y" for yes or "n" for no to change the spanning-tree behavior at the global level:

```
device(config)#spanning-tree
This will change the spanning-tree behavior at the global level.
Are you sure? (enter 'y' or 'n'): y
```

Enter 'y' to change the spanning-tree behavior. Enter 'n' to make no change to the spanning-tree configuration at the global level.

This command enables a separate spanning tree in each VLAN, including the default VLAN.

Syntax: [**no**] **spanning-tree**

Enabling or disabling STP on a VLAN

Use the following procedure to disable or enable STP on the Extreme device on which you have configured a VLAN. Changing the STP state in a VLAN affects only that VLAN.

To enable STP for all ports in a port-based VLAN, enter commands such as the following.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree
that there is no effect on the functionality due to this error message
```

Syntax: [no] spanning-tree

Enabling or disabling STP on a port

Use the following procedure to disable or enable STP on an individual port.

> NOTE
> If you change the STP state of the primary port in a LAG group, the change affects all ports in the LAG group.

To enable STP on an individual port, enter commands such as the following.

```
device(config)# interface 1/1
device(config-if-e1000-1/1)# spanning-tree
```

Syntax: [no] spanning-tree

STP in a LAG

The STP standard indicates that by default the path cost is determined by link speed. For a1 G port the path cost is 4 and for 10G port the path cost is 2. However, if a LAG is made consisting of *n* 1G ports, where *n* is less than 10, the path cost remains as 4. The standard does not indicate pathcost explicitly for LAG interfaces or for bandwidths between standard port bandwidth values, (for example, between 1G and 10G). Therefore, during STP deployment you may find that though a LAG has greater bandwidth, its in blocking/discarding state as its pathcost is the same as any 1G link and the portIndex of 1G port is lower, making the LAG go into a blocking/discarding state. This behavior is not restricted to 1G or 10G link speed but span across different link speeds. The same behavior also holds TRUE for RSTP deployments.

Default STP bridge and port parameters

Default STP bridge and port parameters lists the default STP bridge parameters. The bridge parameters affect the entire spanning tree. If you are using MSTP, the parameters affect the VLAN. If you are using SSTP, the parameters affect all VLANs that are members of the single spanning tree.

NOTE
STP information is specific to a VLAN, and the NetIron OS software uses the CONTROL VLAN to get the STP information for the all MIBs under dot1dStp and dot1DStpPortTable. Due to the limitation of the MIBs, information of per STP implementation of every specific VLAN is not displayed.

1. Forward Delay
2. The period of time a bridge will wait (the listen and learn period) before beginning to forward data packets.
3. seconds Possible values: 4 - 30 seconds
4. Maximum Age
5. The interval a bridge will wait for a hello packet from the root bridge before initiating a topology change.
6. seconds Possible values: 6 - 40 seconds
7. Hello Time
8. The interval of time between each configuration BPDU sent by the root bridge.
9. seconds Possible values: 1 - 10 seconds
10. Priority
11. A parameter used to identify the root bridge in a spanning tree (instance of STP). The bridge with the lowest value has the highest priority and is the root. A higher numerical value means a lower priority; thus, the highest priority is 0.
12. 768 Possible values: 0 - 65535

NOTE
If you plan to change STP bridge timers, it is recommended that you stay within the following ranges, from section 8.10.2 of the IEEE specification: - 2 * (forward_delay -1) >= max_age- max_age >= 2 * (hello_time +1)

Table 33 lists the default STP port parameters. The port parameters affect individual ports and are separately configurable on each port.

TABLE 33 Default STP port parameters

Parameter	Description	Default and valid values
Priority	The preference that STP gives this port relative to other ports for forwarding traffic out of the spanning tree.	128

TABLE 33 Default STP port parameters (continued)

Parameter	Description	Default and valid values
	A higher numerical value means a lower priority; thus, the highest priority is 8.	Possible values: 8 - 252, configurable in increments of 4
Path Cost	The cost of using the port to reach the root bridge. When selecting among multiple links to the root bridge, STP chooses the link with the lowest path cost and blocks the other paths. Each port type has its own default STP path cost.	10 Mbps - 100 100 Mbps - 19 1 Gigabit - 4 10 Gigabit - 2 40 Gigabit - 1 100 Gigabit - 1 Possible values are 1- 65535

Changing STP bridge parameters

To change the Extreme device's STP bridge priority to the highest value, so as to make the Extreme device the root bridge, enter the following command.

```
device(config)# vlan 20
device(config-vlan-20)# spanning-tree priority 0
```

To make this change in the default VLAN, enter the following commands.

```
device(config)# vlan 1
device(config-vlan-1)# spanning-tree priority 0
```

Syntax: [no] spanning-tree [**forward-delay value**] | [**hello-time value**] | [**max-age value**] | [**priority value**]

You can specify some or all of the parameters on the same command line. For information on parameters, possible values and defaults, refer to Changing STP bridge parameters.

NOTE

The **hello-time***value* parameter applies only when the device or VLAN is the root bridge for its spanning tree.

Changing STP port parameters

To change the path and priority costs for a port, enter commands such as the following.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree ethernet 1/5 path-cost 15 priority 64
```

Syntax: [no] spanning-tree ethernet slot/portnum path-cost value | priority value | disable | enable

The **ethernet** *slot/portnum* parameter specifies the interface.

For descriptions of path cost and priority, their default and possible values, refer to Default STP bridge and port parameters on page 219. If you enter a priority value that is not divisible by four, the software rounds it to the nearest value.

The **disable and enable** parameter disables or re-enables STP on the port. The STP state change affects only this VLAN. The port's STP state in other VLANs is not changed.

Root Guard

A new security feature has been added that allows a port to run STP but does not allow the connected device to become the Root. The Root Guard feature provides a way to enforce the root bridge placement in the network and allows STP 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.

> NOTE
> The feature is also available for MSTP and RSTP.

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 BLOCKING 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 will automatically set the port back to a FORWARDING state after the timeout period has expired.

> NOTE
> Root Guard may prevent network connectivity if improperly configured. It needs to be configured on the perimeter of the network rather than the core. Also, Root Guard should be configured only on the primary port of a LAG. If a port configured with Root Guard is made a secondary port, the LAG deployment will be vetoed.

Enabling Root Guard

Root Guard is configured on a per interfaces basis. To enable Root Guard, enter a command such as the following.

```
device(config)# interface ethernet 5/5
device(config-if-e10000-5/5) spanning-tree root-protect
```

Syntax: [no] spanning-tree root-protect

Enter the **no** form of the command to disable Root Guard on the port.

Setting the Root Guard timeout period

To configure the Root Guard timeout period globally, enter a command such as the following.

```
device(config)# spanning-tree root-protect timeout 120
```

Syntax: [no] spanning-tree root-protect timeout timeout in seconds

The *timeout in seconds* parameter allows you to set the timeout period. The timeout period may be configured to anything between 5 and 600 seconds. Default is 30 seconds.

Checking if Root Guard is configured

To determine if Root Guard is configured, enter the following command.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is enabled
, STP BPDU Guard is disabled
```

Syntax: show interface ethernet slot/port

Displaying the Root Guard state

To display the Root Guard state, enter the **show spanning-tree root-protect** command.

```
device#show spanning-tree root-protect
Port VLAN Current State
13/6 3 Consistent state
13/9 2 Inconsistent state (29 seconds left on timer)
```

Syntax: show spanning-tree root-protect

Reconfiguring the timeout period

The timeout period timer is activated whenever a port encounters a superior BPDU, which then results in a Root Guard violation. If the timeout period is reconfigured while a timer is in use, the timer on that port is set to the new timeout period, minus the time elapsed since the superior BPDU was received.

For example, the original timeout period on a device was configured for 60 seconds. The port encounters a superior BPDU and the timer starts. Issuing a **show span root-protect** command displays the following information.

```
device(config)#show span root-protect
Port    VLAN  Current State
1/4     1     Inconsistent state (56 seconds left on timer)
```

While the timer is in use, the timeout period is changed to 30 seconds through the issue of the following command.

```
device(config)# spanning-tree root-protect timeout 30
```

The timer continues the countdown and minus the time that have already elapsed (about 10 seconds) since the superior BPDU was detected. Issuing a **show span root-protect** command displays the following information.

```
device(config)# show span root-protect
Port    VLAN  Current State
1/4     1     Inconsistent state (20 seconds left on timer)
```

Next, the timeout period is increased to 120 seconds.

```
device(config)# spanning-tree root-protect timeout 120
```

Since the timer has not expired, it continues the countdown. The remaining time left is adjusted by the time that has already elapsed (about 18 seconds) since the superior BPDU was detected. Issuing a **show span root-protect** command displays the following information.

```
device(config)# show span root-protect
Port    VLAN  Current State
1/4     1     Inconsistent state (102 seconds left on timer)
```

Checking for Syslog messages

A Syslog message such as the following is generated after the Root Guard blocks a port.

```
Sep 9 18::39:27:I:STP: Root Guard Port 12/21, VLAN 10 inconsistent (Received superior BPDU)
```

A Syslog message such as the following is generated after the Root Guard unblocks a port.

```
Sep 9 18::39:27:I:STP: Root Guard Port 12/21, VLAN 10 consistent (Timeout)
```

Checking for traps

The following SNMP traps are generated for Root Guard:

- snTrapStpRootGuardDetect is generated after the Root Guard blocks a port.
- snTrapStpRootGuardExpire is generated after a blocked port (due to Root Guard) goes back to a Forwarding state

Refer to the *Unified IP MIB Reference* for details.

BPDU Guard

STP protection provides the ability to prohibit an end station from initiating or participating in an STP topology. The Bridge Protocol Data Units (BPDU) Guard is used to keep all active network topologies predictable.

> NOTE
> The feature is also available for MSTP and RSTP.

STP detects and eliminates logical loops in a redundant network by selectively blocking some data paths and allowing only some data paths to forward traffic.

In an STP environment, switches, end stations, and other Layer 2 devices use BPDUs to exchange information that STP will use to determine the best path for data flow. When a Layer 2 device is powered ON and connected to the network, or when a Layer 2 device goes down, it sends out an BPDU, triggering a topology change.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in a topology change. In this case, you can enable the BPDU Guard feature on the Extreme port to which the end station is connected. The BPDU Guard feature disables the connected device's ability to initiate or participate in an topology change, by dropping all BPDUs received from the connected device.

As an extended security measure, the administrator can disable a port if a BPDU is received on a port where BPDU Guard is configured. A Syslog message and SNMP trap are triggered when the port is disabled.

You can re-enable the disabled port from the CLI; however, make sure the offending BPDUs have stopped before re-enabling the port. Otherwise, the port will be disabled again the moment a new BPDU is received.

> NOTE
> BPDU Guard should be configured only on the primary port of a LAG. If a port configured with BPDU guard is made a secondary port, the LAG deployment will be vetoed.

Enabling BPDU Guard

You can enable BPDU Guard on a per-port basis.

To prevent an end station from initiating or participating in topology changes, enter the following command at the interface level of the CLI.

```
device(config) interface ethe 2/1
device(config-if-e1000-2/1)# spanning-tree protect
```

Syntax: [no] spanning-tree protect

This command causes the port to drop BPDUs sent from the device on the other end of the link.

Enter the **no** form of the command to disable BPDU Guard on the port and remove the **spanning-tree protect do-disable** feature if they are configured.

Enabling BPDU Guard and disabling a port that receives BPDUs

You can enable BPDU Guard on a port and at the same time configure a port to be disabled when it receives a BPDU. Enter the following commands.

```
device(config) interface ethe 2/1
device(config-if-e1000-2/1)#spanning-tree protect do-disable
```

Syntax: [no] spanning-tree protect do-disable

If both **spanning-tree protect** and **spanning-tree protect do-disable** are configured on an interface, **spanning-tree protect do-disable** takes precedence. This means that when the port receives a BPDU, the port will drop the BPDU and disable the port.

If you issue a **no spanning-tree protect do-disable** command, the port will be re-enabled and will no longer be disabled when it receives a BPDU. The following message is displayed when you enter the **no spanning-tree protect do-disable** command.

```
This command removes only "spanning-tree protect do-disable". To remove "spanning-tree protect", please
issue a separate command "no spanning-tree protect".
```

Re-Enabling a port disabled due to BPDU guard

A port disabled by the **spanning-tree protect do-disable** command can be enabled by the following commands:

- Entering the **no spanning-tree protect do-disable** command.
- Entering the **spanning-tree protect re-enable** command. Make sure the offending BPDUs have stopped before issuing this command; otherwise, the port will be disabled again once it receives a new BPDU.

```
device(config)# interface ethernet 1/4
device(config-if-e10000-1/4)#spanning-tree protect re-enable
```

Syntax: [no] spanning-tree protect re-enable

Issuing the **spanning-tree protect re-enable** command does not remove the **spanning-tree protect do-disable** configuration on the port. If a new BPDU is received on the port, the port will be disabled again. To prevent this from happening, you can do one of the following:

- Remove the **spanning-tree protect do-disable** configuration by issuing the **no spanning-tree protect do-disable** command, followed by the **spanning-tree protect re-enable** command to re-enable the port.
- Remove the source of the offending BPDUs from the network.

This command does not have a **no** form.

Displaying BPDU Guard configuration

To determine if BPDU Guard is configured on the device, enter the following command.

```
device#show spanning-tree protect
  protect   Show STP BPDU Guard information
devicer# show span protect
Port    Disable Port on BPDU Rx       Current Port State
1/1     No                            down
1/2     Yes                           down
1/3     No                            up
1/4     Yes                           up
```

Syntax: show spanning-tree protect

The command shows the following information.

TABLE 34 CLI display of show spanning-tree bp

This field...	Displays...
Port	The port on which BPDU Guard is configured
Disable Port on BPDU Rx	Indicates if **spanning-tree protect do-disable** is configured on the port: • Yes - **spanning-tree protect do-disable** is configured on the port. The BPDU will be dropped and the port will be disabled when it receives a BPDU. • No - **spanning-tree protect do-disable** is not configured. The BPDU will be dropped but the port will not be disabled.
Current Port State	Indicates if the port is currently UP or DOWN.

Determining if BPDU Guard is enabled

The **show interface** command displays the state of a port.

If BPDU Guard is disabled or has not been configured, the output shows the following information.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
```

If BPDU Guard has been enabled using the **spanning-tree protect** command, the output shows the following.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is enabled
```

If BPDU Guard is enabled using the **spanning-tree protect do-disable** command, the output shows.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is enabled with port to be disabled on BPDU receive
```

Syntax: show interface ethernet slot/port

Checking for Syslog messages

When the **spanning-tree protect do-disable** command is issued, the port becomes disabled and the following Syslog messages are generated.

```
Sep 9 18::39:27:I:STP: BPDU Guard port 1/4 disable
Sep 9 18::39:27:I:System: Interface ethernet 1/4, state down - disabled
```

When the **spanning-tree protect re-enable** command is issued to re-enable a port, the following Syslog messages are generated.

```
Sep 9 18:43:21:I:STP: BPDU Guard re-enabled on ports ethe 1/4
Sep 9 18:43:23:I:System: Interface ethernet 1/4, state up
```

Checking for traps

The following SNMP traps are generated for BPDU Guard. Refer to the *Unified IP MIB Reference* for details:

- snTrapStpBPDUGuardDetect is generated when a port is disabled because **spanning-tree protect do-disable command** on a port and that port received a BPDU and disabled the port.
- snTrapSTPBPDUGuardExpire is generated when a port that has been disabled due to a BPDU Guard violation is re-enabled using the **spanning-tree protect re-enable** command.

Displaying STP information

You can display the following STP information:

- All the global and interface STP settings
- Detailed STP information for each interface
- STP state information for a VLAN
- STP state information for an individual interface

Displaying STP information for an entire device

To display STP information, enter the following command at any level of the CLI.

```
device# show spanning-tree vlan 10
VLAN 10 - STP instance 1
------------------------------------------------------------------------
STP Bridge Parameters:
Bridge           Bridge Bridge Bridge Hold  LastTopology Topology
Identifier       MaxAge Hello  FwdDly Time  Change       Change
hex              sec    sec    sec    sec   sec          cnt
8000000480a04000 20     2      15     1     0            0
RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
8000000480a04000 0         8000000480a04000 Root  20  2   15
STP Port Parameters:
Port  Prio Path      State      Designat- Designated       Designated
Num   rity Cost                ed Cost   Root             Bridge
1/3   128  4         DISABLED   0         0000000000000000 0000000000000000
1/13  128  4         DISABLED   0         0000000000000000 0000000000000000
```

To display only ports blocked by the STP protocol, enter the following command at any level of the CLI.

```
device#show spanning-tree blocked vlan 10
VLAN 10 - STP instance 0
------------------------------------------------------------------------
STP Bridge Parameters:
Bridge           Bridge Bridge Bridge Hold  LastTopology Topology
Identifier       MaxAge Hello  FwdDly Time  Change       Change
hex              sec    sec    sec    sec   sec          cnt
80000024389e2d00 20     2      15     1     718          1
RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
80000024388f6b00 2         80000024388f6b00 3/1   20  2   15
STP Port Parameters:
Port  Prio Path      State      Designat- Designated       Designated
Num   rity Cost                ed Cost   Root             Bridge
3/2   128  2         BLOCKING   0         80000024388f6b00 80000024388f6b00
3/3   128  2         BLOCKING   0         80000024388f6b00 80000024388f6b00
3/4   128  2         BLOCKING   0         80000024388f6b00 80000024388f6b00
```

Syntax: show spanning-tree [**blocked**] [**vlan vlan-id**] | [**pvst-mode**] | **detail** [**vlan vlan-id** [**ethernet slot/port**]] [**begin expression** | **exclude expression** | **include expression**]

The **blocked** parameter displays only ports blocked by the STP protocol. When the **blocked** parameter is not specified, information is displayed for all STP controlled ports.

The **vlan***vlan-id* parameter displays STP information for the specified port-based VLAN.

The **pvst-mode** parameter displays STP information for the Extreme device's Per VLAN Spanning Tree (PVST+) compatibility configuration. Refer to PVST or PVST+ compatibility on page 245.

The **detail** parameter and its additional optional parameters display detailed information for individual ports. Refer to Displaying detailed STP information for each interface on page 228.

The **show spanning-tree** command shows the following information.

TABLE 35 CLI display of STP information

This field...	Displays...
Global STP Parameters	
VLAN ID	The port-based VLAN that contains this spanning tree and the number of STP instance on the VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all STP information is for VLAN 1.
Bridge Parameters	
Bridge Identifier	The ID assigned by STP to this bridge for this spanning tree in hexadecimal. NOTE If this address is the same as the Root ID, then this device or VLAN is the root bridge for its spanning tree.
Bridge MaxAge sec	The number of seconds this bridge waits for a hello message from the root bridge before deciding the root has become unavailable and performing a reconvergence.
Bridge Hello sec	The interval between each configuration BPDU sent by the bridge.
Bridge FwdDly sec	The number of seconds this bridge waits following a topology change and consequent reconvergence.
Hold Time sec	The minimum number of seconds that must elapse between transmissions of consecutive Configuration BPDUs on a port.
Last Topology Change sec	The number of seconds since the last time a topology change occurred.
Topology Change cnt	The number of times the topology has changed since this device was reloaded.
Root Bridge Parameters	
Root Identifier	The ID assigned by STP to the root bridge for this spanning tree in hexadecimal.
Root Cost	The cumulative cost from this bridge to the root bridge. If this device is the root bridge, then the root cost is 0.
DesignatedBridge Identifier	The designated bridge to which the root port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.
Root Port	The port on this device that connects to the root bridge. If this device is the root bridge, then the value is "Root" instead of a port number.
Max Age sec	The number of seconds this root bridge waits for a hello message from the bridges before deciding a bridges has become unavailable and performing a reconvergence.
Hello sec	The interval between each configuration BPDU sent by the root bridge.
FwdDly sec	The number of seconds this root bridge waits following a topology change and consequent reconvergence.
Port STP Parameters	
Port Num	The port number.
Priority	The port's STP priority.

TABLE 35 CLI display of STP information (continued)

This field...	Displays...
	NOTE If you configure this value, specify it in decimal format. Refer to Changing STP port parameters on page 220.
Path Cost	The port's STP path cost.
State	The port's STP state. The state can be one of the following: • BLOCKING - STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED - The port is not participating in STP. This can occur when the port is disconnected or STP is disabled on the port. • FORWARDING - STP is allowing the port to send and receive frames. • LISTENING - STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING - The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
Design Cost	The cost to the root bridge as advertised by the designated bridge that is connected to this port. If the designated bridge is the root bridge itself, then the cost is 0. The identity of the designated bridge is shown in the Design Bridge field.
Designated Root	The root bridge as recognized on this port. The value is the same as the root bridge ID listed in the Root ID field.
Designated Bridge	The bridge as recognized on this port.

Displaying detailed STP information for each interface

To display the detailed STP information, enter the following command at any level of the CLI.

```
device# show spanning-tree detail vlan 10
VLAN 10 - STP instance 1
------------------------------------------------------------------
STP Bridge Parameters:
Bridge identifier - 0x8000000480a04000
Root bridge - 0x8000000480a04000
Control ports - ethernet 1/3 ethernet 1/13
Active global timers - None
STP Port Parameters:
Port 1/3 - DISABLED
Port 1/13 - DISABLED
VLAN 20 - STP instance 2
------------------------------------------------------------------
STP Bridge Parameters:
Bridge identifier - 0x8000000480a04000
Root bridge - 0x8000000480a04000
Control ports - ethernet 1/3 ethernet 1/13
Active global timers - None
STP Port Parameters:
```

```
Port 1/3 - DISABLED
Port 1/13 - DISABLED
```

If a port is disabled, the only information shown by this command is "DISABLED". If a port is enabled, this display shows the following information.

Syntax: show spanning-tree detail [**vlan vlan-id** [**ethernet slot/port**]]

The **vlan***vlan-id* parameter specifies a VLAN.

The **ethernet***slot*/*portnum* parameter specifies an individual port within the VLAN (if specified).

> NOTE
> If the configuration includes VLAN groups, the **show span detail** command displays the master VLANs of each group but not the member VLANs within the groups. However, the command does indicate that the VLAN is a master VLAN. The **show span detail** command with the **vlan***vlan-id* parameters displays the information for the VLAN even if it is a member VLAN. To list all the member VLANs within a VLAN group, enter the **show vlan-group** command with the *group-id* variable.

The **show spanning-tree detail** command shows the following information for each VLAN participating in the spanning tree.

TABLE 36 CLI display of detailed STP information for ports

This field...	Displays...
VLAN ID	The VLAN that contains the listed ports and the number of STP instances on this VLAN. The STP type can be one of the following: • Proprietary multiple Spanning Tree • IEEE 802.1Q Single Spanning Tree (SSTP) NOTE If STP is disabled on a VLAN, the command displays the following message instead: "Spanning-tree of port-vlan *vlan-id* is disabled."
STP Bridge Parameters:	
Bridge identifier	The STP identity of this device.
Root	The ID assigned by STP to the root bridge for this spanning tree.
Control ports	The ports in the VLAN.
Active global timers	The global STP timers that are currently active, and their current values. The following timers can be listed: • Hello - The interval between Hello packets. This timer applies only to the root bridge. • Topology Change (TC) - The amount of time during which the topology change flag in Hello packets will be marked, indicating a topology change. This timer applies only to the root bridge. • Topology Change Notification (TCN) - The interval between Topology Change Notification packets sent by a non-root bridge toward the root bridge. This timer applies only to non-root bridges.
STP Port Parameters:	
Port number and STP state	The internal port number and the port's STP state. The internal port number is one of the following: • The port's interface number, if the port is the designated port for the LAN. • The interface number of the designated port from the received BPDU, if the interface is not the designated port for the LAN.

TABLE 36 CLI display of detailed STP information for ports (continued)

This field...	Displays...
	The state can be one of the following: • BLOCKING - STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED - The port is not participating in STP. This can occur when the port is disconnected or STP is administratively disabled on the port. • FORWARDING - STP is allowing the port to send and receive frames. • LISTENING - STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING - The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table. NOTE If the state is DISABLED, no further STP information is displayed for the port.

IEEE Single Spanning Tree (SSTP)

By default, each port-based VLAN on the Extreme device runs a separate spanning tree, which you can enable or disable on an individual VLAN basis.

Alternatively, you can configure the Extreme device to run a single spanning tree across all of its ports and VLANs. The SSTP feature is especially useful for connecting the Extreme device to third-party devices that run a single spanning tree in accordance with the 802.1q specification.

SSTP uses the same parameters, with the same value ranges and defaults, as the default STP supported on the Extreme device. Refer to Default STP bridge and port parameters on page 219.

SSTP defaults

SSTP is disabled by default. When you enable the feature, all VLANs on which STP is enabled become members of a single spanning tree. All VLANs on which STP is disabled are excluded from the single spanning tree:

- To add a VLAN to the single spanning tree, enable STP on that VLAN.
- To remove a VLAN from the single spanning tree, disable STP on that VLAN.

When you enable SSTP, all the ports that are in port-based VLANs with STP enabled become members of a single spanning tree domain. Thus, the ports share a single BPDU broadcast domain. The Extreme device places all the ports in a non-configurable VLAN, 4095, to implement the SSTP domain. However, this VLAN does not affect port membership in the port-based VLANs you have configured. Other broadcast traffic is still contained within the individual port-based VLANs. Therefore, you can use SSTP while still using

your existing VLAN configurations without changing your network. In addition, SSTP does not affect 802.1q tagging. Tagged and untagged ports alike can be members of the single spanning tree domain.

> NOTE
> When SSTP is enabled, the BPDUs on tagged ports go out untagged.

If you disable SSTP, all VLANs that were members of the single spanning tree now do not run any form of spanning tree. Per VLAN STP can be enabled again using either global STP enable or the spanning-tree enable command under individual VLANs.

> NOTE
> If the Extreme device has only one port-based VLAN (the default VLAN), then it is already running a single instance of STP. In this case, you do not need to enable SSTP. You need to enable SSTP only if the Extreme device contains more than one port-based VLAN and you want all the ports to be in the same STP broadcast domain.

To configure the Extreme device to run a single spanning tree, enter the following command at the global CONFIG level.

```
device(config)# spanning-tree single
```

To change a global STP parameter, enter a command such as the following at the global CONFIG level.

```
device(config) spanning-tree single priority 2
```

This command changes the STP priority for all ports to 2.

To change an STP parameter for a specific port, enter commands such as the following.

```
device(config) spanning-tree single ethernet 1/1 priority 10
```

The commands shown above override the global setting for STP priority and set the priority to 10 for port 1/1.

Here is the syntax for the global STP parameters:

Syntax: [no] spanning-tree single [**forward-delay value**] [**hello-time value**] | [**maximum-age time**] | [**priority value**]

Here is the syntax for the STP port parameters:

Syntax: [no] spanning-tree single [**ethernet slot/portnum path-cost value** | **priority value**]

For the parameter definitions and possible values, refer to Default STP bridge and port parameters on page 219.

> NOTE
> Both commands listed above are entered at the global CONFIG level.

Also, you can use the **rstp single** command to control the topology for VLANs.

Displaying SSTP information

To verify that SSTP is in effect, enter the following commands at any level of the CLI.

```
device(config)# show spanning-tree
VLAN 4095 - STP instance 0
------------------------------------------------------------------------
STP Bridge Parameters:
Bridge           Bridge Bridge Bridge Hold  LastTopology Topology
Identifier       MaxAge Hello  FwdDly Time  Change       Change
hex              sec    sec    sec    sec   sec          cnt
8000000480a04000 20     2      15     1     0            0
RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
8000000480a04000 0         8000000480a04000 Root  20  2   15
STP Port Parameters:
```

```
Port  Prio Path      State      Designat- Designated       Designated
Num   rity Cost                ed Cost   Root             Bridge
1/3   128  4         DISABLED   0         0000000000000000 0000000000000000
1/13  128  4         DISABLED   0         0000000000000000 0000000000000000
SSTP members: 10 20 30 99 to 100
```

For information on the command syntax, refer to Displaying STP information on page 226.

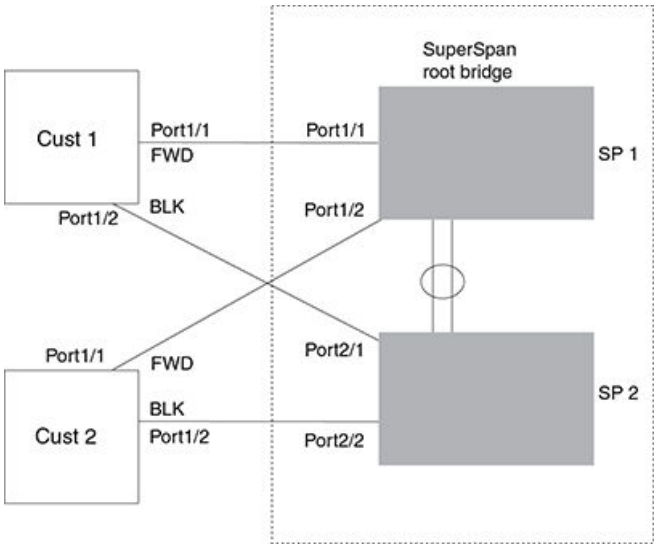
SuperSpan™

SuperSpan is the Extreme STP enhancement that allows Service Providers (SPs) to use STP in both SP networks and customer networks. The SP devices are Extreme devices and are configured to tunnel each customer's STP BPDUs through the SP. From the customer's perspective, the SP network is a loop-free non-blocking device or network. The SP network behaves like a hub in the sense that the necessary blocking occurs in the customer network, not in the SP.

The interfaces that connect the SP to a customer's network are configured as SuperSpan boundary interfaces. Each SuperSpan boundary interface is configured with a customer ID, to uniquely identify the customer's network within SuperSpan.

Figure 54 shows an example SuperSpan implementation. In this example, an SP's network is connected to multiple customers. Each customer network is running its own instance of standard STP. The Extreme devices in the SP are running SuperSpan.

FIGURE 53 SuperSpan example



In this example, the SP network contains two devices that are running SuperSpan. The SP is connected to two customer networks. Each customer network is running its own instance of STP. SuperSpan prevents Layer 2 loops in the traffic flow with each customer while at the same time isolating each customer's traffic and spanning tree from the traffic and spanning trees of other customers. For example, the SP devices provide loop prevention for Customer 1 while ensuring that Customer 1's traffic is never forwarded to Customer 2. In this example, customer 1 has two interfaces to the SP network, ports 1/1 and 1/2 connected to SP 1. The SP network behaves like a non-blocking hub. BPDUs are tunneled through the network. To prevent a Layer 2 loop, customer 1's port 1/2 enters the blocking state.

Customer ID

SuperSpan uses a SuperSpan customer ID to uniquely identify and forward traffic for each customer. You assign the customer ID as part of the SuperSpan configuration of the Extreme devices in the SP. In SuperSpan™ on page 232, the spanning trees of customer 1 and customer 2 do not interfere with one another because the SP network isolates each customer's spanning tree based on the SuperSpan customer IDs in the traffic.

BPDU forwarding

When the Extreme device receives a customer's BPDU on a boundary interface, the Extreme device changes the destination MAC address of the BPDU from the bridge group address (00-00-00-00-00-00) as follows:

- The first byte (locally administered bit) is changed from 01 to 03, to indicate that the BPDU needs to be tunneled.
- The fourth and fifth bytes are changed to the customer STP ID specified on the boundary interface.

For example, if the customer's STP ID is 1, the destination MAC address of the customer's BPDUs is changed to the following: 00-00-00-00-01-00.

Each Extreme device that is configured for SuperSpan forwards the BPDU using the changed destination MAC address. At the other end of the tunnel, the Extreme device connected to the customer's network changes the destination MAC address back to the bridge group address (00-00-00-00-00-00).

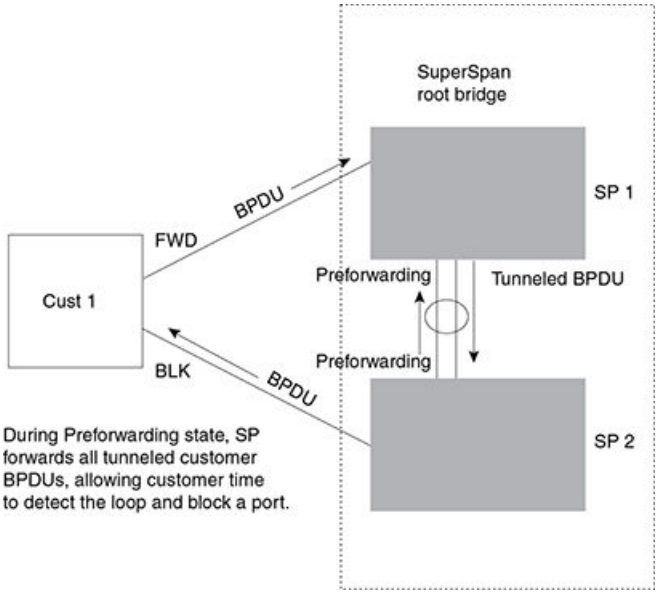
Preforwarding state

To ensure that the customer's network has time to converge at Layer 2 and prevent loops, the Extreme devices configured for SuperSpan use a special forwarding state, Preforwarding. The Preforwarding state occurs between the Learning and Forwarding states and by default lasts for five seconds. During the Preforwarding state, the Extreme device forwards tunneled BPDUs from customers only and does not forward data traffic. This ensures that the customer's network will detect the Layer 2 loop and block a port. The SP network remains unblocked. After the Preforwarding state, the ports change to the Forwarding state and forward data traffic as well as BPDUs.

The default length of the Preforwarding state is five seconds. You can change the length of the Preforwarding state to a value from 3 - 30 seconds.

Figure 55 shows an example of how the Preforwarding state is used.

FIGURE 54 SuperSpan Preforwarding state



In this example, a customer has two links to the SP. Since the SP is running SuperSpan, the SP ports enter the Preforwarding state briefly to allow the customer ports connected to the SP to detect the Layer 2 loop and block one of the ports.

NOTE
If you add a new Extreme device to a network that is already running SuperSpan, you must enable SuperSpan on the Extreme device, at least on the VLANs that will be tunneling the customer traffic. Otherwise, the new Extreme device does not use the Preforwarding state. This can cause the wrong ports to be blocked.

Combining single STP and multiple spanning trees

You can use SuperSpan in any of the following combinations:

- Customer and SP networks both use multiple spanning trees (a separate spanning tree in each VLAN).
- Customer uses multiple spanning trees but SP uses Single STP (all STP-enabled VLANs are in the same spanning tree).
- Customer uses Single STP but SP uses multiple spanning trees.
- Customer and SP networks both use Single STP.

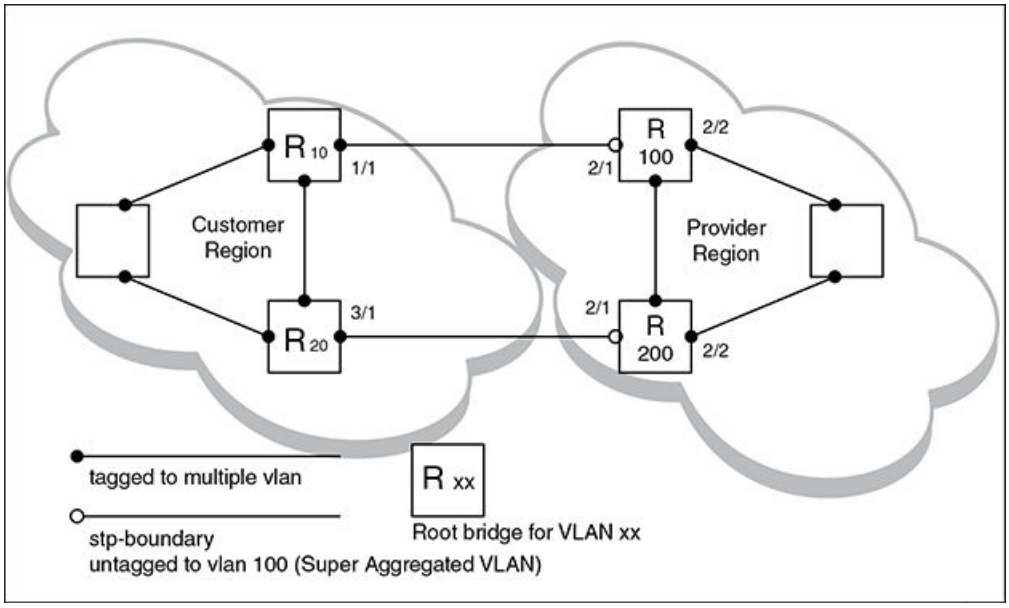
The following sections provide an example of each combination.

NOTE
All the combinations listed above are supported when the boundary ports joining the SP SuperSpan domain to the client spanning trees are untagged. For example, all these combinations are valid in super aggregated VLAN configurations. If the boundary ports are tagged, you cannot use Single STP in the client network in combination with multiple spanning trees in the SP SuperSpan domain.

Customer and SP use multiple spanning trees

Figure 56 shows an example of SuperSpan where both the customer network and the SP network use multiple spanning trees (a separate spanning tree in each port-based VLAN).

FIGURE 55 Customer and SP using multiple spanning trees



Both the customer and SP regions are running multiple spanning trees (one per port-based VLAN) in the Layer 2 switched network. The customer network contains VLANs 10 and 20 while the SP network contains VLANs 100 and 200. Customer traffic from VLAN 10 and VLAN 20 is aggregated by VLAN 100 in the SP since the boundary ports, 2/1 on R100 and R200, are untagged members of VLAN 100. By adjusting the bridge priority on VLANs 10 and 20, the customer can select a different root bridge for each spanning tree running in the customer network.

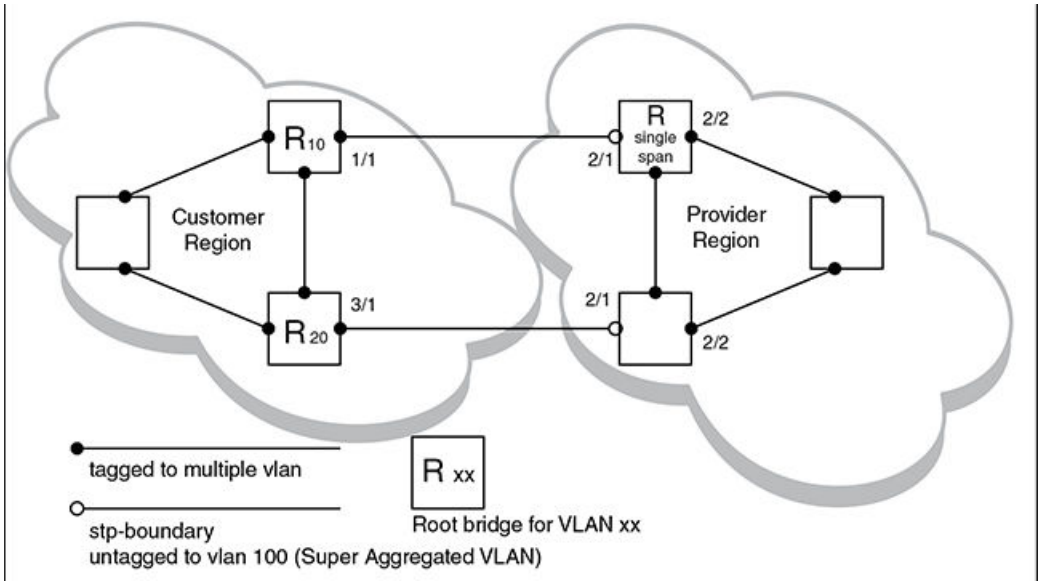
In the above example, STP in VLAN 10 will select R10 as the root bridge and make 1/1 on R10 forwarding while blocking port 3/1 on R20. The opposite occurs for STP in VLAN 20. As a result, both links connecting the customer and SP regions are fully utilized and serve as backup links at the same time, providing loop-free, non-blocking connectivity. In the SP network, multiple STP instances are running (one for VLAN 100 and one for VLAN 200) to ensure loop-free, non-blocking connectivity in each VLAN.

SuperSPAN boundaries are configured at port 2/1 of R100 and R200. Since the customer's traffic will be aggregated into VLAN 100 at the SP, the SP network appears to the customer to be a loop-free non-blocking hub to the customer network when port 2/2 on R200 is blocked by STP in VLAN 100.

Customer uses multiple spanning trees but SP uses single STP

Figure 57 shows an example of SuperSpan where the customer network uses multiple spanning trees while the SP network uses Single STP.

FIGURE 56 Customer using multiple spanning trees and SP using Single STP



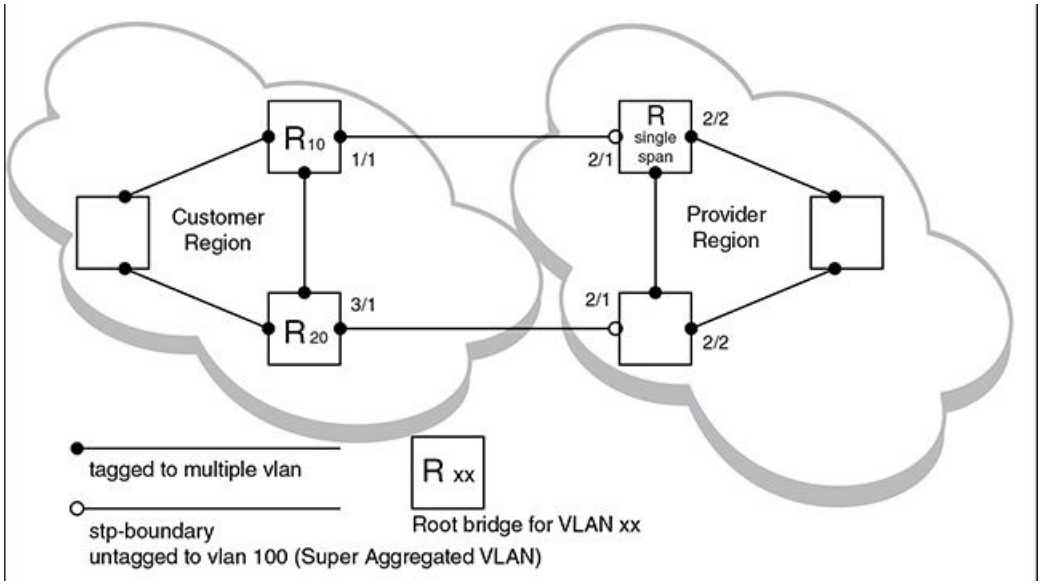
Customer traffic from different VLANs is maintained by different spanning trees, while the SP network is maintained by a single spanning tree. The SP can still use multiple VLANs at the core to separate traffic from different customers. However, all VLANs will have the same network topology because they are all calculated by the single spanning tree. The loop-free, non-blocking network acts like a hub for the customer network, with boundary ports 2/1 on each device being untagged members of VLAN 100.

Traffic from all VLANs in the customer network will be aggregated through VLAN 100 at the SP. This setup leaves the customer network's switching pattern virtually unchanged from the scenario in Customer and SP use multiple spanning trees on page 234, since the SP network still is perceived as a virtual hub, and maintenance of the hub's loop-free topology is transparent to the customer network.

Customer uses single STP but SP uses multiple spanning trees

Figure 58 shows an example of SuperSpan where the customer network uses Single STP while the SP uses multiple spanning trees.

FIGURE 57 Customer using Single STP and SP using multiple spanning trees

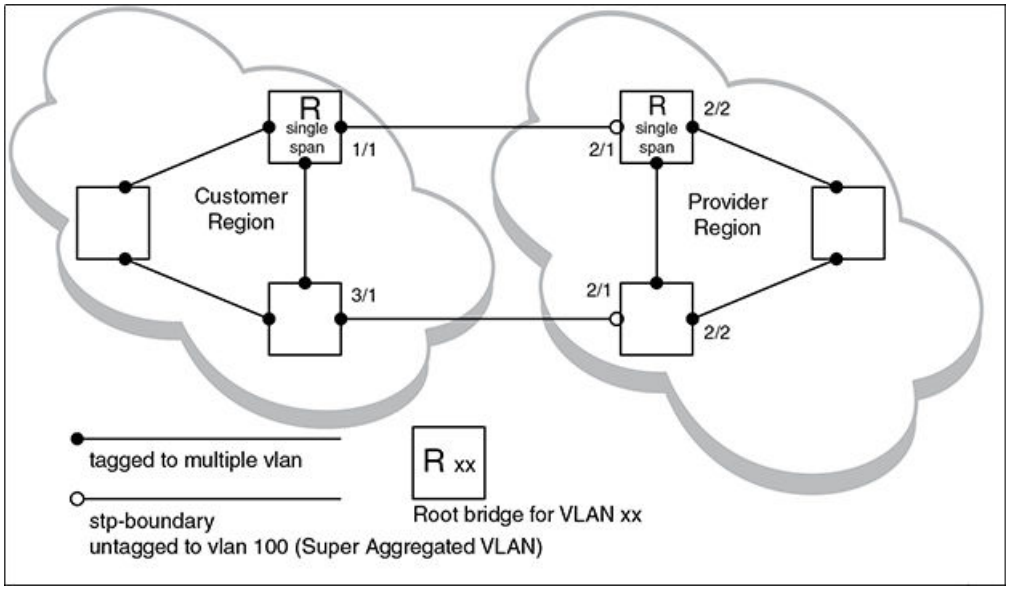


In this setup, the customer network is running a single spanning tree for VLANs 10 and 20. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP's network. The main difference between this scenario and the previous tow scenarios is that all traffic at the customer's network now follows the same path, having the same STP root bridge in all VLANs. Therefore, the customer network will not have the ability to maximize network utilization on all its links. On the other hand, loop-free, non-blocking topology is still separately maintained by the customer network's single spanning tree and the SP's per-VLAN spanning tree on VLAN 100.

Customer and SP use single STP

Figure 59 shows an example of SuperSpan where the customer network and SP both use Single STP.

FIGURE 58 Customer and SP using Single STP



In this setup, both the customer and SP networks are running a single spanning tree at Layer 2. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP network as in the previous scenario. Loop-free, non-blocking topology is still separately maintained by the customer's single spanning tree and the SP's single spanning tree.

Configuring SuperSpan

To configure the Extreme device for SuperSpan:

- Configure each interface on the Extreme device that is connected to customer equipment as a boundary interface. This step enables the interface to convert the destination MAC address in the customer's BPDUs.

The software requires you to specify a SuperSpan customer ID when configuring the boundary interface. Use an ID from 1 - 65535. The customer ID uniquely identifies the customer. Use the same customer ID for each SP interface with the same customer. When tunneling BPDUs through the network, the Extreme devices use the customer ID to ensure that BPDUs are forwarded only to the customer's devices, and not to other customers' devices.

- Globally enable SuperSpan. This step enables the Preforwarding state.

Configuring a boundary interface

To configure the boundary interfaces on SP 1 in SuperSpan™ on page 232, enter the following commands.

```
device(config)# interface 1/1
device(config-if-e1000-1/1)# stp-boundary 1
device(config)# interface 1/2
device(config-if-e1000-1/2)# stp-boundary 2
```

These commands configure two interfaces on the Extreme device as SuperSpan boundary interfaces. Interface 1/1 is a boundary interface with customer 1. Interface 1/2 is a boundary interface with customer 2. Each boundary interface is associated with a number, which is the SuperSpan ID. The SuperSpan ID identifies the instance of SuperSpan you are associating with the interface. Use the same SuperSpan ID for each boundary interface with the same customer. Use a different SuperSpan ID for each customer. For example, use SuperSpan ID 1 for all the boundary interfaces with customer 1 and use SuperSpan ID 2 for all boundary interfaces with customer 2.

Syntax: [no] stp-boundary num

The *num* parameter specifies the SuperSpan ID. Possible values: 1 - 65535.

To configure the boundary interfaces on SP 2 in SuperSpan™ on page 232, enter the following commands.

```
device(config)# interface 2/1
device(config-if-e1000-2/1)# stp-boundary 1
device(config)# interface 2/2
device(config-if-e1000-2/2)# stp-boundary 2
```

Enabling SuperSpan

After you configure the SuperSpan boundary interfaces, enable SuperSpan. You can enable SuperSpan globally or on an individual VLAN level. If you enable the feature globally, the feature is enabled on all VLANs.

> NOTE
> If you enable the feature globally, then create a new VLAN, the new VLAN inherits the global SuperSpan state. For example, if SuperSpan is globally enabled when you create a VLAN, SuperSpan also is enabled in the new VLAN.

You also can change the length of the preforwarding state.

To globally enable SuperSpan, enter the following command.

```
device(config)# super-span
```

Syntax: [no] super-span [**preforward-delay secs**]

The *secs* parameter specifies the length of the preforwarding state. You can specify from 3 - 15 seconds. The default is 5 seconds.

SuperSpan is enabled in all VLANs on the Extreme device. To disable SuperSpan in an individual VLAN, enter commands such as the following.

```
device(config)# vlan 10
device(config-vlan-10)# no super-span
```

Syntax: [no] super-span

Displaying SuperSpan information

To display the boundary interface configuration and BPDU statistics, enter the following command.

```
device(config)# show super-span
CID 1 Boundary Ports:
     Port  Customer    Tunnel
           BPDU Rx     BPDU Rx
     1/1   1            1
     1/2   0            0
     Total 1            1
CID 2 Boundary Ports:
     Port  Customer    Tunnel
           BPDU Rx     BPDU Rx
     2/1   0            3
     2/2   0            0
     Total 0            3
```

In this example, the Extreme device has two SuperSpan customer IDs.

Syntax: show superspan [**cid num**]

The **cid***num* parameter specifies a SuperSpan customer ID. If you do not specify a customer ID, information for all the customer IDs configured on the Extreme device is shown.

This command shows the following information.

TABLE 37 CLI display of SuperSpan customer ID information

This field...	Displays...
CID	The SuperSpan customer ID number.
Port	The boundary port number.
Customer BPDU Rx	The number of BPDUs received from the client spanning tree.
Tunnel BPDU Rx	The number of BPDUs received from the SuperSpan tunnel.

To display general STP information, refer to Displaying STP information on page 226.

STP feature configuration

Spanning Tree Protocol (STP) features extend the operation of standard STP, enabling you to fine- tune standard STP and avoid some of its limitations.

This section describes how to configure these parameters using the CLI.

Fast port span

When STP is running on a device, message forwarding is delayed during the spanning tree recalculation period following a topology change. The STP forward delay parameter specifies the period of time a bridge waits before forwarding data packets. The forward delay controls the listening and learning periods of STP reconvergence. You can configure the forward delay to a value from 4 - 30 seconds. The default is 15 seconds. Thus, using the standard forward delay, convergence requires 30 seconds (15 seconds for listening and an additional 15 seconds for learning) when the default value is used.

This slow convergence is undesirable and unnecessary in some circumstances. The Fast Port Span feature allows certain ports to enter the forwarding state in four seconds. Specifically, Fast Port Span allows faster convergence on ports that are attached to end stations and thus do not present the potential to cause Layer 2 forwarding loops. Because the end stations cannot cause forwarding loops, they can safely go through the STP state changes (blocking to listening to learning to forwarding) more quickly than is allowed by the standard STP convergence time. Fast Port Span performs the convergence on these ports in four seconds (two seconds for listening and two seconds for learning).

In addition, Fast Port Span enhances overall network performance in the following ways:

- Fast Port Span reduces the number of STP topology change notifications on the network. When an end station attached to a Fast Span port comes up or down, the device does not generate a topology change notification for the port. In this situation, the notification is unnecessary since a change in the state of the host does not affect the network topology.
- Fast Port Span eliminates unnecessary MAC cache aging that can be caused by topology change notifications. Bridging devices age out the learned MAC addresses in their MAC caches if the addresses are not refreshed for a given period of time, sometimes called the MAC aging interval. When STP sends a topology change notification, devices that receive the notification use the value of the STP forward delay to quickly age out their MAC caches. For example, if a device normal MAC aging interval is 5 minutes, the aging interval changes temporarily to the value of the forward delay (for example, 15 seconds) in response to an STP topology change.

NOTE
While running STP, the devices set the MAC age to forward delay value when these devices receive BPDUs with TC flag set. Since the default value of forward delay is 15 seconds, it is set to 15 seconds for MAC aging. For 20x10G, 2x100G(half slot), 4x10-4x1G (IPSec security) interface modules, the MAC aging value cannot be set to less than 20 seconds. Hence for any forward delay value less than or equal to 15 seconds, the MAC aging value is set to 20 seconds.

In normal STP, the accelerated cache aging occurs even when a single host goes up or down. Because Fast Port Span does not send a topology change notification when a host on a Fast Port Span port goes up or down, the unnecessary cache aging that can occur in these circumstances under normal STP is eliminated.

Fast Port Span is a system-wide parameter and is enabled by default. Thus, when you boot a device, all the ports that are attached only to end stations run Fast Port Span. For ports that are not eligible for Fast Port Span, such as ports connected to other networking devices, the device automatically uses the normal STP settings. If a port matches any of the following criteria, the port is ineligible for Fast Port Span and uses normal STP instead:

- The port is 802.1Q tagged
- The port is a member of a trunk group
- The port has learned more than one active MAC address
- An STP Configuration BPDU has been received on the port, thus indicating the presence of another bridge on the port.

You also can explicitly exclude individual ports from Fast Port Span if needed. For example, if the only uplink ports for a wiring closet switch are Gbps ports, you can exclude the ports from Fast Port Span.

Disabling and re-enabling fast port span

Fast Port Span is a system-wide parameter and is enabled by default. Therefore, all ports that are eligible for Fast Port Span use it.

To disable or re-enable Fast Port Span, enter the following commands.

```
device(config)#no fast port-span
device(config)#write memory
```

Syntax: [no] fast port-span

NOTE
The **fast port-span** command has additional parameters that let you exclude specific ports. These parameters are shown in the following section.

To re-enable Fast Port Span, enter the following commands.

```
device(config)#fast port-span
device(config)#write memory
```

Excluding specific ports from fast port span

To exclude a port from Fast Port Span while leaving Fast Port Span enabled globally, enter commands such as the following.

```
device(config)#fast port-span exclude ethernet 1
device(config)#write memory
```

To exclude a set of ports from Fast Port Span, enter commands such as the following.

```
device(config)#fast port-span exclude ethernet 1 ethernet 2 ethernet 3
device(config)#write memory
```

To exclude a contiguous (unbroken) range of ports from Fast Span, enter commands such as the following.

```
device(config)#fast port-span exclude ethernet 1 to 24
device(config)#write memory
```

Syntax: [no] fast port-span [**exclude ethernet** *port* [**ethernet** *port*] | **to** [*port*]]

To re-enable Fast Port Span on a port, enter a command such as the following.

```
device(config)#no fast port-span exclude ethernet 1
device(config)#write memory
```

This command re-enables Fast Port Span on port 1 only and does not re-enable Fast Port Span on other excluded ports. You also can re-enable Fast Port Span on a list or range of ports using the syntax shown above this example.

To re-enable Fast Port Span on all excluded ports, disable and then re-enable Fast Port Span by entering the following commands.

```
device(config)#no fast port-span
device(config)#fast port-span
device(config)#write memory
```

Disabling and then re-enabling Fast Port Span clears the exclude settings and thus enables Fast Port Span on all eligible ports. To make sure Fast Port Span remains enabled on the ports following a system reset, save the configuration changes to the startup-config file after you re-enable Fast Port Span. Otherwise, when the system resets, those ports will again be excluded from Fast Port Span.

Fast Uplink Span

The Fast Port Span feature described in the previous section enhances STP performance for end stations. The Fast Uplink Span feature enhances STP performance for wiring closet switches with redundant uplinks. Using the default value for the standard STP forward delay, convergence following a transition from an active link to a redundant link can take 30 seconds (15 seconds for listening and an additional 15 seconds for learning).

You can use the Fast Uplink Span feature on a NetIron OS device deployed as a wiring closet switch to decrease the convergence time for the uplink ports to another device to just one second. The new Uplink port directly goes to forward mode (bypassing listening and learning modes). The wiring closet switch must be a NetIron OS device but the device at the other end of the link can be a NetIron OS device or another vendor's switch.

Configuration of the Fast Uplink Span feature takes place entirely on the device. To configure the Fast Uplink Span feature, specify a group of ports that have redundant uplinks on the wiring closet switch (NetIron OS device). If the active link becomes unavailable, the Fast Uplink Span feature transitions the forwarding to one of the other redundant uplink ports in just one second. All Fast Uplink Span-enabled ports are members of a single Fast Uplink Span group.

NOTE

To avoid the potential for temporary bridging loops, Extreme recommends that you use the Fast Uplink feature only for wiring closet switches (switches at the edge of the network cloud). In addition, enable the feature only on a group of ports intended for redundancy, so that at any given time only one of the ports is expected to be in the forwarding state.

NOTE

When the wiring closet switch (NetIron OS device) first comes up or when STP is first enabled, the uplink ports still must go through the standard STP state transition without any acceleration. This behavior guards against temporary routing loops as the switch tries to determine the states for all the ports. Fast Uplink Span acceleration applies only when a working uplink becomes unavailable.

Active uplink port failure

The active uplink port is the port elected as the root port using the standard STP rules. All other ports in the group are redundant uplink ports. If an active uplink port becomes unavailable, Fast Uplink Span transitions the forwarding of traffic to one of the redundant ports in the Fast Uplink Span group in one second bypassing listening and learning port states.

Switchover to the active uplink port

When a failed active uplink port becomes available again, switchover from the redundant port to the active uplink port is delayed by 30 seconds. The delay allows the remote port to transition to forwarding mode using the standard STP rules. After 30 seconds, the blocked active uplink port begins forwarding in just one second and the redundant port is blocked.

NOTE

Use caution when changing the spanning tree priority. If the switch becomes the root bridge, Fast Uplink Span will be disabled automatically.

Fast Uplink Span Rules for Trunk Groups

If you add a port to a Fast Uplink Span group that is a member of a trunk group, the following rules apply:

- If you add the primary port of a trunk group to the Fast Uplink Span group, all other ports in the trunk group are automatically included in the group. Similarly, if you remove the primary port in a trunk group from the Fast Uplink Span group, the other ports in the trunk group are automatically removed from the Fast Uplink Span group.
- You cannot add a subset of the ports in a trunk group to the Fast Uplink Span group. All ports in a trunk group have the same Fast Uplink Span property, as they do for other port properties.
- If the working trunk group is partially down but not completely down, no switch-over to the backup occurs. This behavior is the same as in the standard STP feature.
- If the working trunk group is completely down, a backup trunk group can go through an accelerated transition only if the following are true:
  - The trunk group is included in the fast uplink group.
  - All other ports except those in this trunk group are either disabled or blocked. The accelerated transition applies to all ports in this trunk group.

When the original working trunk group comes back (partially or fully), the transition back to the original topology is accelerated if the conditions listed above are met.

Configuring a Fast Uplink Port Group

To configure a group of ports for Fast Uplink Span, enter the following commands:

```
device(config)# fast uplink-span ethernet 4/1 to 4/4
device(config)# write memory
```

Syntax: [no] fast uplink-span [**ethernet** *port* [**ethernet** *port* **...** | **to** *port*]]

This example configures four ports, 4/1 - 4/4, as a Fast Uplink Span group. In this example, all four ports are connected to a wiring closet switch. Only one of the links is expected to be active at any time. The other links are redundant. For example, if the link on port 4/1 is the active link on the wiring closet switch but becomes unavailable, one of the other links takes over. Because the ports are configured in a Fast Uplink Span group, the STP convergence takes one second instead of taking atleast 30 seconds using the standard STP forward delay.

You can add ports to a Fast Uplink Span group by entering the fast uplink-span command additional times with additional ports. The device can have only one Fast Uplink Span group, so all the ports you identify as Fast Uplink Span ports are members of the same group.

To remove a Fast Uplink Span group or to remove individual ports from a group, use "no" in front of the appropriate fast uplink-span command. For example, to remove ports 4/3 and 4/4 from the Fast Uplink Span group configured above, enter the following commands:

```
device(config)# no fast uplink-span ethernet 4/3 to 4/4
device(config)# write memory
```

To check the status of ports with Fast Uplink Span enabled.

```
device(config)# show span fast-uplink-span
STP instance owned by VLAN 1
Global STP (IEEE 802.1D) Parameters:
VLAN Root             Root Root   Prio Max He- Ho- Fwd Last    Chg Bridge
 ID   ID              Cost Port   rity Age llo ld  dly Chang   cnt Address
                                  Hex  sec sec sec sec sec
   1 000000c100000001 2    1/1    8000 20  2   1   15  65      15  000011111111
Port STP Parameters:
Port   Prio Path  State       Fwd    Design  Designated       Designated
Num    rity Cost              Trans  Cost    Root             Bridge
       Hex
1/2    80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/3    80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/4    80   4     FORWARDING  1      2       000000c100000001 8000000011111111
1/5    80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/6    80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/7    80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/8    80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/9    80   0     DISABLED    0      0       0000000000000000 0000000000000000
```

Syntax: show span fast-uplink-span

Configuring Fast Uplink Span within a VLAN

You can also configure Fast Uplink Span on the interfaces within a VLAN.

To configure Fast Uplink Span for a VLAN, enter command such as the following.

```
device(config)#vlan 10
device(config-vlan-10)#untag ethernet 8/1 to 8/2
device(config-vlan-10)#fast uplink-span ethernet 8/1 to 8/2
```

Syntax: [no] fast uplink-span ethernet *port-no*

To check the status of Fast Uplink Span for a specified VLAN.

```
device(config-vlan-2)#show span vlan 2 fast-uplink-span
STP instance owned by VLAN 2
Global STP (IEEE 802.1D) Parameters:
VLAN Root             Root Root   Prio Max He- Ho- Fwd Last    Chg Bridge
 ID   ID              Cost Port   rity Age llo ld  dly Chang   cnt Address
                                  Hex  sec sec sec sec sec
   2 8000000011111111 0    Root   8000 20  2   1   15  29596   0   000011111111
Port STP Parameters:
Port   Prio Path  State       Fwd    Design  Designated       Designated
Num    rity Cost              Trans  Cost    Root             Bridge
       Hex
1/1    80   4     LISTENING   0      0       8000000011111111 8000000011111111
```

Syntax: show span vlan *vlan-id* **fast-uplink-span**

The **vlan***vlan-id* parameter displays Fast Uplink Span information for the specified VLAN.

Configuring STP under an ESI VLAN

STP can also be configured under a VLAN that is part of a user-configured ESI. For example, to enable spanning tree on a VLAN that is part of an ESI, configure the following commands.

```
device(config)# esi customer1 encapsulation cvlan
device(config-esi-customer1)# vlan 100
device(config-esi-customer1-vlan-100)# spanning-tree
```

Configuration considerations:

The configuration considerations are as follows:

- MSTP can only be configured under the default ESI. MSTP cannot be configured for VLANs that are configured under a user-defined ESI.
- STP can be configured for VLANs with encapsulation type B-VLAN, S-VLAN or C-VLAN.

When STP or RSTP is configured for VLANs under an ESI, the MRP members must be part of the same ESI.

PVST or PVST+ compatibility

Extreme's support for Cisco's Per VLAN Spanning Tree plus (PVST+) allows the Extreme device to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices. Ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected.

When it is configured for MSTP, the Extreme device can interoperate with PVST.

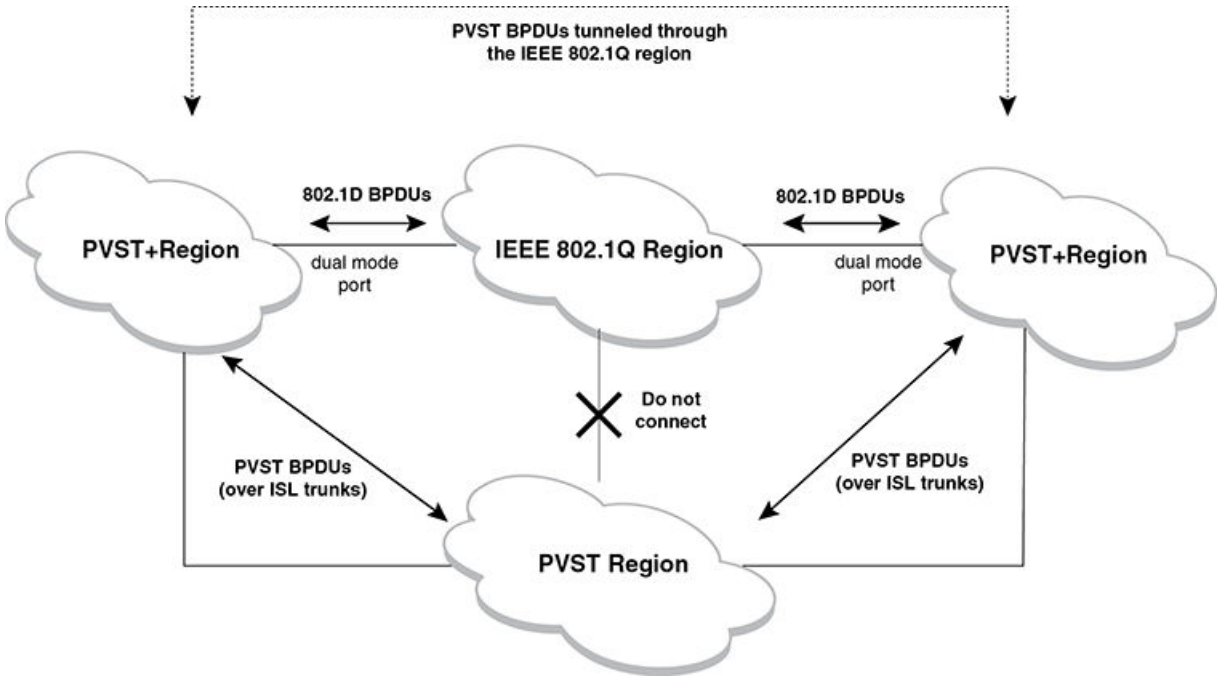
Overview of PVST and PVST+

Per VLAN Spanning Tree (PVST) is a Cisco proprietary protocol that allows a Cisco device to have multiple spanning trees. The Cisco device can interoperate with spanning trees on other PVST devices but cannot interoperate with IEEE 802.1Q devices. An IEEE 802.1Q device has all its ports running a single spanning tree. PVST+ is an extension of PVST that allows a Cisco device to also interoperate with devices that are running a single spanning tree (IEEE 802.1Q).

The PVST+ support allows the Extreme device to interoperate with PVST spanning trees and the IEEE 802.1Q spanning tree at the same time.

IEEE 802.1Q and PVST regions cannot interoperate directly but can interoperate indirectly through PVST+ regions. PVST BPDUs are tunneled through 802.1Q regions, while PVST BPDUs for VLAN 1 (the IEEE 802.1Q VLAN) are processed by PVST+ regions. Figure 60 shows the interaction of IEEE 802.1Q, PVST, and PVST+ regions.

FIGURE 59 Interaction of IEEE 802.1Q, PVST, and PVST+ regions



VLAN Tags and dual mode

The **dual-mode** feature enables the port to send and receive both tagged and untagged frames on a port. When the dual-mode feature is enabled, the port is an untagged member of one of its VLANs and is at the same time a tagged member of all its other VLANs. The untagged frames are supported on the port's **Port Native VLAN** .

To interoperate with other vendors, the dual-mode feature must be enabled on the port. Some vendors use VLAN 1 by default to support the IEEE 802.1Q based standard spanning tree protocols such as 802.1d and 802.1w for sending the untagged frames on VLAN 1. On Extreme devices by default, the Port Native VLAN is the same as the device's **Default VLAN1** , which by default is VLAN 1. Thus, to support IEEE 802.1Q in a typical configuration, the port must be able to send and receive untagged frames for VLAN 1 and tagged frames for the other VLANs and interoperate with the vendors also using VLAN 1. If you want to use tagged frames on VLAN 1, you can change the default VLAN ID to an ID other than 1. You also can specify the VLAN on which you want the port to send and receive untagged frames (the Port Native VLAN). The Port Native VLAN ID does not need to be the same as the Default VLAN. Make sure that untagged (Native) VLAN is also changed on the interoperating vendor side to match with that on the Extreme side.

To support the IEEE 802.1Q with non-standard proprietary protocols such as PVST and PVST+, a port must always send and receive untagged frames on VLAN 1 on both sides. In that case, enable the dual-mode 1 feature to allow untagged BPDUs on VLAN 1and use Native VLAN 1 on the interoperating vendor side. You should not use VLAN 1 for tagged frames in this case.

NOTE
Support for the IEEE 802.1Q spanning tree always uses VLAN 1, regardless of whether the Extreme devices are configured to use tagged or untagged frames on the VLAN.

Enabling PVST+ support

PVST+ support is automatically enabled when the port receives a PVST BPDU. You can manually enable the support at any time or disable the support if desired.

The tagged port also supports IEEE 802.1Q BPDUs, since the dual-mode feature on the port is enabled, by default.

A port that is in PVST+ compatibility mode due to auto-detection reverts to the default MSTP mode when one of the following events occurs:

- The link is disconnected or broken
- The link is administratively disabled
- The link is disabled by interaction with the link-keepalive protocol

This allows a port that was originally interoperable with PVST+ to revert to multiple spanning tree when connected to the Extreme device.

Enabling PVST+ support manually

To immediately enable PVST+ support on a port, enter commands such as the following.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# pvst-mode
```

Syntax: [no] pvst-mode

> NOTE
> If you disable PVST+ support, the software still automatically enables PVST+ support if the port receives a BPDU with PVST+ format.

Displaying PVST+ support information

To display PVST+ information for ports on the Extreme device, enter the following command at any level of the CLI.

```
device(config)# show span pvst-mode
PVST+ Enabled on:
Port      Method
1/1       Set by configuration
1/2       Set by configuration
2/10      Set by auto-detect
3/12      Set by configuration
4/24      Set by auto-detect
```

Syntax: show span pvst-mode

This command displays the following information.

TABLE 38 CLI display of PVST+ information

This field...	Displays...
Port	The port number. NOTE The command lists information only for the ports on which PVST+ support is enabled.
Method	The method by which PVST+ support was enabled on the port. The method can be one of the following: • Set by configuration - You enabled the support. • Set by auto-detect - The support was enabled automatically when the port received a PVST+ BPDU.

Configuration examples

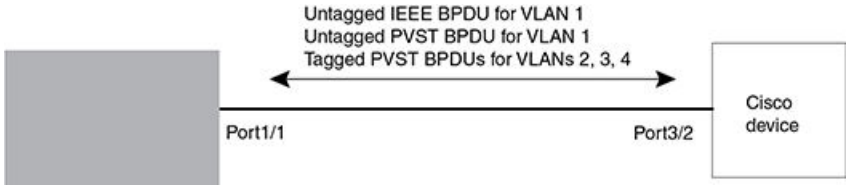
The examples use two common configurations:

- Untagged IEEE 802.1Q BPDUs on VLAN 1 and tagged PVST+ BPDUs on other VLANs
- Tagged IEEE 802.1Q BPDUs on VLAN 1 and untagged BPDUs on another VLAN

Tagged port using default VLAN 1 as its port native VLAN

In Figure 61, a PVST+ configuration uses VLAN 1 as the untagged default VLAN and VLANs 2, 3, and 4 as tagged VLANs.

FIGURE 60 Default VLAN 1 for untagged BPDUs



To implement this configuration, enter the following commands on the Extreme device.

```
device(config)# vlan-group 1 vlan 2 to 4
device(config-vlan-group-1)# tagged ethernet 1/1
device(config-vlan-group-1)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# pvst-mode
```

These commands configure a VLAN group containing VLANs 2, 3, and 4, add port 1/1 as a tagged port to the VLANs, and enable the dual-mode feature and PVST+ support on the port. The dual-mode feature allows the port to send and receive untagged frames for the default VLAN (VLAN 1 in this case) in addition to tagged frames for VLANs 2, 3, and 4. Enabling the PVST+ support ensures that the port is ready to send and receive PVST+ BPDUs. If you do not manually enable PVST+ support, the support is not enabled until the port receives a PVST+ BPDU.

The configuration leaves the default VLAN and the port's native VLAN unchanged. The default VLAN is 1 and the port's Port Native VLAN also is 1. The dual-mode feature supports untagged frames on the default VLAN only. Thus, port 1/1 can send and receive untagged BPDUs for VLAN 1 and can send and receive tagged BPDUs for the other VLANs.

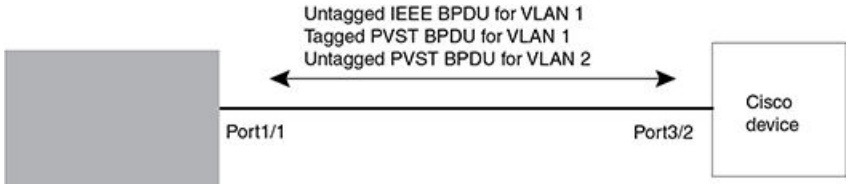
Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process tagged PVST BPDUs for VLANs 2, 3, and 4.
- Drop untagged PVST BPDUs for VLAN 1.

Untagged port using VLAN 2 as port native VLAN

In Figure 62, a port's Port Native VLAN is not VLAN 1. In this case, VLAN 1 uses tagged frames and VLAN 2 uses untagged frames.

FIGURE 61 Port Native VLAN 2 for untagged BPDUs



To implement this configuration, enter the following commands on the Extreme device.

```
device(config)# default-vlan-id 4000
device(config)# vlan 1
device(config-vlan-1)# tagged ethernet 1/1
device(config-vlan-1)# exit
device(config)# vlan 2
device(config-vlan-2)# untagged ethernet 1/1
device(config-vlan-2)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# pvst-mode
device(config-if-e10000-1/1)# exit
```

These commands change the default VLAN ID, configure port 1/1 as a tagged member of VLANs 1 and 2, and enable PVST+ support on port 1/1. Since VLAN 1 is tagged in this configuration, the default VLAN ID must be changed from VLAN 1 to another VLAN ID. Changing the default VLAN ID from 1 allows the port to process tagged frames for VLAN 1. VLAN 2 is the port native VLAN. The port processes untagged frames and untagged PVST BPDUs on VLAN 2.

Port 1/1 will process BPDUs as follows:

- Process IEEE 802.1Q BPDUs for VLAN 1.
- Process untagged PVST BPDUs for VLAN 2.
- Drop tagged PVST BPDUs for VLAN 1.

Note that when VLAN 1 is not the default VLAN, the ports must have an untagged VLAN enabled in order to process IEEE 802.1Q BPDUs.

For example, the following configuration is incorrect.

```
device(config)# default-vlan-id 1000
device(config)# vlan 1
device(config-vlan-1)# tagged ethernet 1/1 to 1/2
device(config-vlan-1)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# pvst-mode
device(config-if-e10000-1/1)# exit
device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# pvst-mode
device(config-if-e10000-1/2)# exit
```

In the configuration above, all PVST BPDUs associated with VLAN 1 would be discarded. Since IEEE BPDUs associated with VLAN 1 are untagged, they are discarded because the ports in VLAN 1 are tagged. Effectively, the BPDUs are never processed by the Spanning Tree Protocol. STP assumes that there is no better bridge on the network and sets the ports to FORWARDING. This could cause a Layer 2 loop.

The following configuration is correct.

```
device(config)# default-vlan-id 1000
device(config)# vlan 1
device(config-vlan-1)# tagged ethernet 1/1 to 1/2
device(config-vlan-1)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# pvst-mode
```

```
device(config-if-e10000-1/1)# exit
device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# pvst-mode
device(config-if-e10000-1/2)# exit
```

Setting the ports as dual-mode ensures that the untagged IEEE 802.1Q BPDUs reach the VLAN 1 instance.

802.1s Multiple Spanning Tree Protocol

Multiple Spanning Tree Protocol (MSTP) as defined in IEEE 802.1s allows you to configure multiple STP instances. This will allow several VLANs to be mapped to a reduced number of spanning-tree instances. This ensures loop-free topology for 1 or more VLANs that have the same Layer 2 topology.

NOTE
In addition to the features described in this chapter, Root Guard and BPDU Guard are supported. Refer to Root Guard on page 221 and BPDU Guard on page 223 for details.

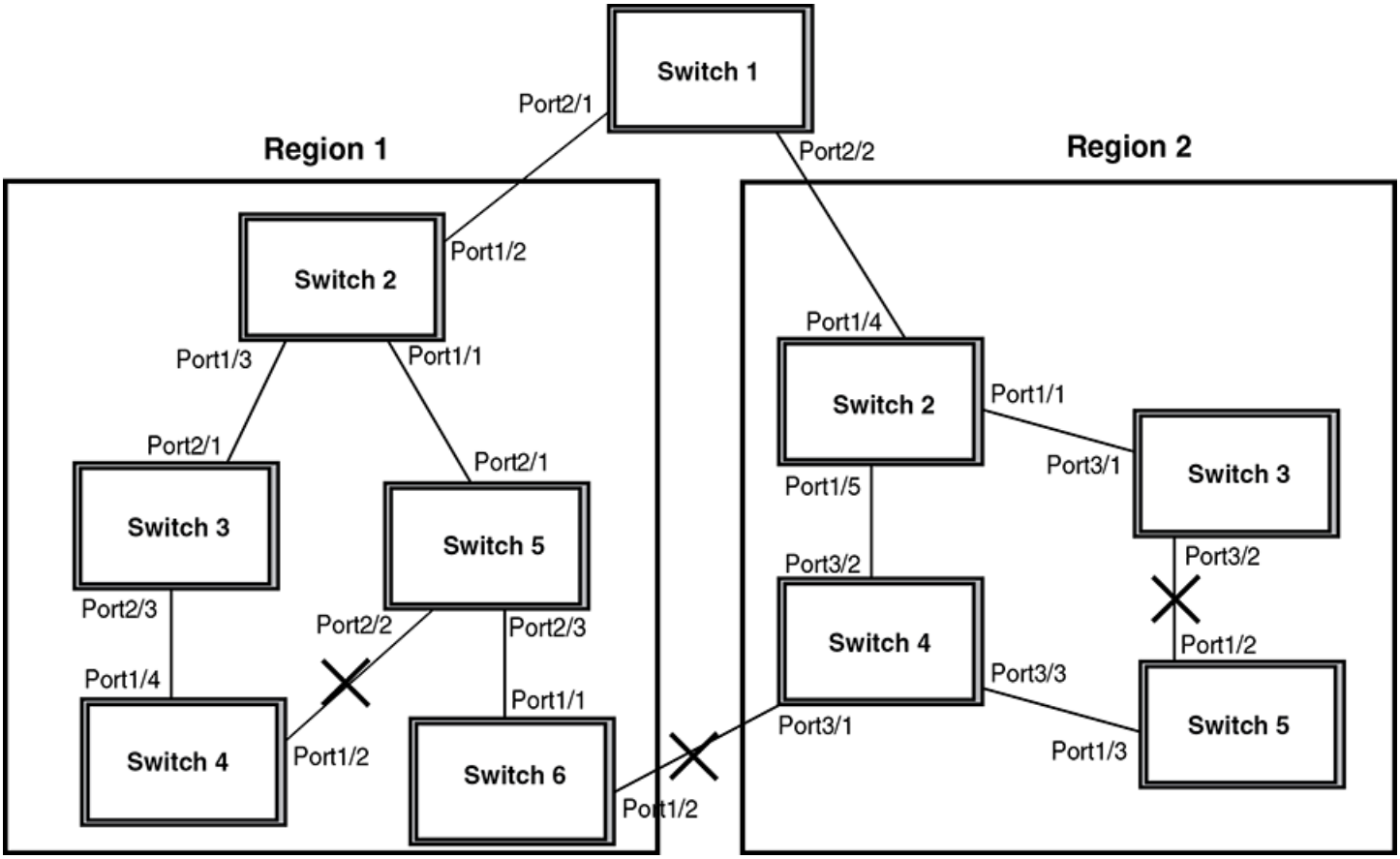
Multiple Spanning-Tree regions

Using MSTP, the entire network runs a common instance of RSTP. Within that common instance, one or more VLANs can be individually configured into distinct regions. The entire network runs the common spanning tree instance (CST) and the regions run a local instance. The local instance is known as Internal Spanning Tree (IST). The CST treats each instance of IST as a single bridge. Consequently, ports are blocked to prevent loops that might occur within an IST and also throughout the CST. In addition, MSTP can coexist with individual devices running STP or RSTP in the Common and Internal Spanning Trees instance (CIST). With the exception of the provisions for multiple instances, MSTP operates exactly like RSTP.

For example, in Figure 63 a network is configured with two regions: Region 1 and Region 2. The entire network is running an instance of CST. Each of the regions is running an instance of IST. In addition, this network contains Switch 1 running RSTP that is not configured in a region and, consequently, is running in the CIST instance. In this configuration, the regions are each regarded as a single bridge to the rest of the network, as is Switch 1. The CST prevents loops from occurring across the network. Consequently, a port is blocked at port 1/2 of Switch 4.

Additionally, loops must be prevented in each of the IST instances. Within the IST Region 1, a port is blocked at port 1/2 of Switch 4 to prevent a loop in that region. Within Region 2, a port is blocked at port 3/2 of Switch 3 to prevent a loop in that region.

FIGURE 62 MSTP configured network



The following definitions describe the STP instances that define an MSTP configuration:

Common Spanning (CST) - MSTP runs a single instance of spanning tree, called the Common Spanning Tree (CST), across all the bridges in a network. This instance treats each region as a single bridge. It all other ways, it operates exactly like Rapid Spanning Tree (RSTP).

Internal Spanning Tree (IST) - Instances of spanning tree that operate within a defined region are called ISTs (Internal Spanning Tree).

Common and Internal Spanning Trees (CIST) - This is the default MSTP instance 0. It contains all of the ISTs and all bridges that are not formally configured into a region. This instance interoperates with bridges running legacy STP and RSTP implementations.

Multiple Spanning Tree Instance (MSTI) - The MSTI is identified by an MST identifier (MSTid) value between 1 and 4094. This defines an individual instance of an IST. One or more VLANs can be assigned to an MSTI. A VLAN cannot be assigned to multiple MSTIs.

MSTP Region - These 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.

Configuring MSTP

To configure a device for MSTP for 1 or more VLANs that have the same Layer 2 topology, you could configure the name and the revision on each device that is being configured for MSTP. This name is unique to each device. 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 devices that interoperate with the same VLAN assignments. Port cost, priority and global parameters can then be configured for individual ports and instances. In addition, operational edge ports and point-to-point links can be created and MSTP can be disabled on individual ports.

MSTP can be configured on a device with MRP. However, they are mutually exclusive on a specific VLAN. Also, MSTP can be configured on a port that is part of a LAG following the same rules as used for STP and RSTP.

Each of the commands used to configure and operate MSTP are described in the following:

Setting the MSTP name

Each device that is running MSTP is configured with a name. It applies to the device which can have many different VLANs that can belong to many different MSTP regions. By default, the name is the MAC address of the device.

To configure an MSTP name, use a command such as the following at the Global Configuration level.

```
device(config)# mstp name mstp1
```

Syntax: [no] mstp name name

The *name* parameter defines an ASCII name for the MSTP configuration. The default name is the MAC address of the device expressed as a string.

Setting the MSTP revision number

Each device that is running MSTP is configured with a revision number. It applies to the device which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP revision number, use a command such as the following at the Global Configuration level.

```
device(config)# mstp revision 4
```

Syntax: [no] mstp revision revision-number

The *revision-number* parameter specifies the revision level for MSTP that you are configuring on the device. It can be a number from 0 and 65535.

Configuring an MSTP instance

An MSTP instance is configured with an MSTP ID for each region. Each region can contain one or more VLANs. To configure an MSTP instance and assign a range of VLANs, use a command such as the following at the Global Configuration level.

```
device(config) # mstp instance 7 vlan 4 to 7
```

Syntax: [no] mstp instance instance-number [**vlan vlan-id** | **vlan-group group-id**]

The **instance** parameter defines the number for the instance of MSTP that you are configuring. The maximum number of instances that can be configured is 16.

The **vlan** parameter assigns one or more VLANs or a range of VLANs to the instance defined in this command.

The **vlan-group** parameter assigns one or more VLAN groups to the instance defined in this command.

Configuring port priority and port path cost

Priority and path cost can be configured for a specified instance. To configure an MSTP instance, use a command such as the following at the Global Configuration level.

```
device(config)# mstp instance 7 ethernet 3/1 priority 32 path-cost 200
```

Syntax: [no] mstp instance instance-number ethernet slot/port priority port-priority path-cost cost

The *instance-number* variable is the number of the instance of MSTP that you are configuring priority and path cost for.

The **ethernet***slot/port* parameter specifies a port within a VLAN. The priority and path cost configured with this command will apply to VLAN that the port is contained within.

You can set a **priority** to the port that gives it forwarding preference over lower priority instances within a VLAN or on the device. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 240 in increments of 16. The default value is 128.

A **path-cost** can be assigned to a port to bias traffic towards or away from a path during periods of rerouting. Possible values are 1 - 200000000.

Configuring bridge priority for an MSTP instance

Priority can be configured for a specified instance. To configure priority for an MSTP instance, use a command such as the following at the Global Configuration level.

```
device(config)# mstp instance 1 priority 8192
```

Syntax: [no] mstp instance instance-number priority priority-value

The *instance-number* variable is the number for the instance of MSTP that you are configuring.

You can set a **priority** to the instance that gives it forwarding preference over lower priority instances within a VLAN or on the device. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 61440 in increments of 4096. The default value is 32768.

Setting the MSTP global parameters

MSTP has many of the options available in RSTP as well as some unique options. To configure MSTP Global parameters for all instances on a device.

```
device(config)# mstp force-version 0 forward-delay 10 hello-time 4 max-age 12 max-hops 9
```

Syntax: [no] mstp force-version mode-number forward-delay value hello-time value max-age value max-hops value

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following *mode-number* values:

- 0 - The STP compatibility mode. Only STP BPDUs will be sent. This is equivalent to single STP.
- 2 - The RSTP compatibility mode. Only RSTP BPDUS will be sent. This is equivalent to single STP.
- 3 - MSTP mode. In this default mode, only MSTP BPDUS will be sent.

The **forward-delay***value* specifies how long a port waits before it forwards an RST BPDU after a topology change. This can be a value from 4 - 30 seconds. The default is 15 seconds.

The **hello-time***value* parameter specifies the interval between two hello packets. The parameter can have a value from 1 - 10 seconds. The default is 2 seconds.

The **max-age***value* parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 - 40 seconds. The default value is 20 seconds.

The **max-hops***value* parameter specifies the maximum hop count. You can specify a value from 1 - 40 hops. The default value is 20 hops.

Setting ports to be operational edge ports

You can define specific ports as edge ports for the region in which they are configured to connect to devices (such as a host) that are not running STP, RSTP, or MSTP. If a port is connected to an end device such as a PC, the port can be configured as an edge port. To configure ports as operational edge ports enter a command such as the following.

```
device(config)# mstp admin-edge-port ethernet 3/1
```

Syntax: [no] mstp admin-edge-port ethernet slot/port

The *slot/port* parameter specifies a port or range of ports as edge ports in the instance they are configured in.

Setting point-to-point link

You can set a point-to-point link between ports to increase the speed of convergence. To create a point-to-point link between ports, use a command such as the following at the Global Configuration level.

```
device(config)# mstp admin-pt2pt-mac ethernet 2/5 ethernet 4/5
```

Syntax: [no] mstp admin-pt2pt-mac ethernet slot/port

The *slot/port* parameter specifies a port or range of ports to be configured for point-to-point links to increase the speed of convergence.

Disabling MSTP on a port

To disable MSTP on a specific port, use a command such as the following at the Global Configuration level.

```
device(config)# mstp disable 2/1
```

Syntax: [no] mstp disable slot/port

The *slot/port* variable specifies the location of the port that you want to disable MSTP for.

Forcing ports to transmit an MSTP BPDU

To force a port to transmit an MSTP BPDU, use a command such as the following at the Global Configuration level.

```
device(config)# mstp force-migration-check ethernet 3/1
```

Syntax: [no] mstp force-migration-check ethernet slot/port

The *slot/port* variable specifies the port or ports that you want to transmit an MSTP BPDU from.

Enabling MSTP on a device

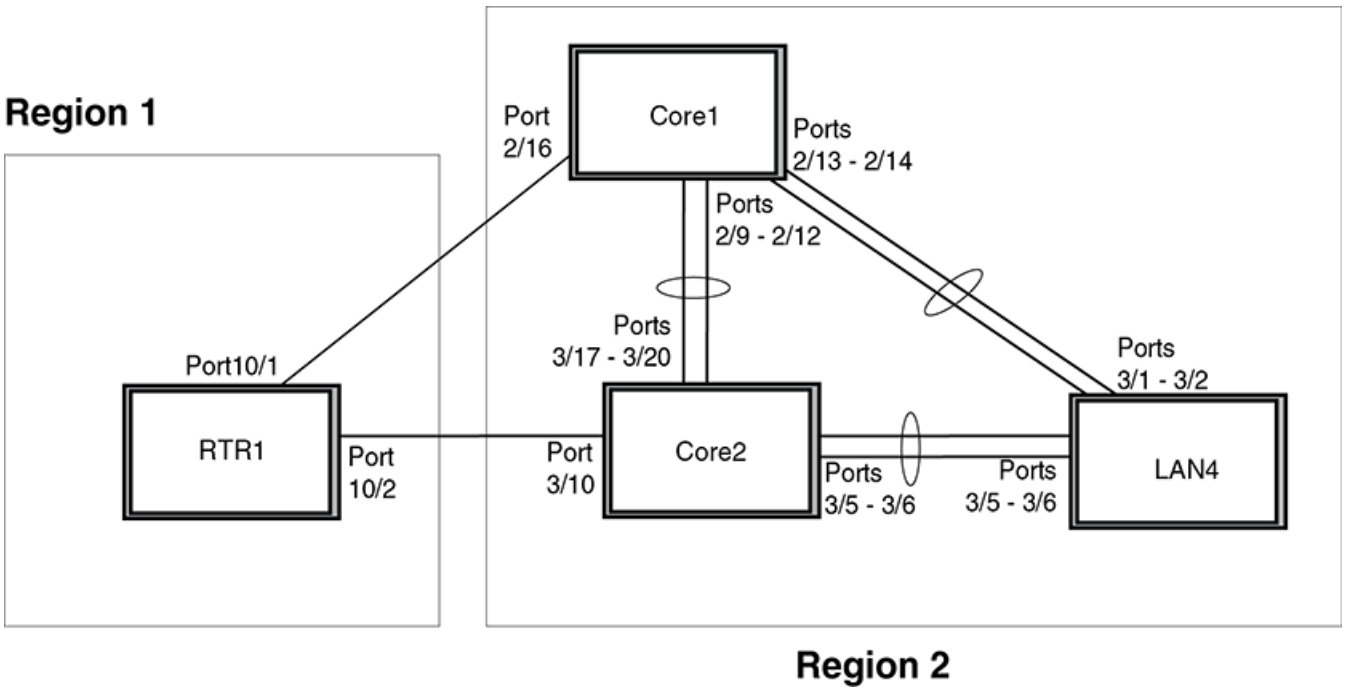
To enable MSTP on your device, use a command such as the following at the Global Configuration level.

```
device(config)# mstp start
```

Syntax: [no] start

In Figure 64 four Extreme devices are configured in two regions. There are four VLANs in four instances in Region 2. Region 1 is in the CIST.

FIGURE 63 SAMPLE MSTP configuration



RTR1 configuration

```
device(config-vlan-4093)tagged ethernet 10/1 to 10/2
device(config-vlan-4093)exit
device(config) mstp name Reg1
```

```
device(config) mstp revision 1
device(config) mstp instance 0 vlan 4093
device(config) mstp admin-pt2pt-mac ethernet 10/1 to 10/2
device(config) mstp start
device(config) hostname RTR1
```

Core 1 configuration

```
device(config-vlan-1) name DEFAULT-VLAN
device(config-vlan-1)no spanning-tree
device(config-vlan-1) exit
device(config) vlan 20
device(config-vlan-20) tagged ethernet 2/9 to 2/14 ethernet 2/16
device(config-vlan-20) no spanning-tree
device(config-vlan-20) exit
device(config) vlan 21
device(config-vlan-21) tagged ethernet 2/9 to 2/14 ethernet 2/16
device(config-vlan-21) no spanning-tree
device(config-vlan-21) exit
device(config) vlan 22
device(config-vlan-22) tagged ethernet 2/9 to 2/14 ethernet 2/16
device(config-vlan-22) no spanning-tree
device(config-vlan-22) exit
device(config) vlan 23
device(config) mstp name HR
device(config) mstp revision 2
device(config) mstp instance 20 vlan 20
device(config) mstp instance 21 vlan 21
device(config) mstp instance 22 vlan 22
device(config) mstp instance 0 priority 8192
device(config) mstp admin-pt2pt-mac ethernet 2/9 to 2/14
device(config) mstp admin-pt2pt-mac ethernet 2/16
device(config) mstp disable ethernet 2/240.
device(config) mstp start
device(config) hostname CORE1
```

Core2 configuration

```
device(config) vlan 1 name DEFAULT-VLAN
device(config-vlan-1) no spanning-tree
device(config-vlan-1) exit
device(config) vlan 20
device(config-vlan-20) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
device(config-vlan-20) no spanning-tree
device(config-vlan-20) exit
device(config) vlan 21
device(config-vlan-21) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
device(config-vlan-21) no spanning-tree
device(config-vlan-21) exit
device(config) vlan 22
device(config-vlan-22) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
device(config-vlan-22) no spanning-tree
device(config-vlan-22) exit
device(config) mstp name HR
device(config) mstp revision 2
device(config) mstp instance 20 vlan 20
device(config) mstp instance 21 vlan 21
device(config) mstp instance 22 vlan 22
device(config) mstp admin-pt2pt-mac ethernet 3/17 to 3/20 ethernet 3/5 to 3/6
device(config) mstp admin-pt2pt-mac ethernet 3/10
device(config) mstp disable ethernet 3/7 ethernet 3/24
device(config) mstp start
device(config) hostname CORE2
```

LAN 4 configuration

```
device(config) vlan 1 name DEFAULT-VLAN
device(config-vlan-1) no spanning-tree
device(config-vlan-1) exit
device(config) vlan 20
device(config-vlan-20) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
device(config-vlan-20) no spanning-tree
device(config) exit
device(config) vlan 21
device(config-vlan-21) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
device(config-vlan-21) no spanning-tree
device(config-vlan-21) exit
device(config) vlan 22
device(config-vlan-22) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
device(config-vlan-22) no spanning-tree
device(config) mstp config name HR
device(config) mstp revision 2
device(config) mstp instance 20 vlan 20
device(config) mstp instance 21 vlan 21
device(config) mstp instance 22 vlan 22
device(config) mstp admin-pt2pt-mac ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
device(config) mstp start
device(config) hostname LAN4
```

Displaying MSTP statistics

MSTP statistics can be displayed using the commands shown below.

To display all general MSTP information, enter the following command.

```
device(config)#show mstp
MSTP Instance 0 (CIST) - VLANs: 1
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876
MSTP Instance 1 - VLANs: 2
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20
Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                          ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

To display all general MSTP information for blocked ports only, enter the following command

```
device# show mstp blocked
MSTP Instance 0 (CIST) - VLAN Scope: None
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
80000024389e2d00 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
80000024388f6b00 0         80000024388f6b00 2000      80000024388f6b00 3/1
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
```

```
Num       Cost      Mac Port                        ted cost  bridge
3/2   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/3   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/4   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/5   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/6   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/7   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/8   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
MSTP Instance 1 - VLANs: 10
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
80010024389e2d00 20  80010024388f6b00 2000      80010024388f6b00 3/1   19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
3/2   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
3/3   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
3/4   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
MSTP Instance 2 - VLANs: 20
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
80020024389e2d00 20  80020024388f6b00 2000      80020024388f6b00 3/5   19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
3/6   128 2000      F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
3/7   128 2000      F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
3/8   128 2000      F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
```

The following example displays MSTP information for a specified MSTP instance.

```
device(config)# show mstp 1

MSTP Instance 1 - VLANs: 2
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20
Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                         ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

The following example displays blocked ports only, for a specified MSTP instance

```
device# show mstp blocked 1
MSTP Instance 1 - VLANs: 10
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
80010024389e2d00 20  80010024388f6b00 2000      80010024388f6b00 3/1   19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
3/2   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
3/3   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
```

3/4 128 2000 F F ALTERNATE DISCARDING 0 80010024388f6b00

.

Refer to Table 39 for details about the display parameters.

Syntax: show mstp [**blocked**] [**mstp-id**]

The **blocked** parameter displays information for blocked ports only. When the blocked parameter is not specified, information is displayed for all ports.

The *mstp-id* variable specifies the MSTP instance for which you want to display information.

TABLE 39 Output from show MSTP

This field...	Displays...
MSTP Instance	The ID of the MSTP instance whose statistics are being displayed. For the CIST, this number is 0.
VLANs:	The number of VLANs that are included in this instance of MSTP. For the CIST this number will always be 1.
Bridge Identifier	The MAC address of the bridge.
Bridge MaxAge sec	Displays configured Max Age.
Bridge Hello sec	Displays configured Hello variable.
Bridge FwdDly sec	Displays configured FwdDly variable.
Bridge Hop cnt	Displays configured Max Hop count variable.
Root MaxAge sec	Max Age configured on the root bridge.
Root Hello sec	Hello interval configured on the root bridge.
Root FwdDly sec	FwdDly interval configured on the root bridge.
Root Hop Cnt	Current hop count from the root bridge.
Root Bridge	Bridge identifier of the root bridge.
ExtPath Cost	The configured path cost on a link connected to this port to an external MSTP region.
Regional Root Bridge	The Regional Root Bridge is the MAC address of the Root Bridge for the local region.
IntPath Cost	The configured path cost on a link connected to this port within the internal MSTP region.
Designated Bridge	The MAC address of the bridge that sent the best BPDU that was received on this port.
Root Port	Port indicating shortest path to root. Set to "Root" if this bridge is the root bridge.
Port Num	The port number of the interface.
Pri	The configured priority of the port. The default is 128.
PortPath Cost	Configured or auto detected path cost for port.
P2P Mac	Indicates if the port is configured with a point-to-point link: • **T** - The port is configured in a point-to-point link • **F** - The port is not configured in a point-to-point link
Edge	Indicates if the port is configured as an operational edge port: • **T** - indicates that the port is defined as an edge port. • **F** - indicates that the port is not defined as an edge port
Role	The current role of the port: • Master • Root • Designated • Alternate • Backup • Disabled
State	The port's current 802.1w state. A port can have one of the following states: • Forwarding

TABLE 39 Output from show MSTP (continued)

This field...	Displays...
	• Discarding • Learning • Disabled
Designated Cost	Port path cost to the root bridge.
Max Hop cnt	The maximum hop count configured for this instance.
Root Hop cnt	Hop count from the root bridge.

Displaying MSTP information for CIST instance 0

Instance 0 is the Common and Internal Spanning Tree Instance (CIST). When you display information for this instance there are some differences with displaying other instances. The following example displays MSTP information for CIST Instance 0.

```
device(config)#show mstp 0
MSTP Instance 0 (CIST) - VLANs: 1
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876
```

Refer to Displaying MSTP statistics on page 257 for explanation about the parameters in the output.

To display MSTP configuration information, enter the following command.

```
device(config)# show mstp configuration
mstp name test
mstp revision 1
mstp instance 1  vlan 100
mstp admin-pt2pt-mac ethe 4/7 to 4/8
mstp start
```

To display details about the MSTP that is configured on the device, enter the following command.

```
device(config)# show mstp detail
MSTP Instance 0 (CIST) - VLAN Scope: None
--------------------------------------------------------------------------------
  Bridge: 8000002438a5a800 [Priority 32768, SysId 0, Mac 002438a5a800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 4/7 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128, OperEdge F, OperPt2PtMac T, Boundary F
  Designated   - Root 8000002438a5a800, RegionalRoot 8000002438a5a800,
                 Bridge 8000002438a5a800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2 recover timer 0
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 5, RST 0, Config 0, TCN 0
                 Sent MST 186, RST 0, Config 0, TCN 0
Port 4/8 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128, OperEdge F, OperPt2PtMac T, Boundary F
  Designated   - Root 8000002438a5a800, RegionalRoot 8000002438a5a800,
                 Bridge 8000002438a5a800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2 recover timer 0
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

```
  BPDUs        - Rcvd MST 17, RST 0, Config 0, TCN 0
                 Sent MST 173, RST 0, Config 0, TCN 0
MSTP Instance 1 - VLANs: 100
------------------------------------------------------------------------------
  Bridge: 8001002438a5a800 [Priority 32768, SysId 1, Mac 002438a5a800]
Port 4/7 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128
  Designated   - RegionalRoot 8001002438a5a800, IntCost 0
                 Bridge 8001002438a5a800
  ActiveTimers - recover timer 0
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 4/8 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128
  Designated   - RegionalRoot 8001002438a5a800, IntCost 0
                 Bridge 8001002438a5a800
  ActiveTimers - recover timer 0
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp [**mstp-id** | **configuration** | **detail**] [**begin string** | **exclude string** | **include string**]

The *mstp-id* variable specifies the MSTP instance for which you want to display information.

The **begin***string* parameter specifies the display of information from the first line containing the "string". Information prior to the first occurence of the "string" will not be displayed.

The **exclude***string* parameter specifies the exclusion of lines containing the "string". All other information will be displayed.

The **include***string* parameter specifies the display of information containing the "string" only. All other information will be filtered out.

Interoperability between MSTP and Single STP or Single RSTP

- MSTP can interoperate with SSTP or a RSTP. However it is recommended to assign all VLANs to CIST (MSTP instance 0) while operating in a SSTP or a RSTP domain since only CIST participates in the convergence with SSTP or a RSTP.
- If the STP or RSTP is enabled on a native VLAN or on an untagged VLAN (or if SSTP or a single RSTP is configured), it sends out untagged BPDUs and can interoperate with MSTP which consumes these untagged BPDUs and converges accordingly. These untagged BPDUs can also converge with SSTP or RSTP configured on the receiving side resulting in the MSTP not being configured.
- If STP is enabled on a native VLAN or an untagged VLAN or if SSTP or RSTP is configured, it will converge with STP enabled on native or an untagged VLAN if configured, or with RSTP enabled on native or untagged VLAN if configured, or otherwise with SSTP/Single RSTP/MSTP if configured on the receiving side.

MSTP support for PBB

When applied to a PBB environment, this feature will ensure a loop free topology.

Scalability

- A maximum of 16 MSTP Instances are allowed per MSTP region.
- A maximum of 10 MSTP regions are allowed.
- A maximum of 40 MSTP Instances are allowed.

Limitations

The following restrictions apply when are using MSTP with PBB

- Dual homing of an AS configured with SVLANs in an Edge RSTP and Dual homing of a CS configured with regular VLANs in a Core MSTP is supported, but Dual homing of both AS and CS in an Edge and Core MSTP together is not supported.
- Each switch in a single MSTP region needs to be configured with unique name, but must use the same region name and revision number.
- For each MSTP region, the configured VLANs for an MSTP instance need to be of the same type, for example, either VLANs (that is, not VPLS VLANs), SVLANs, or CVLANs. This means VLANS, SVLANs, and CVLANs cannot be configured to co-exist within the same region.
- In the case of an VLAN, SVLAN, or CVLAN, the same VLAN ID cannot be configured as a part of two different MSTP instances within an MSTP region.
- MSTP configuration for VPLS VLANs with ISIDs is not supported.
- An interface can be a part of only one region when multiple regions are configured on a switch.
- Legacy and multi region MSTP configuration is not allowed at the same time on a switch.
- VPLS instances having two different VLANs is not supported.
- There is no MIB support for PBB MSTP.
- MSTP should not be configured for (1) topology groups having layer 2 (L2) member VLANs (2) member VLANs configured in a topology group. If a topology group is configured with a master vlan running MSTP, layer 2 (L2) VLANs should not be configured as members until MSTP is disabled on the master VLAN of this topology group. Such configurations via CLI are blocked.

Use case scenario

The figure below displays the use case. In this scenario, the network has two Core Switches (CSs) to provide resiliency in the core as well as service load sharing.

The CS functions as a Backbone Core Bridge (BCB). Every AS (Access Switch) in the network will dual home to the two CSs. The AS functions as a Backbone Edge Bridge (BEB). The function of the CS is to switch traffic between ASs. As a BCB, a CS will switch on the outer PBB B-tag and will not perform any PBB encapsulation.

The AS accepts 802.1q, or Q-in-Q traffic, from ESs and encapsulates the traffic into PBB frames. The SVLAN of the customer frame is the service delimiter and is mapped to a specific ISID in the PBB network. The ASs are logically interconnected with PBB tunnels (BVLANs), which always traverse a CS. The ASs will connect to the ESs as shown in the figure below.

The AS and CS switches in a network will form a single MSTP region. With the dual homing of the ASs to the CSs, all failures are protected against.

Each ES will attach to an AS and will form its own MSTP domain with the AS as the root. ES traffic that is destined for a port on the same ES does not need to enter the network core. The traffic will stay intra-switch.

Traffic that is destined for a port on another ES attached to the same AS will switch directly on the AS and will not have to enter the network core. The traffic will stay within the AS and attached ES.

Traffic that is destined for a port on an ES that is attached to a different AS will be encapsulated into PBB and will traverse the PBB core.

Figure to be added here for Ethernet Switch Architecture Use Case

Figure to be added here for AS, CS, and ES Physical Connections

Edge MSTP in a PB network

The following deployment scenario is a case where MSTP is deployed for a single S-VLAN in a PB network. PBB traffic uses S-VLAN 200.

The procedure to configure the nodes in the topology are discussed below.

Figure to be added here for Edge MSTP in a PB network.

High availability

MSTP supports MP switchover and hitless software upgrade. When an MSTP root bridge undergoes MP switchover and hitless upgrade, there will be no break in transmission of the MSTP BPDU from it during reboot of the line cards. Due to this, there will be no re-convergence of the topology and no disruption in traffic.

MSTP PBB with multi region feature also supports MP switchover and hitless software upgrade. There will be no traffic disruption during an upgrade.

MSTP PBB Configuration Commands

MSTP-region

To configure multiple MSTP regions on a single bridge to represent different bridging domains, use the **MSTP-region** command. This command is available only in the configuration mode.

All the existing MSTP commands can be executed from this mode.

```
device(config)#mstp-region 1
device(config-mstp-region-1)# mstp-region ?
```

Syntax: [no] mstp-region r *egion-id*

The acceptable range for this command is 1 to 10.

Executing the **no mstp-region** command will delete all the configurations that are configured under the **region** submode.

MSTP Instance Mapping

To configure Regular VLANs or VPLS VLANs mapped to an MSTP instance in a PB or PBB network, use the following command options.

Syntax: [no] mstp-region instance *instance-id* **vlan** *vlan-id*

Syntax: [no] mstp-region instance *instance-id* **vlan** *vlan-id* **to** *vlan-id*

Syntax: [no] mstp-region instance *instance-id* **vlan-group** *group-id*

For MLX Series and XMR Series devices

On the MLX Series and XMR Series, use the following commands to configure VPLS VLANs mapped to an MSTP instance.

Syntax: [no] mstp-region instance *instance-id* **vpls** *vpls-id* **vlan** *vlan-id*

Syntax: [no] mstp-region instance *instance-id* **vpls** *vpls-id* **vlan** *vlan-id* **to** *vlan-id*

For CER 2000 Series and CES 2000 Series devices

On the CER 2000 Series and CES 2000 Series, use the following command to configure VPLS VLANs mapped to an MSTP instance.

Syntax: [no] mstp-region instance *instance-id* **esi** *esi-name* **vlan** *vlan-id*

Each variable has the following range.

- Instance ID: 0-4094 (0 for CIST and 1-4094 for MSTI)
- Path Cost: 1-200000000
- Port Priority: 0-240 in the increments of 16
- Instance Priority Value: 0-61440 in the increments of 4096
- VLAN ID: 1-4090
- VPLS ID: 1-4294967294

Executing the **no mstp-region instance** command deletes the configured MST instance to VLAN mapping.

Configuring the MLX Series and XMR Series devices

The following procedure describes how to configure AS-1, AS-2, and ES-1 in the scenario shown in the figure above.

Configuring AS-1

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/2
```

MSTP Configuration

```
device_AS-1(config)#mstp-region 2
device_AS-1(config-mstp-region-2)#mstp-region name PB-Domain1
device_AS-1(config-mstp-region-2)#mstp-region rev 1
device_AS-1(config-mstp-region-2)#mstp-region instance 1 vpls 1 vlan 200
device_AS-1(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_AS-1(config-mstp-region-2)#mstp-region start
```

Configuring AS-2

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/3
```

MSTP Configuration

```
device_AS-2(config)#mstp-region 2
device_AS-2(config-mstp-region-2)#mstp-region name PB-Domain1
device_AS-2(config-mstp-region-2)#mstp-region rev 1
device_AS-2(config-mstp-region-2)#mstp-region instance 1 vpls 1 vlan 200
device_AS-2(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/3
device_AS-2(config-mstp-region-2)#mstp-region start
```

Configuring ES-1

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration:

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-svlan 1
device_ES-1(config-mpls-vpls-pb-svlan)#pbb
device_ES-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 300 on customer port.

```
device_ES-1(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-svlan-vlan-300)#tag eth 1/1 eth 1/4
```

MSTP Configuration

```
device_ES-1(config)#mstp-region 2
device_ES-1(config-mstp-region-2)#mstp-region name PB-Domain1
device_ES-1(config-mstp-region-2)#mstp-region rev 1
device_ES-1(config-mstp-region-2)#mstp-region instance 1 vpls 1 vlan 200
device_ES-1(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/2 to 1/3
device_ES-1(config-mstp-region-2)#mstp-region start
```

```
device_ES-1(config)#mstp-region 100
device_ES-1(config-mstp-region-100)#mstp-region name Cust-Domain
device_ES-1(config-mstp-region-100)#mstp-region rev 1
device_ES-1(config-mstp-region-100)#mstp-region instance 1 vlan 300
device_ES-1(config-mstp-region-100)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/4
device_ES-1(config-mstp-region-100)#mstp-region start
```

Configuring CE-1 and CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 300 (C-VLAN) and add port 1/1(CE-1) and 1/4 (CE-2) to it.

C-VLAN Configuration

```
device_CE-1(config)#vlan 300
device_CE-1(config-vlan-300)#tagged ethernet 1/1
```

MSTP Configuration

```
device_CE-1(config)#mstp name Cust-Domain
device_CE-1(config)#mstp rev 1
device_CE-1(config)#mstp instance 1 vlan 300
device-1(config)#mstp admin-pt2pt-mac ethernet 1/1
device_CE-1(config)#mstp start
```

C-VLAN Configuration

```
device_CE-2(config)#vlan 300
device_CE-2(config-vlan-300)#tagged ethernet 1/4
```

MSTP Configuration

```
device_CE-2(config)#mstp name Cust-Domain
device_CE-2(config)#mstp rev 1
device_CE-2(config)#mstp instance 1 vlan 300
device_CE-2(config)#mstp admin-pt2pt-mac ethernet 1/4
device_CE-2(config)#mstp start
```

CES/CER configuration

Configuring AS1

```
device_AS1(config)#int e 1/1
device_AS1(config-if-e1000-1/1)#port-type backbone-edge
device_AS1(config-if-e1000-1/1)#int e 1/2
device_AS1(config-if-e1000-1/2)#port-type backbone-edge
device_AS1(config-if-e1000-1/2)#esi svlan1 encap svlan
device_AS1(config-esi-svlan1)#vlan 200
device_AS1(config-esi-svlan1-vlan-200)#tag e 1/1 e 1/2
```

Configuring AS2

```
device_AS2(config)#int e 1/1
device_AS2(config-if-e1000-1/1)#port-type backbone-edge
device_AS2(config-if-e1000-1/1)#int e 1/3
```

```
device_AS2(config-if-e1000-1/3)#port-type backbone-edge
device_AS2(config-if-e1000-1/3)#esi svlan1 encap svlan
device_AS2(config-esi-svlan1)#vlan 200
device_AS2(config-esi-svlan1-vlan-200)#tag e 1/1 e 1/3
```

MSTP configuration for AS1 and AS2

```
device_AS2(config)#mstp-region 2
device_AS2(config-mstp-region-2)#mstp-region name PB-Domain1
device_AS2(config-mstp-region-2)#mstp-region rev 1
device_AS2(config-mstp-region-2)#mstp-region instance 1 esi svlan1 vlan 200
device_AS2(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/3
device_AS2(config-mstp-region-2)#mstp-region start
```

Configuring ES1

```
device_ES1(config)#int e 1/1
device_ES1(config-if-e1000-1/1)#port-type customer-edge
device_ES1(config-if-e1000-1/1)#int e 1/4
device_ES1(config-if-e1000-1/4)#port-type customer-edge
device_ES1(config-if-e1000-1/4)#esi cvlan1 encap cvlan
device_ES1(config-esi-cvlan1)#vlan 300
device_ES1(config-esi-cvlan1-vlan-300)#tag e 1/1 e 1/4
device_ES1(config-esi-cvlan1-vlan-300)#int e 1/2
device_ES1(config-if-e1000-1/2)#port-type provider-network
device_ES1(config-if-e1000-1/2)#int e 1/3
device_ES1(config-if-e1000-1/3)#port-type provider-network
device_ES1(config-if-e1000-1/3)#esi svlan2 encap svlan
device_ES1(config-esi-svlan2)#esi-client cvlan1
device_ES1(config-esi-svlan2)#vlan 200
device_ES1(config-esi-svlan2-vlan-200)#tag e 1/2 e 1/3
```

MSTP configuration for ES1

```
device_ES1(config)#mstp-region 2
device_ES1(config-mstp-region-2)#mstp-region name PB-Domain1
device_ES1(config-mstp-region-2)#mstp-region rev 1
device_ES1(config-mstp-region-2)#mstp-region instance 1 esi svlan2 vlan 200
device_ES1(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/2 to 1/3
device_ES1(config-mstp-region-2)#mstp-region start
device_ES1(config)#mstp-region 100
device_ES1(config-mstp-region-100)#mstp-region name Cust-Domain
device_ES1(config-mstp-region-100)#mstp-region rev 1
device_ES1(config-mstp-region-100)#mstp-region instance 1 vlan 300
device_ES1(config-mstp-region-100)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/4
device_ES1(config-mstp-region-100)#mstp-region start
```

Configuring CE1

```
device_CE1(config)#vlan 300
device_CE1(config-vlan-300)#tag e 1/1
```

MSTP configuration for CE1

```
device_CE1(config)#mstp name Cust-Domain
device_CE1(config)#mstp rev 1
device_CE1(config)#mstp instance 1 vlan 300
device_CE1(config)#mstp admin-pt2pt-mac ethernet 1/1
device_CE1(config)#mstp start
```

Configuring CE2

```
device_CE2(config)#vlan 300
device_CE2(config-vlan-300)#tag e 1/4
```

MSTP configuration for CE2

```
device_CE2(config)#mstp name Cust-Domain
device_CE2(config)#mstp rev 1
device_CE2(config)#mstp instance 1 vlan 300
device_CE-2(config)#mstp admin-pt2pt-mac ethernet 1/4
device_CE2(config)#mstp start
```

Configuring MSTP in a PBB network

The following deployment scenario is a case where MSTP is deployed for a single B-VLAN in a PBB network. PBB traffic uses B-VLAN 100. The procedure to configure the nodes in the topology are discussed below.

Figure to be added here for Core MSTP in a PBB network.

XMR Series and MLX Series configuration

AS-1 Configuration

> NOTE
> The configuration of AS-2 is similar to the AS-1 configuration.

For PBB traffic you will configure a VPLS instance, the B-VLAN used here is 100. For PB traffic, the S-VLAN used is 200 and C-VLAN 300. Here, you need topology groups to configure the BVLAN as the master VLAN and VPLS VLANs with different ISIDs mapping to the same BVLAN as member VLANs. Then the MSTP instance is configured to be mapped to the master VLAN of the topology group (the BVLAN).

Tag type configuration

```
device_AS-1(config)#tag-type 88e8 eth 1/1
device_AS-1(config)#tag-type 88e8 eth 1/2
device_AS-1(config)#tag-type 9100 eth 1/10
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 100
device_AS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan 1
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb) #vlan 100 isid 101010
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan1 2
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb) #vlan 100 isid 10101
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan2 3
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
```

```
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb) #vlan 100 isid 1010
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-bvlan)#vlan 200
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-200) #tagged ethernet 1/10
```

Topology Group Configuration

```
device_AS-1(config)#topology-group 1
device_AS-1(config-topo-group-1)#master-vlan 100
device_AS-1(config-topo-group-1)#member-vlan vpls name bvlan vlan 100 isid 101010
device_AS-1(config-topo-group-1)#member-vlan vpls name bvlan1 vlan 100 isid 10101
device_AS-1(config-topo-group-1)#member-vlan vpls name bvlan2 vlan 100 isid 1010
```

MSTP Configuration

```
device_AS-1(config)#mstp-region 1
device_AS-1(config-mstp-region-1)#mstp-region name PBB-Domain
device_AS-1(config-mstp-region-1)#mstp-region rev 1
device_AS-1(config-mstp-region-1)#mstp-region instance 1 vlan 100
device_AS-1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_AS-1(config-mstp-region-1)#mstp-region start
```

CS-1 Configuration

NOTE
The CS-2 configuration is similar to the CS-1 configuration.

Port type configuration

```
device_CS-1(config)#tag-type 88e8 eth 1/1
device_CS-1(config)#tag-type 88e8 eth 1/2
```

B-VLAN Configuration:

```
device_CS-1(config)#vlan 100
device_CS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

MSTP Configuration

```
device_CS-1(config)#mstp-region 1
device_CS-1(config-mstp-region-1)#mstp-region name PBB-Domain
device_CS-1(config-mstp-region-1)#mstp-region rev 1
device_CS-1(config-mstp-region-1)#mstp-region instance 1 vlan 100
device_CS-1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_CS-1(config-mstp-region-1)#mstp-region start
```

Configuring the CER 2000 Series and CES 2000 Series devices

Configuring AS1

NOTE
The AS-2 configuration is similar to the AS-1 configuration.

```
device_AS1(config)#int e 1/1
device_AS1(config-if-e1000-1/1)#port-type backbone-network
device_AS1(config-if-e1000-1/1)#int e 1/2
device_AS1(config-if-e1000-1/2)#port-type backbone-network
device_AS1(config-if-e1000-1/2)#int e 1/10
device_AS1(config-if-e1000-1/10)#port-type backbone-edge
device_AS1(config-if-e1000-1/10)#esi svlan2 encap svlan
device_AS1(config-esi-svlan2)#vlan 200
device_AS1(config-esi-svlan2-vlan-200)#tag e 1/10
device_AS1(config-esi-svlan2-vlan-200)#esi isid1 encap isid
device_AS1(config-esi-isid1)#isid 101010
device_AS1(config-esi-isid1-isid-101010)#esi-client svlan2
device_AS1(config-esi-isid1-isid-101010)#esi pbb-bvlan encap bvlan
device_AS1(config-esi-pbb-bvlan)#esi-client isid1
device_AS1(config-esi-pbb-bvlan)#vlan 100
device_AS1(config-esi-pbb-bvlan-vlan-100)#tag e 1/1 e 1/2
```

MSTP configuration for AS1

```
device_AS1(config)#mstp-region 1
device_AS1(config-mstp-region-1)#mstp name PBB-Domain
device_AS1(config-mstp-region-1)#mstp rev 1
device_AS1(config-mstp-region-1)#mstp instance 1 esi pbb-bvlan vlan 100
device_AS1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_AS1(config-mstp-region-1)#mstp start
```

Configuring CSs

```
device_CS1(config)#int e 1/1
device_CS1(config-if-e1000-1/1)#port-type backbone-network
device_CS1(config-if-e1000-1/1)#int e 1/2
device_CS1(config-if-e1000-1/2)#port-type backbone-network
device_CS1(config-vlan-100)#esi pbb-bvlan encap bvlan
device_CS1(config-esi-pbb-bvlan)#vlan 100
device_CS1(config-esi-pbb-bvlan-vlan-100)#tag e 1/1 e 1/2
```

MSTP configuration:

```
device_CS1(config)#mstp-region 1
device_CS1(config-mstp-region-1)#mstp-region name PBB-Domain
device_CS1(config-mstp-region-1)#mstp-region rev 1
device_CS1(config-mstp-region-1)#mstp-region instance 1 esi pbb-bvlan vlan 100
device_CS1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_CS1(config-mstp-region-1)#mstp-region start
```

Show commands

The output of the following commands include the information about the configured Layer 2 VLANs, VPLS VLANs (MLX Series and XMR Series) or ESI VLANs (CER 2000 Series and CES 2000 Series) within MSTP.

Show MSTP config

The **show mstp config** command displays the MSTP configuration as it appears in the running config.

For MLX Series and XMR Series devices

```
device_DUT# show mstp config
Mstp-region 1
Mstp-region name PBB-Domain
Mstp-region revision 1
Mstp-region instance 1 vpls 1 vlan 100
Mstp-region start
Mstp-region 2
Mstp-region name PB-Domain1
Mstp-region revision 1
Mstp-region instance 1 vpls 1 vlan 200
Mstp-region start
```

Syntax: show mstp config

For CER 2000 Series and CES 2000 Series devices

```
device_DUT# show mstp config
Mstp-region 1
Mstp-region name PBB-Domain
Mstp-region revision 1
Mstp-region instance 1 esi pbb-bvlan vlan 100
Mstp-region start
Mstp-region 2
Mstp-region name PB-Domain1
Mstp-region revision 1
Mstp-region instance 1 esi pb-svlan vlan 200
Mstp-region start
```

Syntax: show mstp config

Show MSTP

The **show mstp** command displays information about all the MSTP regions for all configured instances.

For MLX Series and XMR Series devices

The following example displays information about all the MSTP regions for all configured instances.

```
device# show mstp
Region 1:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
```

```
Num       Cost      Mac Port                        ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
Region 2:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
-------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 200
-------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example displays information about blocked ports only, for all MSTP regions and all configured instances in a VPLS VLAN.

```
device# sh mstp blocked
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
-------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000748ef82ba800 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
80000024388f6b00 0         80000024388f6b00 20000     80000024388f6b00 1/13
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
MSTP Instance 1 - VPLS VLANs: 200 to 201
-------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001748ef82ba800 20  80010024388f6b00 20000     80010024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
MSTP Instance 2 - VPLS VLANs: 300
-------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8002748ef82ba800 20  80020024388f6b00 20000     80020024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
```

1/14 128 20000 F F ALTERNATE DISCARDING 0 80020024388f6b00

Syntax: show mstp [**blocked**]

For CER 2000 Series and CES 2000 Series devices

```
device# show mstp
Region 1:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
--------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
--------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
Region 2:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
--------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - ESI VLANs: esi pb-svlan - Encapsulation svlan - vlan 200
--------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example displays MSTP information for blocked ports only, in an ESI VLAN configuration.

```
device# show mstp blocked

MSTP Instance 0 (CIST) - ESI VLAN Scope: None
--------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedb59a40 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
```

```
8000001bedb4e740 20000     8000001bedb59a40 0         8000001bedb4e740 1/17
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 0         8000001bedb4e740
MSTP Instance 1 - ESI VLANs: 10 to 11
------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedb59a40 20  8001001bedb59a40 0         8001001bedb59a40 Root  20
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 0         8001001bedb59a40
```

Syntax: show mstp [**blocked**]

Show mstp region

The **show mstp region** command displays similar output as the **show mstp** command, filtered for the queried region ID.

For MLX Series and XMR Series devices

The following example displays MSTP information for region "1".

```
device# show mstp region 1
Region 1:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example displays MSTP information for blocked ports only, in region "1" in a VPLS VLAN configuration.

```
device#show mstp blocked region 1
Region 1
--------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000748ef82ba800 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
80000024388f6b00 0         80000024388f6b00 20000     80000024388f6b00 1/13
```

```
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
MSTP Instance 1 - VPLS VLANs: 100
------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001748ef82ba800 20  80010024388f6b00 20000     80010024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
MSTP Instance 2 - VPLS VLANs: 200
------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8002748ef82ba800 20  80020024388f6b00 20000     80020024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
```

Syntax: show mstp [**blocked**] **region** *region-id*

For CER 2000 Series and CES 2000 Series devices

```
device# show mstp region 1
Region 1:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example shows MSTP information for blocked ports only, filtered for region "1" in an ESI VLAN configuration.

```
device#show mstp blocked region 1
Region 1
--------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedb59a40 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
```

```
8000001bedb4e740 0         8000001bedb4e740 20000     8000001bedb4e740 1/17
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 0         8000001bedb4e740
MSTP Instance 1 - ESI VLANs: 10
------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedb59a40 20  8001001bedb4e740 20000     8001001bedb4e740 1/17  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 20000     8001001bedb59a40
MSTP Instance 2 - ESI VLANs: 11
------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8002001bedb59a40 20  8002001bedb4e740 20000     8002001bedb4e740 1/18  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 20000     8002001bedb59a40
```

Syntax: show mstp [**blocked**] **region** *region-id*

Show MSTP detail

The **show mstp detail** command displays information about all of the MSTP regions for all configured instances in detail.

For MLX Series and XMR Series devices

```
device_DUT# show mstp detail
Region 1:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
```

```
                Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Region 2:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
--------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
--------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail

For CER 2000 Series and CES 2000 Series devices

```
device_DUT# show mstp detail
Region 1:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
--------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

```
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
------------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Region 2:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - ESI VLANs: esi pb-svlan - Encapsulation svlan - vlan 200
------------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail

Show MSTP detail region

The **show mst detail region** command output is similar to the **show mstp detail** command, but is filtered for the queried region ID.

For MLX Series and XMR Series devices

```
device_DUT# show mstp detail region 1
Region 1:
-----------
```

```
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail region *region-id*

For CER 2000 Series and CES 2000 Series devices

```
device_DUT# show mstp detail region 1
Region 1:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
```

```
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail region *region-id*

Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol overview

This chapter explains the IEEE 802.1W-2001Rapid Spanning Tree Protocols (RSTP) support on Extreme devices.

> NOTE
> In addition to the features described in this chapter, refer to Root Guard and BPDU Guard in the *Configuring Spanning Tree Protocol* chapter for more details.

IEEE 802.1W-2001 RSTP provides rapid traffic reconvergence for point-to-point links within a few milliseconds (< 500 milliseconds), following the failure of a bridge or bridge port.

This reconvergence occurs more rapidly than the reconvergence provided by the IEEE 802.1D Spanning Tree Protocol or by RSTP Draft 3 because:

- STP requires a newly selected Root port to go through listening and learning stages before traffic convergence can be achieved. The STP traffic convergence time is calculated using the following formula:

  2 x FORWARD_DELAY + BRIDGE_MAX_AGE.

- Convergence in RSTP bridges is not based on any timer values. Rather, it is based on the explicit handshakes between Designated ports and their connected Root ports to achieve convergence in less than 500 milliseconds.

  > NOTE
  > The rapid convergence will not occur on ports connected to shared media devices, such as hubs. To take advantage of the rapid convergence provided by RSTP, make sure to explicitly configure all point-to-point links in a topology.

Bridges and bridge port roles

A bridge in an RSTP rapid spanning tree topology is assigned as the root bridge if it has the highest priority (lowest bridge identifier) in the topology. Other bridges are referred to as non-root bridges.

Unique roles are assigned to ports on the root and non-root bridges. Role assignments are based on the following information contained in the BPDU (RSTp packet):

- Root bridge ID
- Path cost value
- Transmitting bridge ID
- Designated port ID

RSTP algorithm uses this information to determine if the RST BPDU received by a port is superior to the RST BPDU that the port transmits. The two values are compared in the order as given above, starting with the Root bridge ID. The RST BPDU with a lower value is considered superior. The superiority and inferiority of the RST BPDU is used to assign a role to a port.

If the value of the received RST BPDU is the same as that of the transmitted RST BPDU, then the port ID in the RST BPDUs are compared. The RST BPDU with the lower port ID is superior. Port roles are then calculated appropriately.

The port's role is included in the BPDU that it transmits. The BPDU transmitted by an RSTP port is referred to as an RST BPDU, while it is operating in RSTP mode.

Ports can have one of the following roles:

- **Root** - Provides the lowest cost path to the root bridge from a specific bridge
- **Designated** - Provides the lowest cost path to the root bridge from a LAN to which it is connected
- **Alternate** - Provides an alternate path to the root bridge when the root port goes down
- **Backup** - Provides a backup to the LAN when the Designated port goes down
- **Disabled** - Has no role in the topology

Assignment of port roles

At system start-up, all RSTP-enabled bridge ports assume a Designated role. Once start-up is complete, RSTP algorithm calculates the superiority or inferiority of the RST BPDU that is received and transmitted on a port.

On a root bridge, each port is assigned a Designated port role, except for ports on the same bridge that are physically connected together. In these type of ports, the port that receives the superior RST BPDU becomes the Backup port , while the other port becomes the Designated port .

On non-root bridges, ports are assigned as follows:

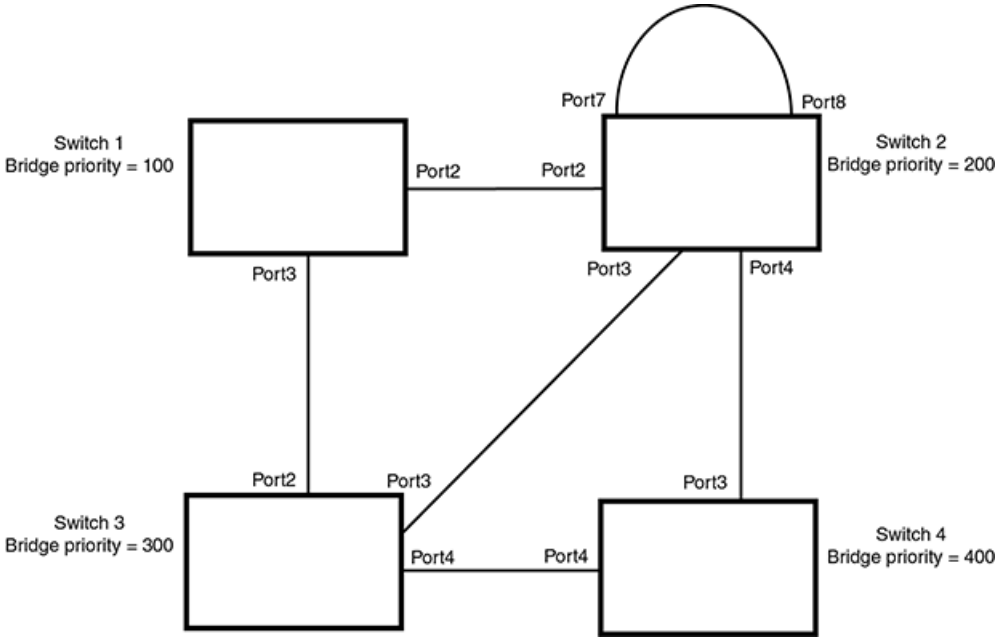
- The port that receives the RST BPDU with the lowest path cost from the root bridge becomes the Root port .
- If two ports on the same bridge are physically connected, the port that receives the superior RST BPDU becomes the Backup port , while the other port becomes the Designated port .
- If a non-root bridge already has a Root port, then the port that receives an RST BPDU that is superior to those it can transmit becomes the Alternate port .
- If the RST BPDU that a port receives is inferior to the RST BPDUs it transmits, then the port becomes a Designated port .
- If the port is down or if RSTP is disabled on the port, that port is given the role of Disabled port . Disabled ports have no role in the topology. However, if RSTP is enabled on a port with a link down and the link of that port comes up, then that port assumes one of the following port roles: Root, Designated, Alternate, or Backup.

The following example (Figure 65) explains role assignments in a simple RSTP topology.

NOTE
All examples in this document assume that all ports in the illustrated topologies are point-to-point links and are homogeneous (they have the same path cost value) unless otherwise specified.

The topology in Figure 65 contains four bridges. Switch 1 is the root bridge since it has the lowest bridge priority. Switch 2 through Switch 4 are non-root bridges.

FIGURE 64 Simple RSTP topology



Ports on Switch 1

All ports on Switch 1, the root bridge, are assigned Designated port roles.

Ports on Switch 2

Port2 on Switch 2 directly connects to the root bridge; therefore, Port2 is the Root port.

Switch 2's bridge priority value is superior to that of Switch 3 and Switch 4; therefore, the ports on Switch 2 that connect to Switch 3 and Switch 4 are given the Designated port role.

Furthermore, Port7 and Port8 on Switch 2 are physically connected. The RST BPDUs transmitted by Port7 are superior to those Port8 transmits. Therefore, Switch 2 is the Backup port and Port7 is the Designated port.

Ports on Switch 3

Port2 on Switch 3 directly connects to the Designated port on the root bridge; therefore, it assumes the Root port role.

The root path cost of the RST BPDUs received on Port4/Switch 3 is inferior to the RST BPDUs transmitted by the port; therefore, Port4/Switch 3 becomes the Designated port.

Similarly, Switch 3 has a bridge priority value inferior to Switch 2. Port3 on Switch 3 connects to Port 3 on Switch 2. This port will be given the Alternate port role, since a Root port is already established on this bridge.

Ports Switch 4

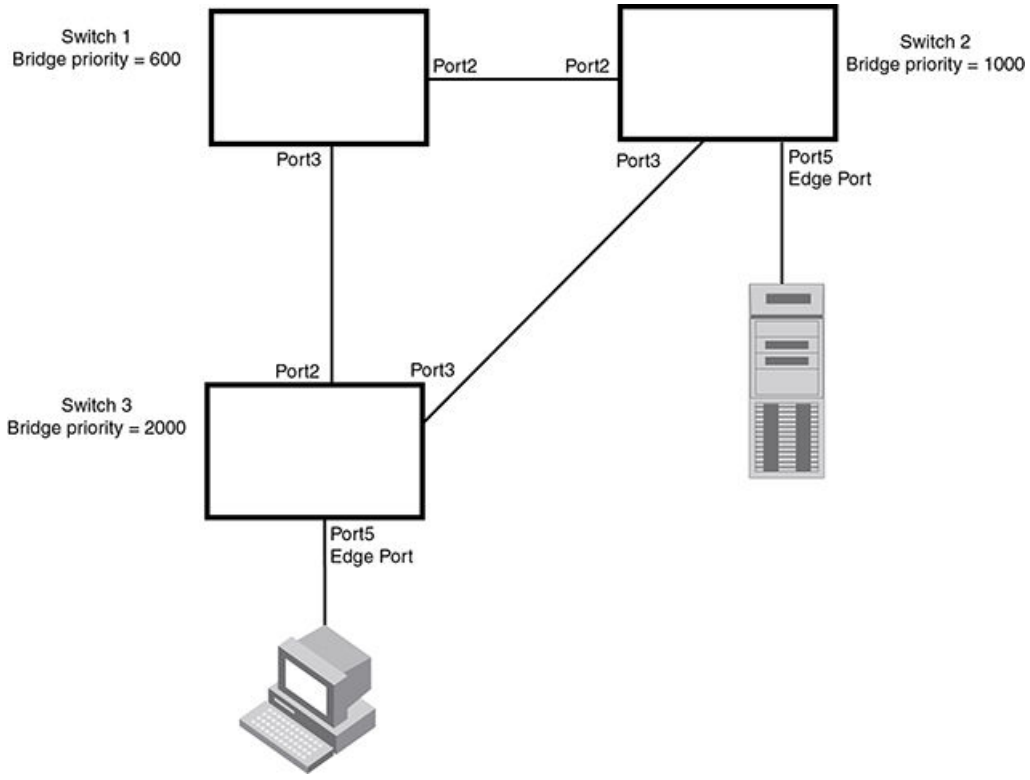
Switch 4 is not directly connected to the root bridge. It has two ports with superior incoming RST BPDUs from two separate LANs: Port3 and Port4. The RST BPDUs received on Port3 are superior to the RST BPDUs received on port 4; therefore, Port3 becomes the Root port and Port4 becomes the Alternate port.

Edge ports and Edge port roles

Extreme's implementation of RSTP allows ports that are configured as Edge ports to be present in an RSTP topology. (Figure 66). Edge ports are ports of a bridge that connect to workstations or computers. Edge ports do not register any incoming BPDU activities.

Edge ports assume Designated port roles. Port flapping does not cause any topology change events on Edge ports since RSTP does not consider Edge ports in the spanning tree calculations.

FIGURE 65 Topology with edge ports



However, if any incoming RST BPDU is received from a previously configured Edge port, RSTP automatically makes the port as a non-edge port. This is extremely important to ensure a loop free Layer 2 operation since a non-edge port is part of the active RSTP topology.

The bridge detection state module can auto-detect an Edge port and a non-edge port. An administrator can also configure a port to be an Edge port. It is recommended that Edge ports are configured explicitly to take advantage of the Edge port feature, instead of allowing the protocol to auto-detect them.

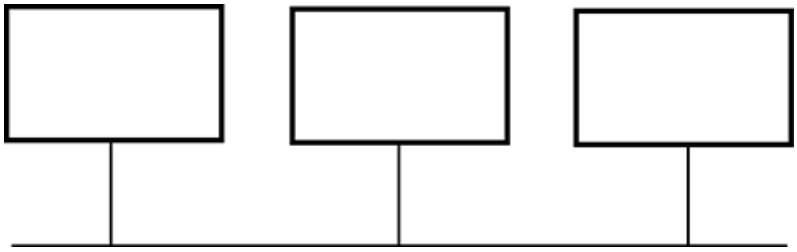
Point-to-point ports

To take advantage of the RSTP features, ports on an RSTP topology should be explicitly configured as point-to-point links. Shared media should not be configured as point-to-point links.

> NOTE
> Configuring shared media or non-point-to-point links as point-to-point links could lead to Layer 2 loops.

The topology in Figure 67 is an example of shared media that should not be configured as point-to-point links. In Figure 67, a port on a bridge communicates or is connected to at least two ports.

FIGURE 66 Example of shared media



Bridge port states

Ports roles can have one of the following states:

- **Forwarding** - RSTP is allowing the port to send and receive all packets.
- **Discarding** - RSTP has blocked data traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is forwarding. When a port is in this state, the port does not transmit or receive data frames, but the port does continue to receive RST BPDUs. This state corresponds to the listening and blocking states of 802.1D.
- **Learning** - RSTP is allowing MAC address entries to be added to the filtering database but does not permit forwarding of data frames. The device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
- **Disabled** - The port is not participating in RSTP. This can occur when the port is disconnected or RSTP is administratively disabled on the port.

A port on a non-root bridge with the role of Root port is always in a forwarding state. If another port on that bridge assumes the Root port role, then the old Root port moves into a discarding state as it assumes another port role.

A port on a non-root bridge with a Designated role starts in the discarding state. When that port becomes elected to the Root port role, RSTP quickly places it into a forwarding state. However, if the Designated port is an Edge port, then the port starts and stays in a forwarding state and it cannot be elected as a Root port.

A port with an Alternate or Backup role is always in a discarding state. If the port's role changes to Designated, then the port changes into a forwarding state.

If a port on one bridge has a Designated role and that port is connected to a port on another bridge that has an Alternate or Backup role, the port with a Designated role cannot be given a Root port role until two instances of the forward delay timer expires on that port.

Edge port and non-Edge port states

As soon as a port is configured as an Edge port, it goes into a forwarding state instantly (in less than 100 msec).

When the link to a port comes up and RSTP detects that the port is an Edge port, that port instantly goes into a forwarding state.

If RSTP detects that port as a non-edge port, the port goes into a forwarding state within four seconds of link up or after two hello timer expires on the port.

Changes to port roles and states

To achieve convergence in a topology, a port's role and state changes as it receives and transmits new RST BPDUs. Changes in a port's role and state constitute a topology change. Besides the superiority and inferiority of the RST BPDU, bridge-wide and per-port state machines are used to determine a port's role as well as a port's state. Port state machines also determine when port role and state changes occur.

State machines

The bridge uses the Port Role Selection state machine to determine if port role changes are required on the bridge. This state machine performs a computation when one of the following events occur:

- New information is received on any port on the bridge
- The timer expires for the current information on a port on the bridge

Each port uses the following state machines:

- **Port Information** - This state machine keeps track of spanning-tree information currently used by the port. It records the origin of the information and ages out any information that was derived from an incoming BPDU.
- **Port Role Transition** - This state machine keeps track of the current port role and transitions the port to the appropriate role when required. It moves the Root port and the Designated port into forwarding states and moves the Alternate and Backup ports into discarding states.
- **Port Transmit** - This state machine is responsible for BPDU transmission. It checks to ensure only the maximum number of BPDUs per hello interval are sent every second. Based on what mode it is operating in, it sends out either legacy BPDUs or RST BPDUs. In this document legacy BPDUs are also referred to as STP BPDUs.
- **Port Protocol Migration** - This state machine deals with compatibility with 802.1D bridges. When a legacy BPDU is detected on a port, this state machine configures the port to transmit and receive legacy BPDUs and operate in the legacy mode.
- **Topology Change** - This state machine detects, generates, and propagates topology change notifications. It acknowledges Topology Change Notice (TCN) messages when operating in 802.1D mode. It also flushes the MAC table when a topology change event takes place.
- **Port State Transition** - This state machine transitions the port to a discarding, learning, or forwarding state and performs any necessary processing associated with the state changes.
- **Port Timers** - This state machine is responsible for triggering any of the state machines described above, based on expiration of specific port timers.

In contrast to the 802.1D standard, the RSTP standard does not have any bridge specific timers. All timers in the CLI are applied on a per-port basis, even though they are configured under bridge parameters.

RSTP state machines attempt to quickly place the ports into either a forwarding or discarding state. Root ports are quickly placed in forwarding state when both of the following events occur:

- It is assigned to be the Root port.
- It receives an RST BPDU with a proposal flag from a Designated port. The proposal flag is sent by ports with a Designated role when they are ready to move into a forwarding state.

When a the role of Root port is given to another port, the old Root port is instructed to reroot. The old Root port goes into a discarding state and negotiates with its peer port for a new role and a new state. A peer port is the port on the other bridge to which the port is connected. For example, in Handshake when no root port is elected on page 287, Port1 of Switch 200 is the peer port of Port2 of Switch 100.

A port with a Designated role is quickly placed into a forwarding state if one of the following occurs:

- The Designated port receives an RST BPDU that contains an agreement flag from a Root port
- The Designated port is an Edge port

However, a Designated port that is attached to an Alternate port or a Backup port must wait until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state.

Backup ports are quickly placed into discarding states.

Alternate ports are quickly placed into discarding states.

A port operating in RSTP mode may enter a learning state to allow MAC address entries to be added to the filtering database; however, this state is transient and lasts only a few milliseconds, if the port is operating in RSTP mode and if the port meets the conditions for rapid transition.

Handshake mechanisms

To rapidly transition a Designated or Root port into a forwarding state, the Port Role Transition state machine uses handshake mechanisms to ensure loop free operations. It uses one type of handshake if no Root port has been assigned on a bridge, and another type if a Root port has already been assigned.

Handshake when no root port is elected

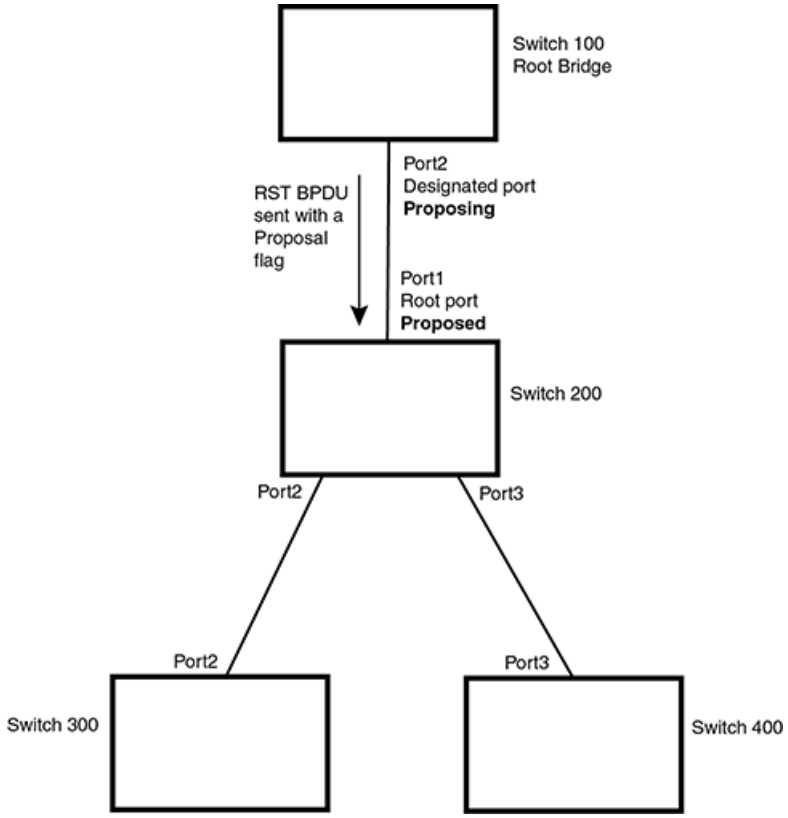
If a Root port has not been assigned on a bridge, RSTP uses the Proposing -> Proposed -> Sync -> Synced -> Agreed handshake:

- **Proposing** - The Designated port on the root bridge sends an RST BPDU packet to its peer port that contains a proposal flag. The proposal flag is a signal that indicates that the Designated port is ready to put itself in a forwarding state (see the diagram below). The Designated port continues to send this flag in its RST BPDU until it is placed in a forwarding state (see the diagram titled "Agree stage" below) or is forced to operate in 802.1D mode. (Refer to the Compatibility of RSTP with 802.1D section)
- **Proposed** - When a port receives an RST BPDU with a proposal flag from the Designated port on its point-to-point link, it asserts the Proposed signal and one of the following occurs (see the diagram below):
  - If the RST BPDU that the port receives is superior to what it can transmit, the port assumes the role of a Root port. (Refer to the section on Bridges and bridge port roles.)
  - If the RST BPDU that the port receives is inferior to what it can transmit, then the port is given the role of Designated port.

  NOTE
  Proposed will never be asserted if the port is connected on a shared media link.

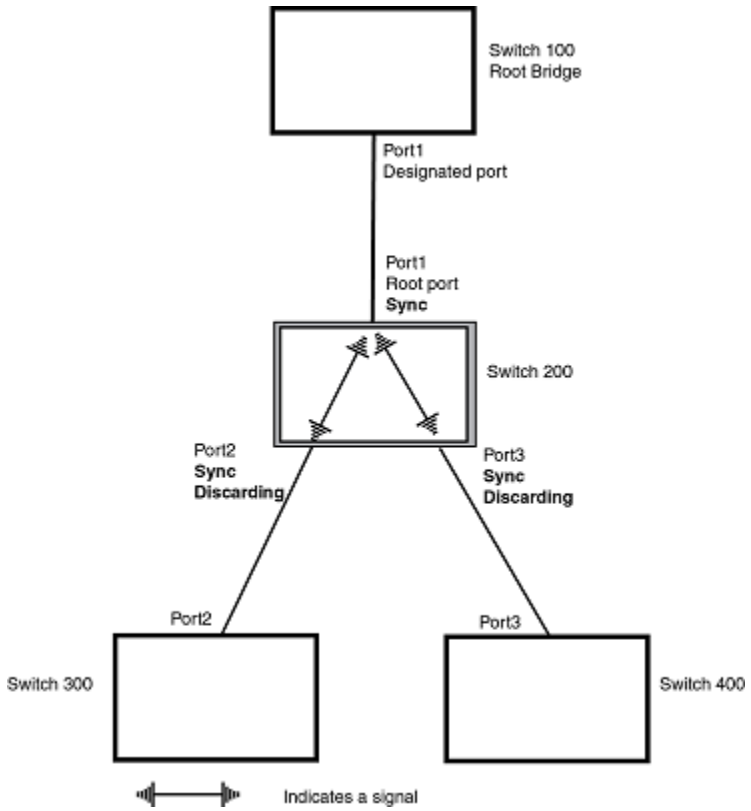
In the following diagram, Port3/Switch 200 is elected as the Root port.

FIGURE 67 Proposing and proposed stage



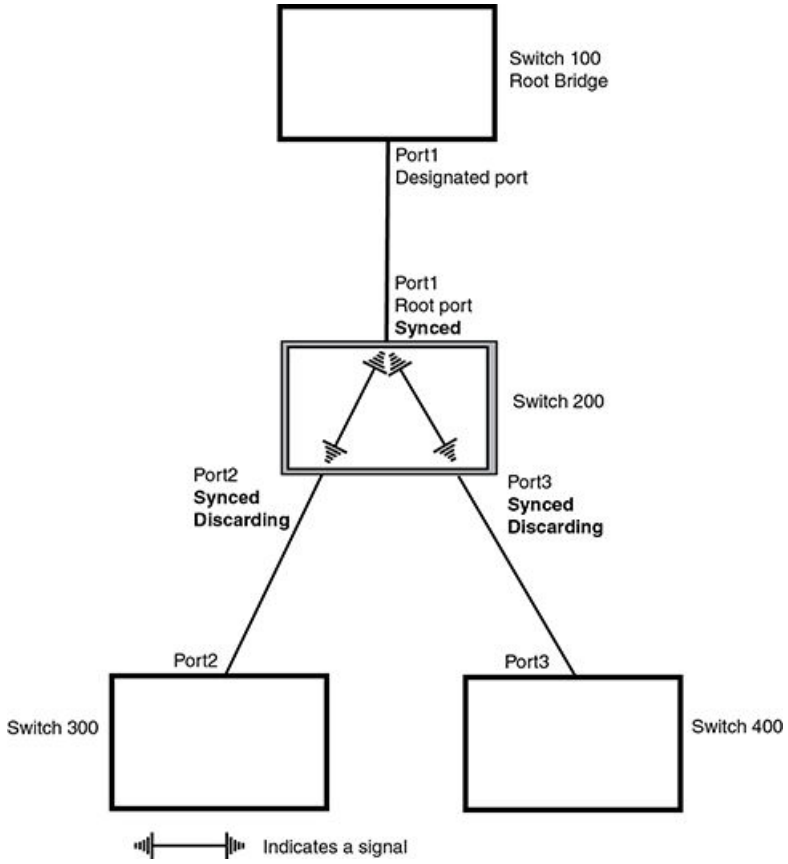
- **Sync** - Once the Root port is elected, it sets a sync signal on all the ports on the bridge. The signal tells the ports to synchronize their roles and states (see the following diagram). Ports that are non-edge ports with a role of Designated port change into a discarding state. These ports have to negotiate with their peer ports to establish their new roles and states.

FIGURE 68 Sync stage



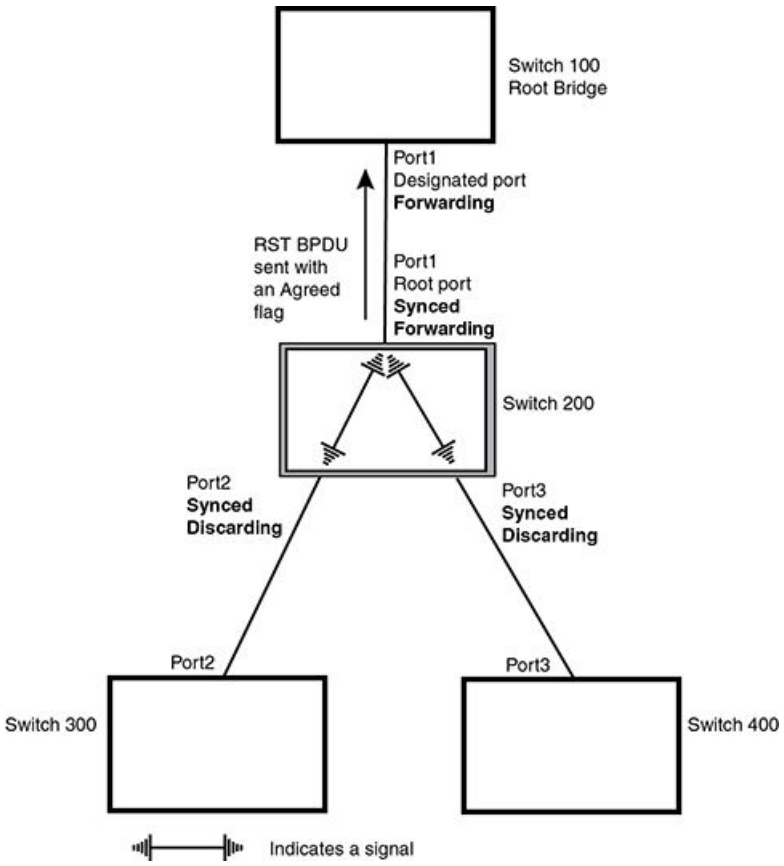
- **Synced** - Once the Designated port changes into a discarding state, it asserts a synced signal. Immediately, Alternate ports and Backup ports are synced. The Root port monitors the synced signals from all the bridge ports. Once all bridge ports asserts a synced signal, the Root port asserts its own synced signal (see the following diagram).

FIGURE 69 Synced stage



- **Agreed** - The Root port sends back an RST BPDU containing an agreed flag to its peer Designated port and moves into the forwarding state. When the peer Designated port receives the RST BPDU, it rapidly transitions into a forwarding state.

FIGURE 70 Agree stage



At this point, the handshake mechanism is complete between Switch 100, the root bridge, and Switch 200.

Switch 200 updates the information on the Switch 200's Designated ports (Port2 and Port3) and identifies the new root bridge. The Designated ports send RST BPDUs, containing proposal flags, to their downstream bridges, without waiting for the hello timers to expire on them. This process starts the handshake with the downstream bridges.

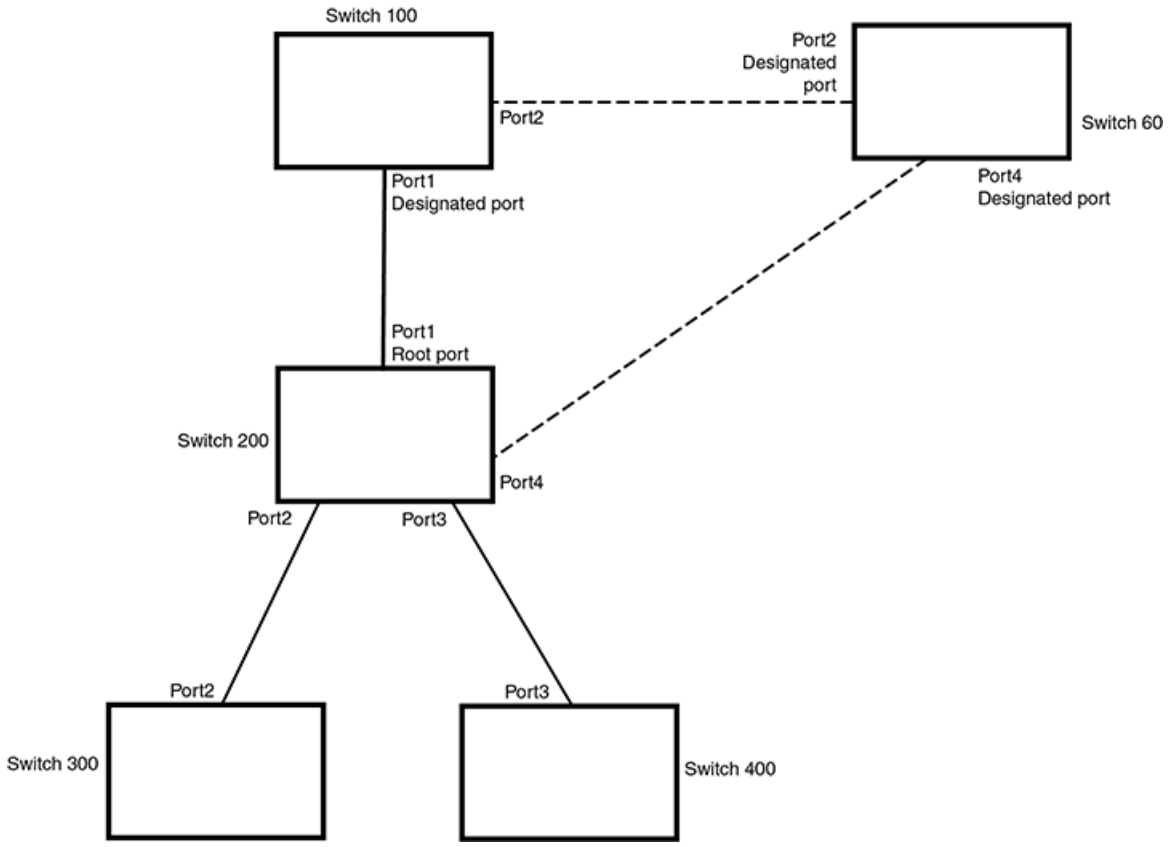
For example, Port2/Switch 200 sends an RST BPDU to Port2/Switch 300 that contains a proposal flag. Port2/Switch 300 asserts a proposed signal. Ports in Switch 300 then set sync signals on the ports to synchronize and negotiate their roles and states. Then the ports assert a synced signal and when the Root port in Switch 300 asserts it's synced signal, it sends an RST BPDU to Switch 200 with an agreed flag.

This handshake is repeated between Switch 200 and Switch 400 until all Designated and Root ports are in forwarding states.

Handshake when a root port has been elected

If a non-root bridge already has a Root port, RSTP uses a different type of handshake. For example, in Figure 72, a new root bridge is added to the topology.

FIGURE 71 Addition of a new root bridge

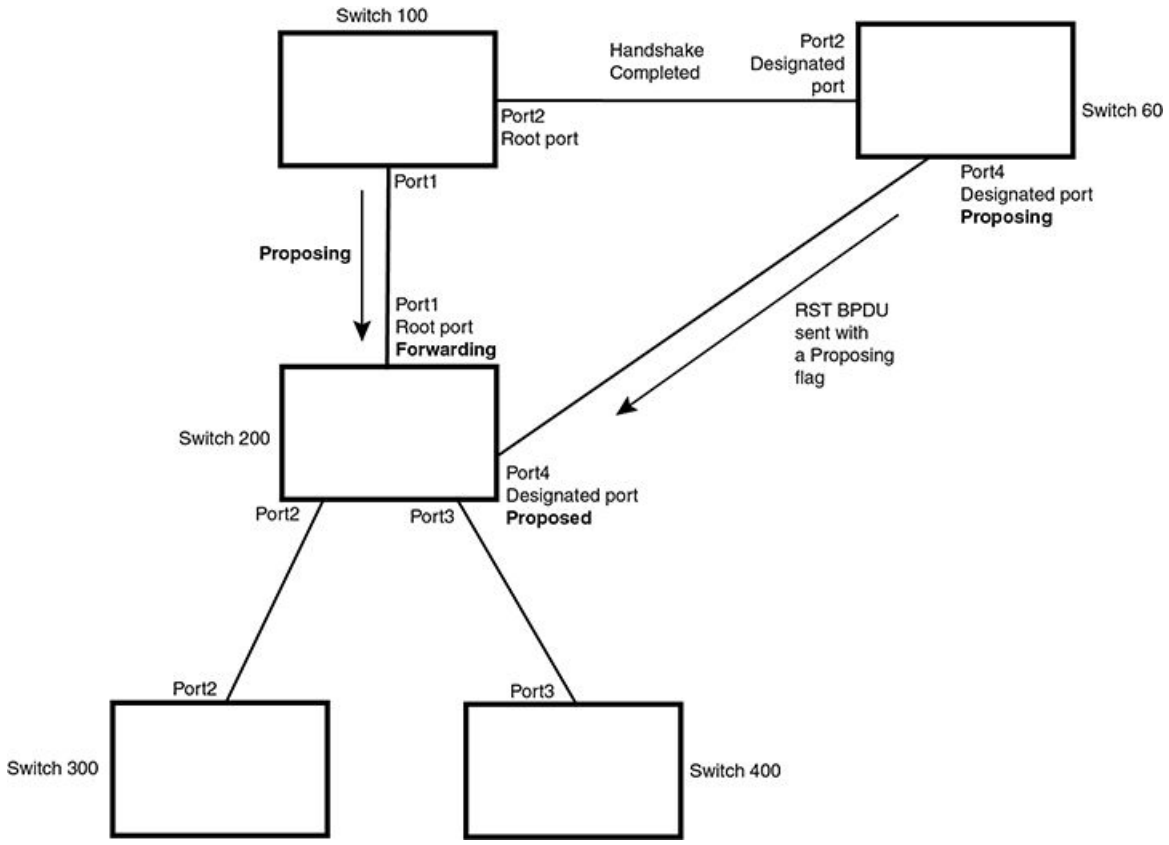


The handshake that occurs between Switch 60 and Switch 100 follows the one described in the previous section (Handshake when no root port is elected on page 287). The former root bridge becomes a non-root bridge and establishes a Root port (Figure 73).

However, since Switch 200 already had a Root port in a forwarding state, RSTP uses the Proposing -> Proposed -> Sync and Reroot -> Sync and Rerooted -> Rerooted and Synced -> Agreed handshake:

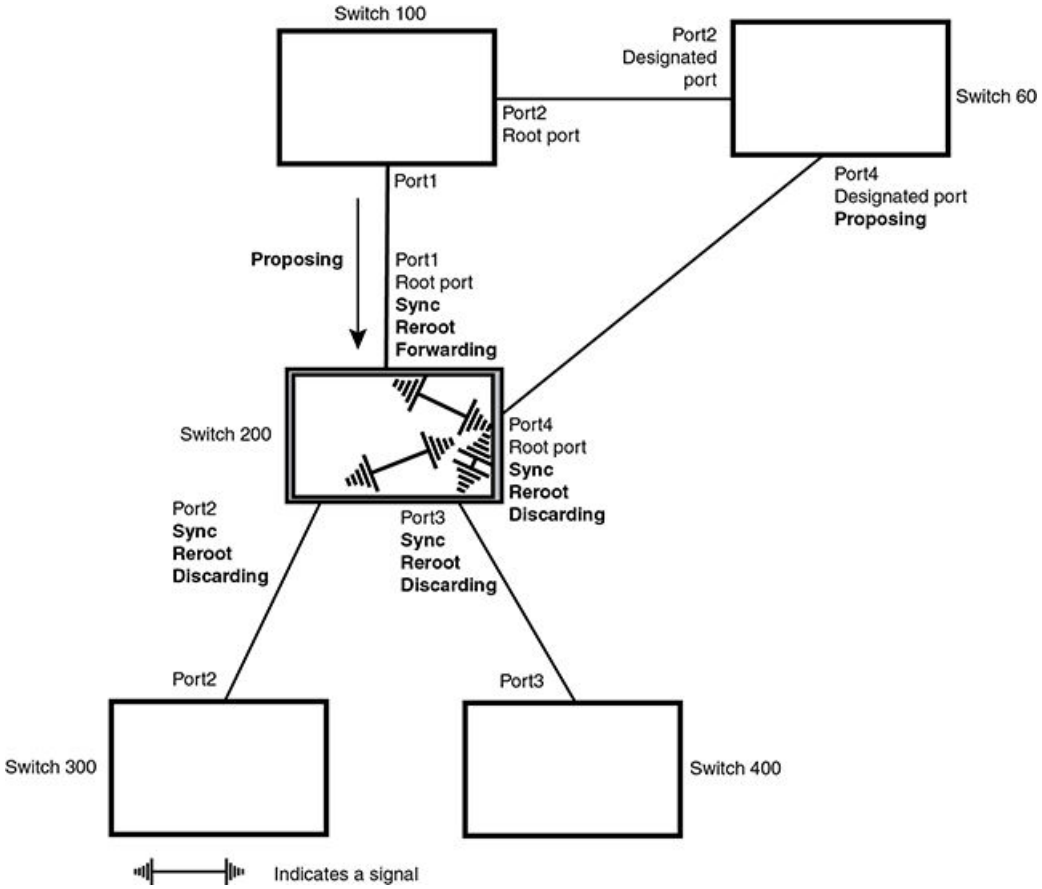
- **Proposing and Proposed** - The Designated port on the new root bridge (Port4/Switch 60) sends an RST BPDU that contains a proposing signal to Port4/Switch 200 to inform the port that it is ready to put itself in a forwarding state (Figure 73). RSTP algorithm determines that the RST BPDU that Port4/Switch 200 received is superior to what it can generate, so Port4/Switch 200 assumes a Root port role.

FIGURE 72 New root bridge sending a proposal flag



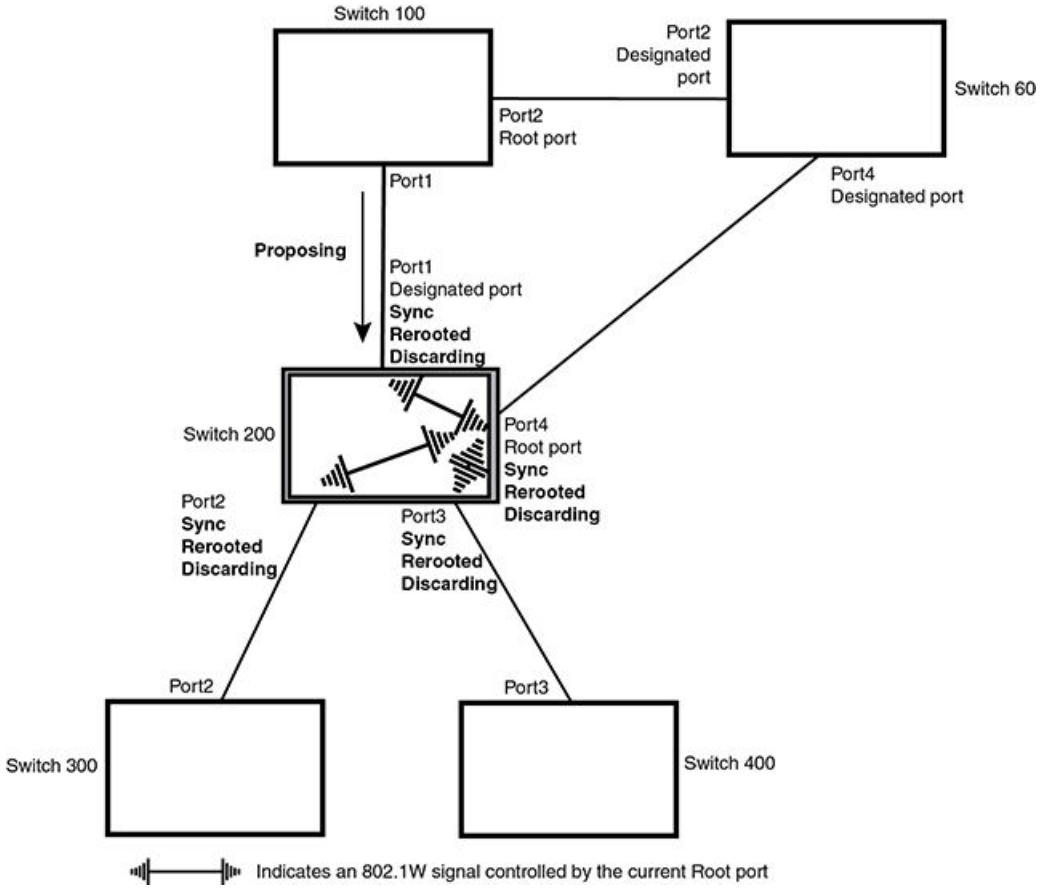
- **Sync and Reroot** - The Root port then asserts a sync and a reroot signal on all the ports on the bridge. The signal tells the ports that a new Root port has been assigned and they are to renegotiate their new roles and states. The other ports on the bridge assert their sync and reroot signals. Information about the old Root port is discarded from all ports. Designated ports change into discarding states (Figure 74).

FIGURE 73 Sync and reroot



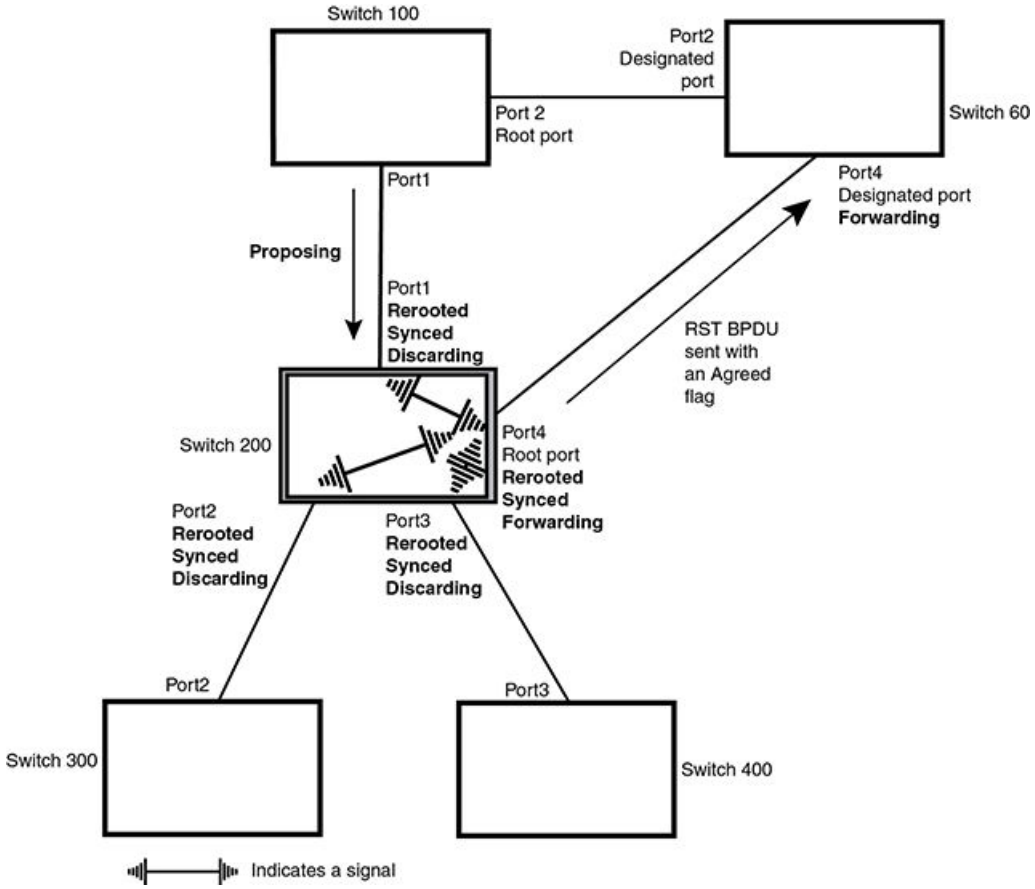
- **Sync and Rerooted** – When the ports on Switch 200 have completed the reroot phase, they assert their rerooted signals and continue to assert their sync signals as they continue in their discarding states. They also continue to negotiate their roles and states with their peer ports (Figure 75).

FIGURE 74 Sync and rerooted



- **Synced and Agree** - When all the ports on the bridge assert their synced signals, the new Root port asserts its own synced signal and sends an RST BPDU to Port4/Switch 60 that contains an agreed flag (Figure 75). The Root port also moves into a forwarding state.

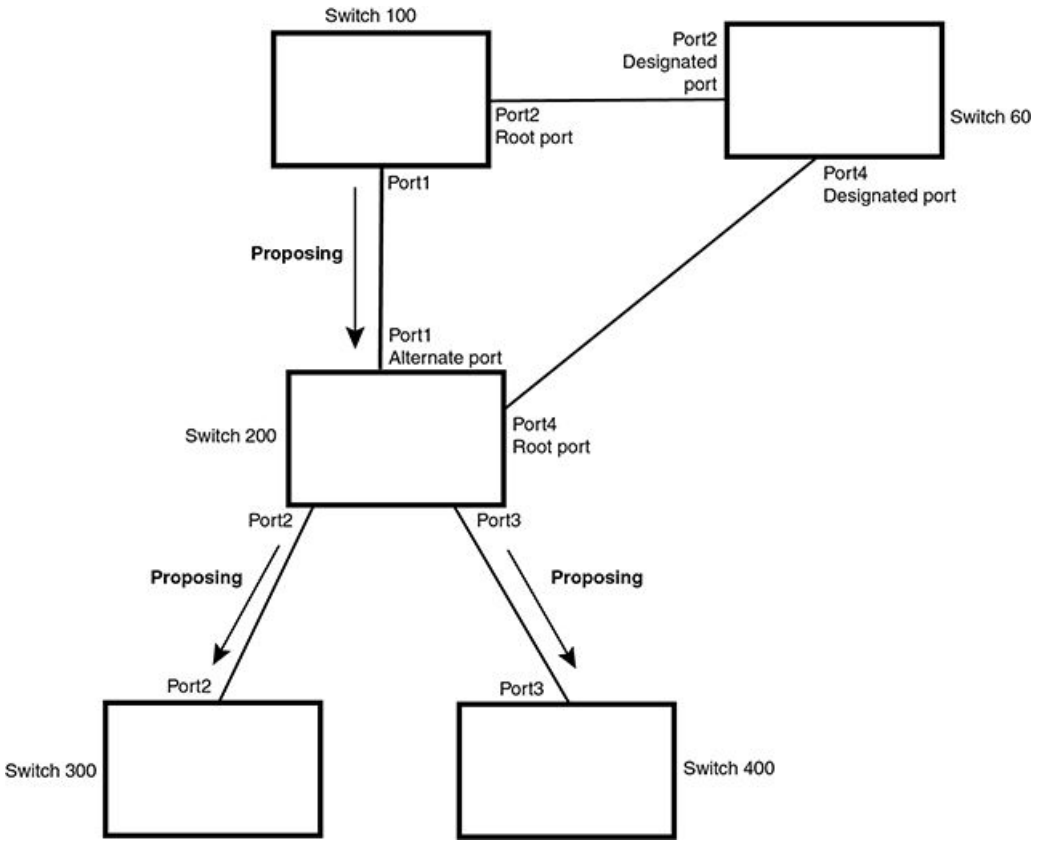
FIGURE 75 Rerooted, synced, and agreed



The old Root port on Switch 200 becomes an Alternate Port (Figure 77). Other ports on that bridge are elected to appropriate roles.

The Designated port on Switch 60 goes into a forwarding state once it receives the RST BPDU with the agreed flag.

FIGURE 76 Handshake completed after election of new root port



Recall that Switch 200 sent the agreed flag to Port4/Switch 60 and not to Port1/Switch 100 (the port that connects Switch 100 to Switch 200). Therefore, Port1/Switch 100 does not go into forwarding state instantly. It waits until two instances of the forward delay timer expires on the port before it goes into forwarding state.

At this point the handshake between the Switch 60 and Switch 200 is complete.

The remaining bridges (Switch 300 and Switch 400) may have to go through the reroot handshake if a new Root port needs to be assigned.

Convergence in a simple topology

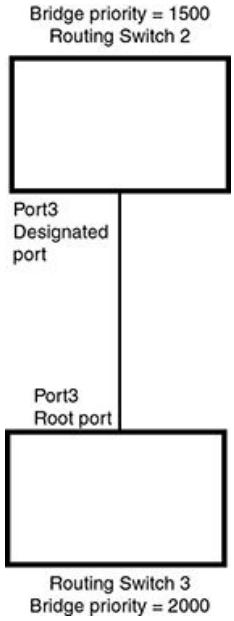
The examples in this section illustrate how RSTP convergence occurs in a simple Layer 2 topology at start-up.

NOTE
The remaining examples assume that the appropriate handshake mechanisms occur as port roles and states change.

Convergence at start up

In Figure 78, two bridges Switch 2 and Switch 3 are powered up. There are point-to-point connections between Port3/Switch 2 and Port3/Switch 3.

FIGURE 77 Convergence between two bridges



At power up, all ports on Switch 2 and Switch 3 assume Designated port roles and are at discarding states before they receive any RST BPDU.

Port3/Switch 2, with a Designated role, transmits an RST BPDU with a proposal flag to Port3/Switch 3. A ports with a Designated role sends the proposal flag in its RST BPDU when they are ready to move to a forwarding state.

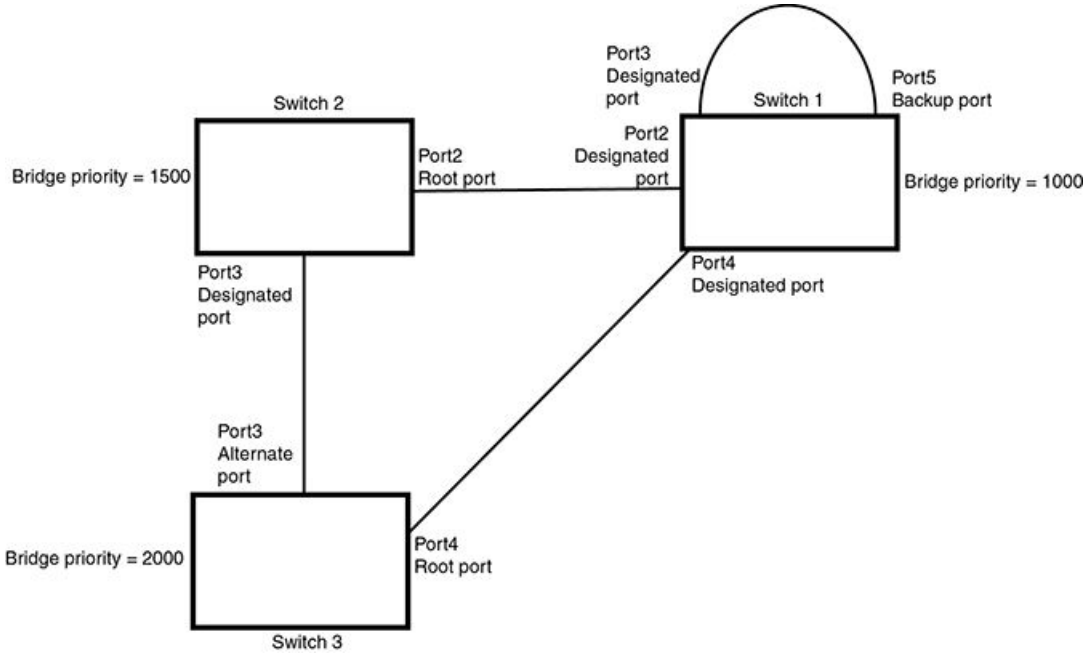
Port3/Switch 3, which starts with a role of Designated port, receives the RST BPDU and finds that it is superior to what it can transmit; therefore, Port3/Switch 3 assumes a new port role, that of a Root port. Port3/Switch 3 transmits an RST BPDU with an agreed flag back to Switch 2 and immediately goes into a forwarding state.

Port3/Switch 2 receives the RST BPDU from Port3/Switch 3 and immediately goes into a forwarding state.

Now RSTP has fully converged between the two bridges, with Port3/Switch 3 as an operational root port in forwarding state and Port3/Switch 2 as an operational Designated port in forwarding state.

Next, Switch 1 is powered up (Figure 79).

FIGURE 78 Simple Layer 2 topology



The point-to-point connections between the three bridges are as follows:

- Port2/Switch 1 and Port2/Switch 2
- Port4/Switch 1 and Port4/Switch 3
- Port3/Switch 2 and Port3/Switch 3

Ports 3 and 5 on Switch 1 are physically connected together.

At start up, the ports on Switch 1 assume Designated port roles, which are in discarding state. They begin sending RST BPDUs with proposal flags to move into a forwarding state.

When Port4/Switch 3 receives these RST BPDUs RSTP algorithm determines that they are better than the RST BPDUs that were previously received on Port3/Switch 3. Port4/Switch 3 is now selected as Root port. This new assignment signals Port3/Switch 3 to begin entering the discarding state and to assume an Alternate port role. As it goes through the transition, Port3/Switch 3 negotiates a new role and state with its peer port, Port3/Switch 2.

Port4/Switch 3 sends an RST BPDU with an agreed flag to Port4/Switch 1. Both ports go into forwarding states.

Port2/Switch 2 receives an RST BPDU. The RSTP algorithm determines that these RST BPDUs that are superior to any that any port on Switch 2 can transmit; therefore, Port2/Switch 2 assumes the role of a Root port.

The new Root port then signals all ports on the bridge to start synchronization. Since none of the ports are Edge ports, they all enter the discarding state and assume the role of Designated ports. Port3/Switch 2, which previously had a Designated role with a forwarding state, starts the discarding state. They also negotiate port roles and states with their peer ports. Port3/Switch 2 also sends an RST BPU to Port3/Switch 3 with a proposal flag to request permission go into a forwarding state.

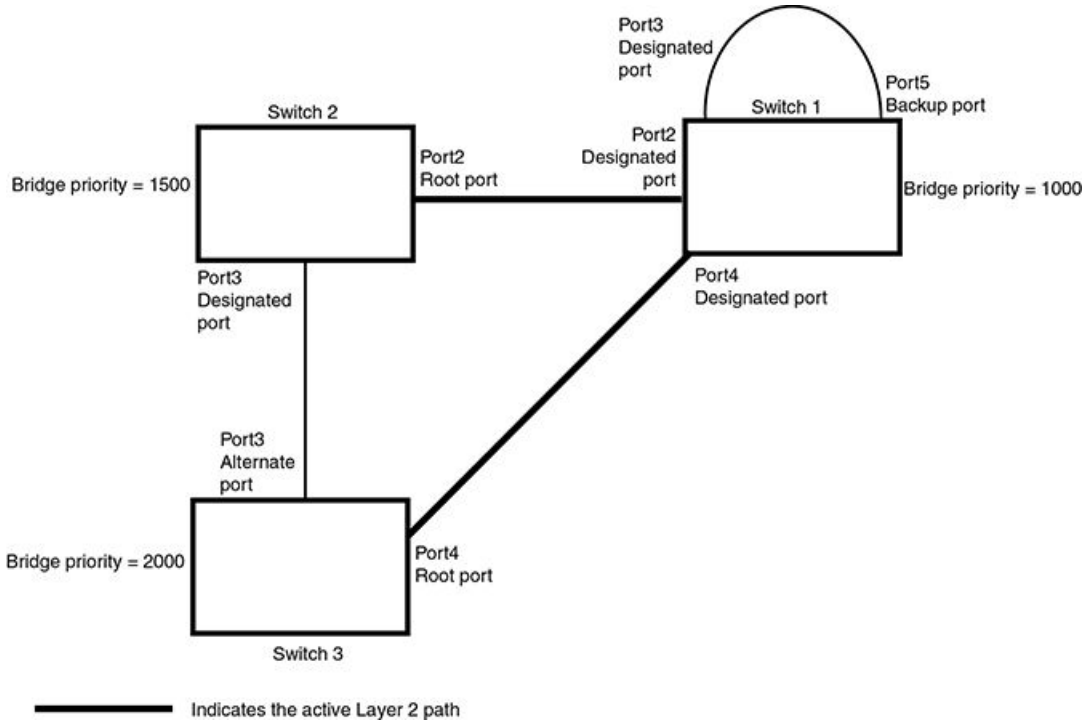
The Port2/Switch 2 bridge also sends an RST BPDU with an agreed flag Port2/Switch 1 that Port2 is the new Root port. Both ports go into forwarding states.

Now, Port3/Switch 3 is currently in a discarding state and is negotiating a port role. It received RST BPDUs from Port3/Switch 2. The RSTP algorithm determines that the RST BPDUs Port3/Switch 3 received are superior to those it can transmit; however, they are not superior to those that are currently being received by the current Root port (Port4). Therefore, Port3 retains the role of Alternate port.

Ports 3/Switch 1 and Port5/Switch 1 are physically connected. Port5/Switch 1 received RST BPDUs that are superior to those received on Port3/Switch 1; therefore, Port5/Switch 1 is given the Backup port role while Port3 is given the Designated port role. Port3/Switch 1, does not go directly into a forwarding state. It waits until the forward delay time expires twice on that port before it can proceed to the forwarding state.

Once convergence is achieved, the active Layer 2 forwarding path converges as shown in Figure 80.

FIGURE 79 Active Layer 2 path

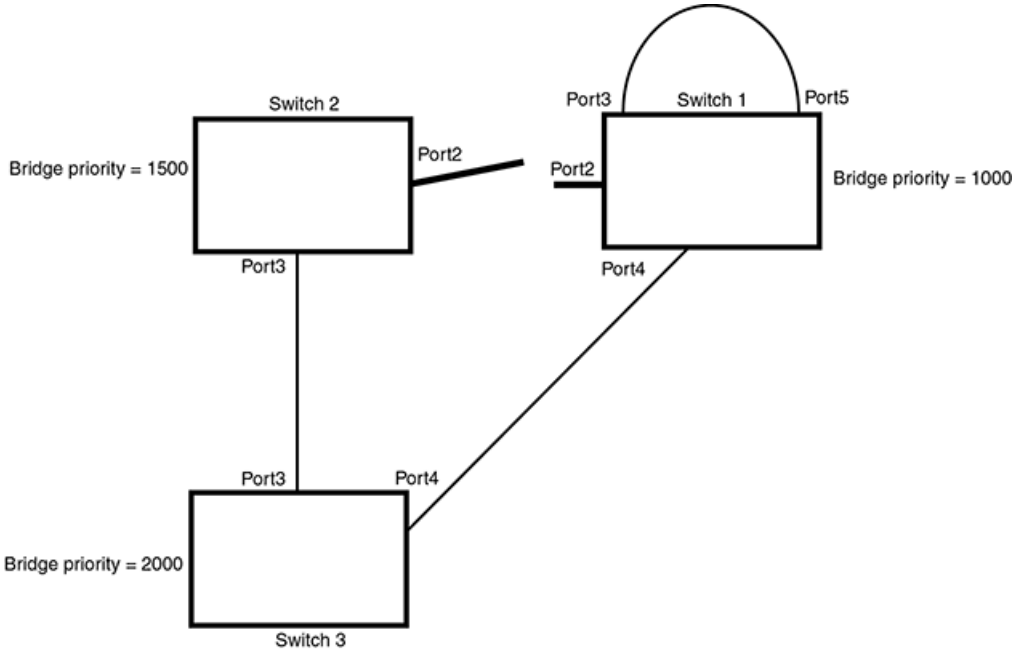


Convergence after a link failure

What happens if a link in the RSTP topology fails?

For example, Port2/Switch, which is the port that connects Switch 2 to the root bridge (Switch 1), fails. Both Switch 2 and Switch 1 notice the topology change (Figure 81).

FIGURE 80 Link failure in the topology



Switch 1 sets its Port2 into a discarding state.

At the same time, Switch 2 assumes the role of a root bridge since its root port failed and it has no operational Alternate port. Port3/Switch 2, which currently has a Designated port role, sends an RST BPDU to Switch 3. The RST BPDU contains a proposal flag and a bridge ID of Switch 2 as its root bridge ID.

When Port3/Switch 3 receives the RST BPDUs, RSTP algorithm determines that they are inferior to those that the port can transmit. Therefore, Port3/Switch 3 is given a new role, that of a Designated port. Port3/Switch 3 then sends an RST BPDU with a proposal flag to Switch 2, along with the new role information. However, the root bridge ID transmitted in the RST BPDU is still Switch 1.

When Port3/Switch 2 receives the RST BPDU, RSTP algorithm determines that it is superior to the RST BPDU that it can transmit; therefore, Port3/Switch 2 receives a new role; that of a Root port. Port3/Switch 2 then sends an RST BPDU with an agreed flag to Port3/Switch 3. Port3/Switch 2 goes into a forwarding state.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, Port3/Switch 3 changes into a forwarding state, which then completes the full convergence of the topology.

Convergence at link restoration

When Port2/Switch 2 is restored, both Switch 2 and Switch 1 recognize the change. Port2/Switch 1 starts assuming the role of a Designated port and sends an RST BPDU containing a proposal flag to Port2/Switch 2.

When Port2/Switch 2 receives the RST BPDUs, RSTP algorithm determines that the RST BPDUs the port received are better than those received on Port3/Switch 3; therefore, Port2/Switch 2 is given the role of a Root port. All the ports on Switch 2 are informed that a new Root port has been assigned which then signals all the ports to synchronize their roles and states. Port3/Switch 2, which was the previous Root port, enters a discarding state and negotiates with other ports on the bridge to establish its new role and state, until it finally assumes the role of a Designated port.

Next, the following happens:

- Port3/Switch 2, the Designated port, sends an RST BPDU, with a proposal flag to Port3/Switch 3.

- Port2/Switch 2 also sends an RST BPDU with an agreed flag to Port2/Switch 1 and then places itself into a forwarding state.

When Port2/Switch 1 receives the RST BPDU with an agreed flag sent by Port2/Switch 2, it puts that port into a forwarding state. The topology is now fully converged.

When Port3/Switch 3 receives the RST BPDU that Port3/Switch 2 sent, RSTP algorithm determines that these RST BPDUs are superior to those that Port3/Switch 3 can transmit. Therefore, Port3/Switch 3 is given a new role, that of an Alternate port. Port3/Switch 3 immediately enters a discarding state.

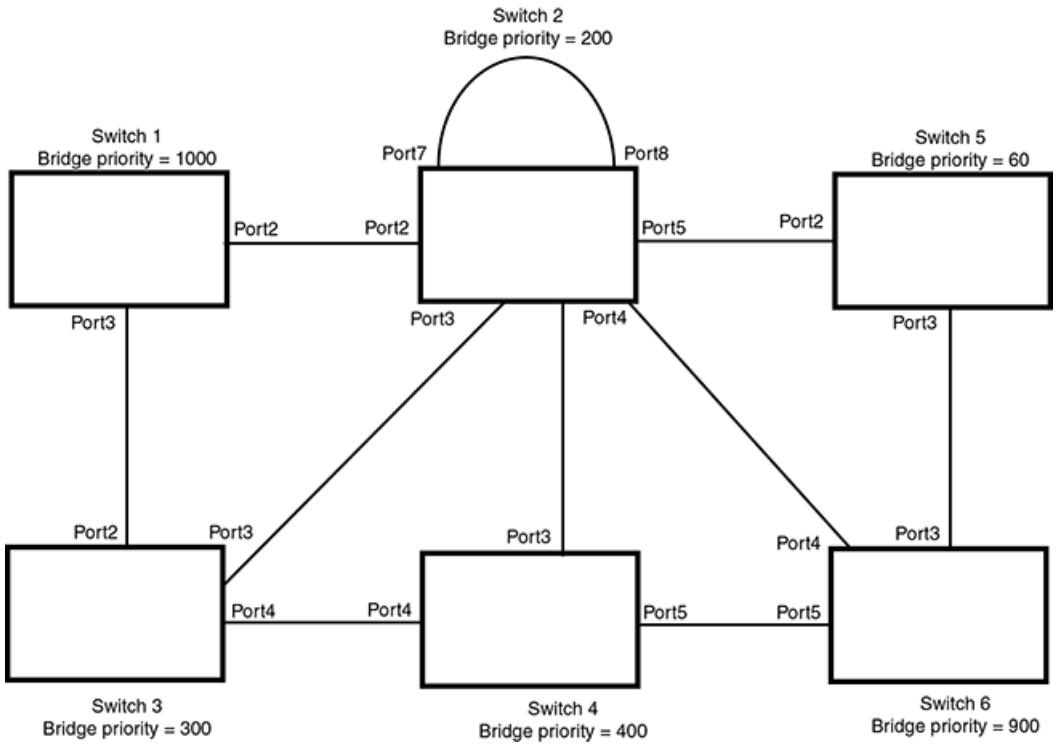
Now Port3/Switch 2 does not go into a forwarding state instantly like the Root port. It waits until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state. The wait, however, does not cause a denial of service, since the essential connectivity in the topology has already been established.

When fully restored, the topology is the same as that shown on Convergence at start up on page 297.

Convergence in a complex RSTP topology

The following is an example of a complex RSTP topology.

FIGURE 81 Complex RSTP topology



In Figure 82, Switch 5 is selected as the root bridge since it is the bridge with the highest priority. Lines in the figure show the point-to-point connection to the bridges in the topology.

Switch 5 sends an RST BPDU that contains a proposal flag to Port5/Switch 2. When handshakes are completed in Switch 5, Port5/Switch 2 is selected as the Root port on Switch 2. All other ports on Switch 2 are given Designated port role with discarding states.

Port5/Switch 2 then sends an RST BPDU with an agreed flag to Switch 5 to confirm that it is the new Root port and the port enters a forwarding state. Port7 and Port8 are informed of the identity of the new Root port. RSTP algorithm selects Port7 as the Designated port while Port8 becomes the Backup port.

Port3/Switch 5 sends an RST BPDU to Port3/Switch 6 with a proposal flag. When Port3/Switch 5 receives the RST BPDU, handshake mechanisms select Port3 as the Root port of Switch 6. All other ports are given a Designated port role with discarding states. Port3/Switch 6 then sends an RST BPDU with an agreed flag to Port3/Switch 5 to confirm that it is the Root port. The Root port then goes into a forwarding state.

Now, Port4/Switch 6 receives RST BPDUs that are superior to what it can transmit; therefore, it is given the Alternate port role. The port remains in discarding state.

Port5/Switch 6 receives RST BPDUs that are inferior to what it can transmit. The port is then given a Designated port role.

Next Switch 2 sends RST BPDUs with a proposal flag to Port3/Switch 4. Port3 becomes the Root port for the bridge; all other ports are given a Designated port role with discarding states. Port3/Switch 4 sends an RST BPDU with an agreed flag to Switch 2 to confirm that it is the new Root port. The port then goes into a forwarding state.

Now Port4/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is then given an Alternate port role, and remains in discarding state.

Likewise, Port5/Switch 4 receives an RST BPDU that is superior to what it can transmit. The port is also given an Alternate port role, and remains in discarding state.

Port2/Switch 2 transmits an RST BPDU with a proposal flag to Port2/Switch 1. Port2/Switch 1 becomes the Root port. All other ports on Switch 1 are given Designated port roles with discarding states.

Port2/Switch 1 sends an RST BPDU with an agreed flag to Port2/Switch 2 and Port2/Switch 1 goes into a forwarding state.

Port3/Switch 1 receives an RST BPDUs that is inferior to what it can transmit; therefore, the port retains its Designated port role and goes into forwarding state only after the forward delay timer expires twice on that port while it is still in a Designated role.

Port3/Switch 2 sends an RST BPDU to Port3/Switch 3 that contains a proposal flag. Port3/Switch 3 becomes the Root port, while all other ports on Switch 3 are given Designated port roles and go into discarding states. Port3/Switch 3 sends an RST BPDU with an agreed flag to Port3/Switch 2 and Port3/Switch 3 goes into a forwarding state.

Now, Port2/Switch 3 receives an RST BPDUs that is superior to what it can transmit so that port is given an Alternate port state.

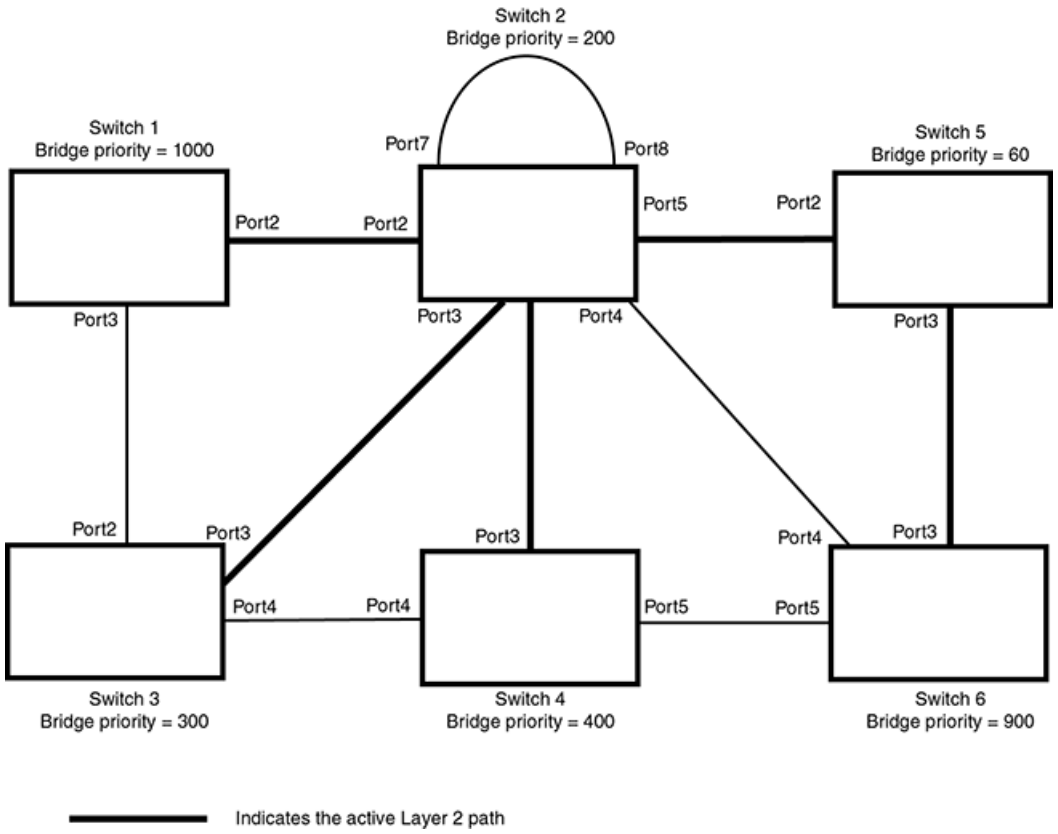
Port4/Switch 3 receives an RST BPDU that is inferior to what it can transmit; therefore, the port retains its Designated port role.

Ports on all the bridges in the topology with Designated port roles that received RST BPDUs with agreed flags go into forwarding states instantly. However, Designated ports that did not receive RST BPDUs with agreed flags must wait until the forward delay timer expires twice on those port. Only then will these port move into forwarding states.

The entire RSTP topology converges in less than 300 msec and the essential connectivity is established between the designated ports and their connected root ports.

After convergence is complete, Figure 83 shows the active Layer 2 path of the topology in Figure 82.

FIGURE 82 Active Layer 2 path in complex topology



Propagation of topology change

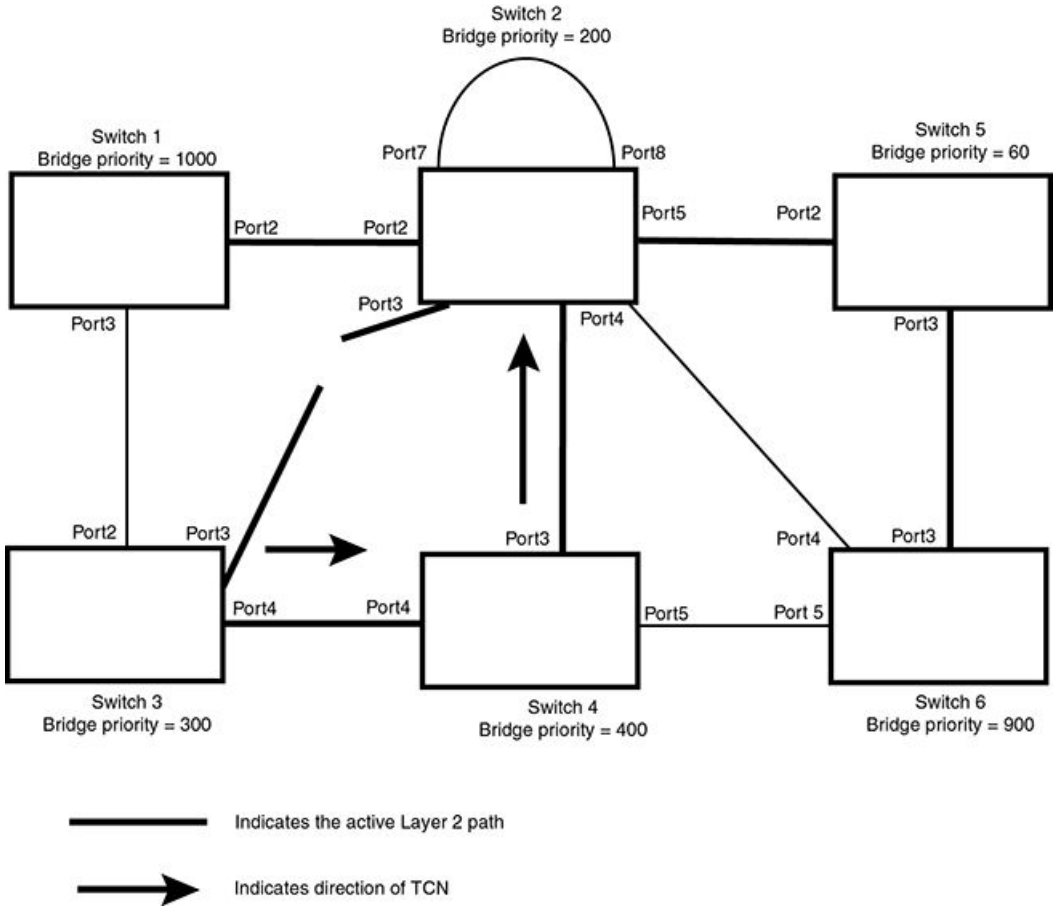
The Topology Change state machine generates and propagates the topology change notification messages on each port. When a Root port or a Designated port goes into a forwarding state, the Topology Change state machine on those ports send a topology change notice (TCN) to all the bridges in the topology to propagate the topology change.

NOTE
Edge ports, Alternate ports, or Backup ports do not need to propagate a topology change.

The TCN is sent in the RST BPDU that a port sends. Ports on other bridges in the topology then acknowledge the topology change once they receive the RST BPDU, and send the TCN to other bridges until all the bridges are informed of the topology change.

For example, Port3/Switch 2 in Figure 84, fails. Port4/Switch 3 becomes the new Root port. Port4/Switch 3 sends an RST BPDU with a TCN to Port4/Switch 4. To propagate the topology change, Port4/Switch 4 then starts a TCN timer on itself, on the bridge's Root port, and on other ports on that bridge with a Designated role. Then Port3/Switch 4 sends RST BPDU with the TCN to Port4/Switch 2. (Note the new active Layer 2 path in Figure 84.)

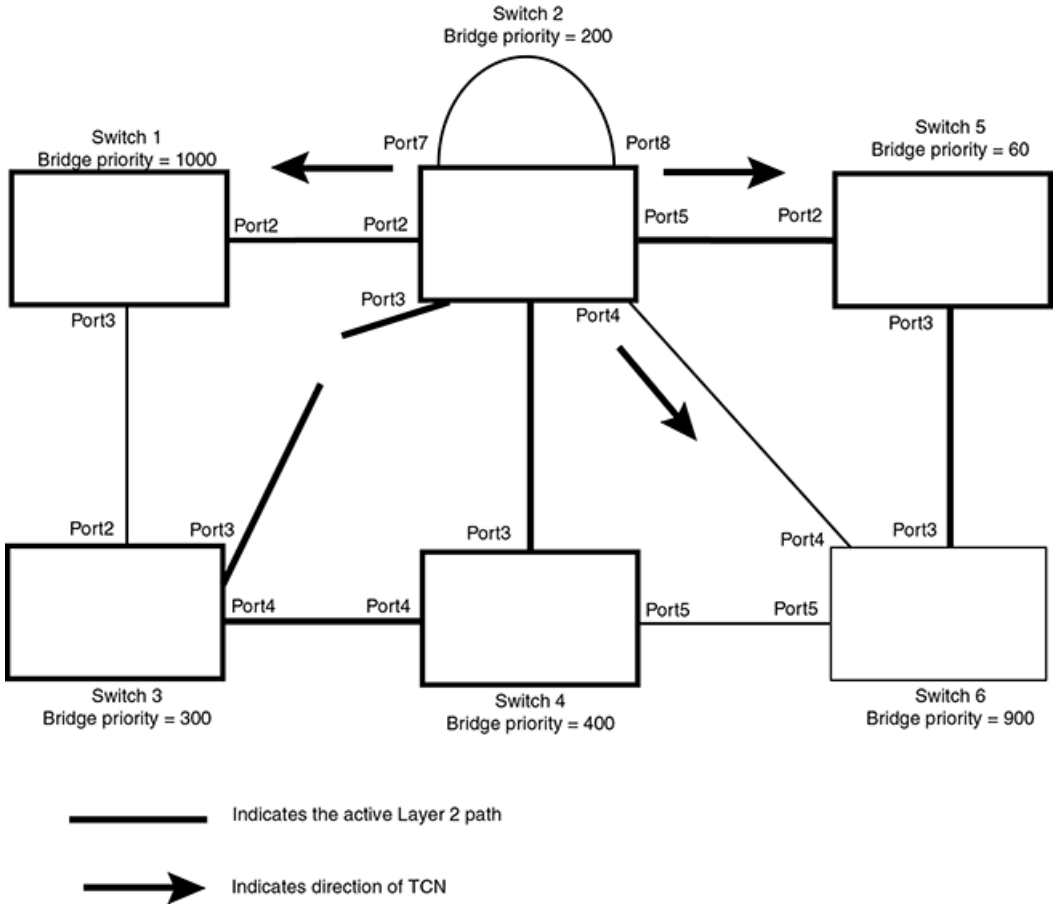
FIGURE 83 Beginning of topology change notice



Switch 2 then starts the TCN timer on the Designated ports and sends RST BPDUs that contain the TCN as follows:

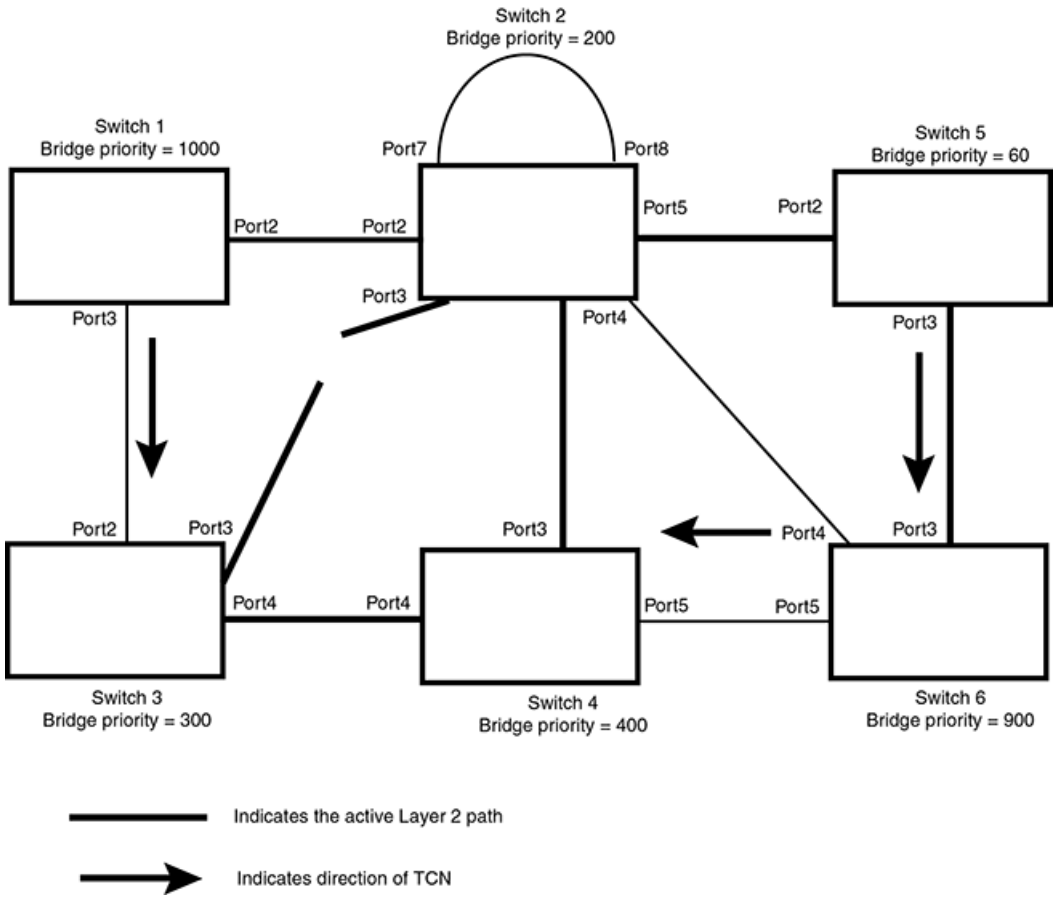
- Port5/Switch 2 sends the TCN to Port2/Switch 5
- Port4/Switch 2 sends the TCN to Port4/Switch 6
- Port2/Switch 2 sends the TCN to Port2/Switch 1

FIGURE 84 Sending TCN to bridges connected to Switch 2



Then FRY1, Switch 5, and Switch 6 send RST BPDUs that contain the TCN to Switch 3 and Switch 4 to complete the TCN propagation.

FIGURE 85 Completing the TCN propagation



Compatibility of RSTP with 802.1D

RSTP-enabled bridges are backward compatible with IEEE 802.1D bridges. This compatibility is managed on a per-port basis by the Port Migration state machine.

> NOTE
> **Intermixing the two types of bridges in the network topology is not advisable if you want to take advantage of the rapid convergence feature.**

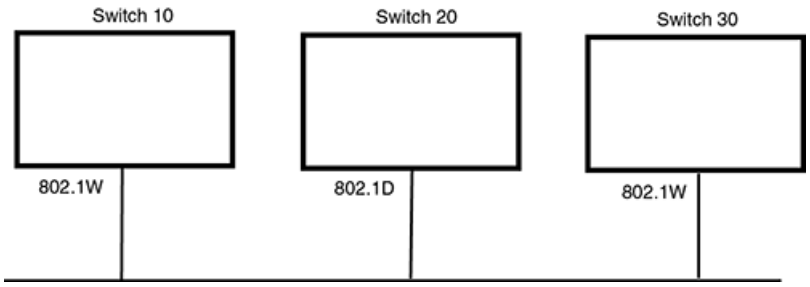
Compatibility with 802.1D means that an RSTP-enabled port can send BPDUs in the STP or 802.1D format when one of the following events occur:

- The port receives a legacy BPDU. A legacy BPDU is an STP BPDU or a BPDU in an 802.1D format. The port that receives the legacy BPDU automatically configures itself to behave like a legacy port. It sends and receives legacy BPDUs only.
- The entire bridge is configured to operate in an 802.1D mode when an administrator sets the
- bridge parameter to zero at the CLI, forcing all ports on the bridge to send legacy BPDUs only.

Once a port operates in the 802.1D mode, 802.1D convergence times are used and rapid convergence is not realized.

For example, in Figure 87, Switch 10 and Switch 30 receive legacy BPDUs from Switch 20. Ports on Switch 10 and Switch 30 begin sending BPDUs in STP format to allow them to operate transparently with Switch 20.

FIGURE 86 RSTP bridges with an 802.1D bridge



Once Switch 20 is removed from the LAN, Switch 10 and Switch 30 receive and transmit BPDUs in the STP format to and from each other. This state will continue until the administrator enables the **force-migration-check** command to force the bridge to send RSTP BPDU during a migrate time period. If ports on the bridges continue to hear only STP BPDUs after this migrate time period, those ports will return to sending STP BPDUs. However, when the ports receive RST BPDUs during the migrate time period, the ports begin sending RST BPDUs. The migrate time period is non-configurable. It has a value of three seconds.

NOTE

The IEEE standards state that RSTP bridges need to interoperate with 802.1D bridges. IEEE standards set the path cost of RSTP bridges to be between 1 and 200,000,000; whereas path cost of 802.1D bridges are set between 1 and 65,535. In order for the two bridge types to be able to interoperate in the same topology, the administrator needs to configure the bridge path cost appropriately. Path costs for either RSTP bridges or 802.1D bridges need to be changed; in most cases, path costs for RSTP bridges need to be changed.

Configuring RSTP parameters

The remaining RSTP sections explain how to configure the RSTP protocol on the Extreme device.

You can enable or disable RSTP at the following levels:

- **Port-based VLAN** - Affects all ports within the specified port-based VLAN. When you enable or disable RSTP within a port-based VLAN, the setting overrides the global setting. Thus, you can enable RSTP for the ports within a port-based VLAN even when RSTP is globally disabled, or disable the ports within a port-based VLAN when RSTP is globally enabled.
- **Individual port** - Affects only the individual port. However, if you change the RSTP state of the primary port in a LAG group, the change affects all ports in the LAG group.

RSTP in a LAG

The RSTP standard indicates that by default the path cost is determined by link speed. For a port having 1G the path cost is 20,000 and for 10G the path cost is 2,000. However, if a LAG is made consisting of n 1G ports where n is less than 10, the path cost remains as 20,000. The standard does not indicate pathcost explicitly for LAG interfaces or if the bandwidth is in intermediate value. Therefore, during RSTP deployment you may find that though a LAG has greater bandwidth, its in blocking/discarding state as its pathCost is same as any 1G link and the portIndex of 1G port is lower, making the LAG go into a blocking/discarding state. This behavior is not restricted to 1G or 10G link speed but span across different link speeds. The same behavior also holds TRUE for STP deployments.

Enabling or disabling RSTP in a port-based VLAN

Use the following procedure to disable or enable RSTP on the Extreme device on which you have configured a port-based VLAN. Changing the RSTP state in a VLAN affects only that VLAN.

To enable RSTP for all ports in a port-based VLAN, enter commands such as the following.

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

Syntax: [no] rstp

Enabling or disabling RSTP on a single spanning tree

To globally enable RSTP for all ports of a single spanning tree, enter the following command.

```
device(config)# rstp single
```

Syntax: [no] rstp single

Disabling or enabling RSTP on a port

The **rstp** command must be used to initially enable RSTP on ports. Both commands enable RSTP on all ports that belong to the VLAN or to the single spanning tree.

Once RSTP is enabled on a port, it can be disabled on individual ports. RSTP that have been disabled on individual ports can then be enabled as required.

> NOTE
> If you change the RSTP state of the primary port in a LAG group, the change affects all ports in that LAG group.

To disable or enable RSTP on a port, enter commands such as the following.

```
device(config)# interface 1/1
device(config-if-e1000-1/1)# no spanning-tree
```

Syntax: [no] spanning-tree

Configuring maximum number of RSTP instances

NetIron OS devices support the **system-max rstp** command to configure the maximum number of supported RSTP instances on a system.

[no] system-max rstp *number of instances*

The *number of instances* variable indicates the maximum number of RSTP instances that can be configured on the device. The valid number of instances are 1 through 256. The default number is 32 instances.

> NOTE
> Before you downgrade from NetIron OS Release 5.9 to a lower release and restart the device, it is recommended that you reduce the number of RSTP instances to 128 or a lower value using the **system-max rstp** command. However, if you upgrade from NetIron OS Release 5.8 (or previous releases) to 5.9 and restart, there is no change in the RSTP configuration or operation since the lower number of RSTP instances are anyway supported.

Changing RSTP bridge parameters

When you make changes to RSTP bridge parameters, the changes are applied to individual ports on the bridge.

To designate a priority for a bridge, enter a command such as the following at the VLAN level.

```
device(config)# vlan 20
device(config-vlan-20)# rstp priority 0
```

To make this change in the default VLAN, enter the following commands.

```
device(config)# vlan 1
device(config-vlan-1)# rstp priority 0
```

Syntax: [**rstp forward-delay** *value*] | [**hello-time** *value*] | [**max-age** *time*] | [**force-version** *value*] | [**priority** *value*]

The **forward-delay***value* parameter specifies how long a port waits before it forwards an RST BPDU after a topology change. Possible values: 4 - 30 seconds. The default is 15 seconds.

The **hello-time** *value* parameter specifies the interval between two hello packets. Possible values: 1 - 10 seconds. The default is 2 seconds.

The **max-age***value* parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. Possible values: 6 - 40 seconds. The default is 20 seconds.

The value of **max-age** must be greater than the value of **forward-delay** to ensure that the downstream bridges do not age out faster than the upstream bridges (those bridges that are closer to the root bridge).

The **force-version***value* parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following values:

- 0 - The STP compatibility mode. Only STP (or legacy) BPDUs will be sent.
- 2 - The default. RST BPDUs will be sent unless a legacy bridge is detected. If a legacy bridge is detected, STP BPDUs will be sent instead.

The **priority***value* parameter specifies the priority of the bridge. You can enter a value from 0 - 65535. A lower numerical value means a the bridge has a higher priority. Thus, the highest priority is 0. The default is 32768.

You can specify some or all of these parameters on the same command line.

Changing port parameters

The RSTP port commands can be enabled on individual ports or on multiple ports, such as all ports that belong to a VLAN.

The RSTP port parameters are preconfigured with default values. If the default parameters meet your network requirements, no other action is required.

You can change the following RSTP port parameters using the following methods.

```
device(config)# vlan 10
device(config-vlan-10)# rstp ethernet 1/5 path-cost 15 priority 64
```

At the VLAN configuration level of the CLI:

Syntax: rstp ethernet slot/portnum path-cost value | **priority value** | [**admin-edge-port**] | [**admin-pt2pt-mac**] | [**force-migration-check**]

At the interface level of the CLI:

Syntax: rstp [**admin-edge-port**] | [**admin-pt2pt-mac**]

The **ethernet** *slot/portnum* parameter specifies the interface used.

The **path-cost***value* parameter specifies the cost of the port's path to the root bridge. RSTP prefers the path with the lowest cost. You can specify a value from 1 - 20,000,000. Table 40 shows the recommended path cost values from the IEEE standards.

TABLE 40 Recommended path cost values of RSTP

Link speed	Recommended (default) RSTP path cost values	Recommended RSTP path cost range
Less than 100 kilobits per second	200,000,000	20,000,000 - 200,000,000
1 Megabit per second	20,000,000	2,000,000 - 200,000,000
10 Megabits per second	2,000,000	200,000 - 200,000,000
100 Megabits per second	200,000	20,000 - 200,000,000
1 Gigabit per second	20,000	2,000 - 200,000,000
10 Gigabits per second	2,000	200 - 20,000
100 Gigabits per second	200	20 - 2,000
1 Terabits per second	20	2 - 200
10 Terabits per second	2	1 - 20

The **priority***value* parameter specifies the preference that RSTP gives to this port relative to other ports for forwarding traffic out of the topology. You can specify a value from 0 - 240, in increments of 16. If you enter a value that is not divisible by four, the software rounds to the nearest value that is divisible by four. The default is 128. A higher numerical value means a lower priority; thus, the highest priority is 8.

Set the **admin-edge-port** to enabled or disabled. If set to enabled, then the port becomes an edge port in the domain.

Set the **admin-pt2pt-mac** to enabled or disabled. If set to enabled, then a port is connected to another port through a point-to-point link. The point-to-point link increases the speed of convergence. This parameter, however, does not auto-detect whether or not the link is a physical point-to-point link.

The **force-migration-check** parameter forces the specified port to sent one RST BPDU. If only STP BPDUs are received in response to the sent RST BPDU, then the port will go return to sending STP BPDUs.

Syslogs for RSTP

By default, syslog messages for RSTP are enabled. To disable syslogs generated by a Topology Change Notice (TCN) for RSTP, enter the following command.

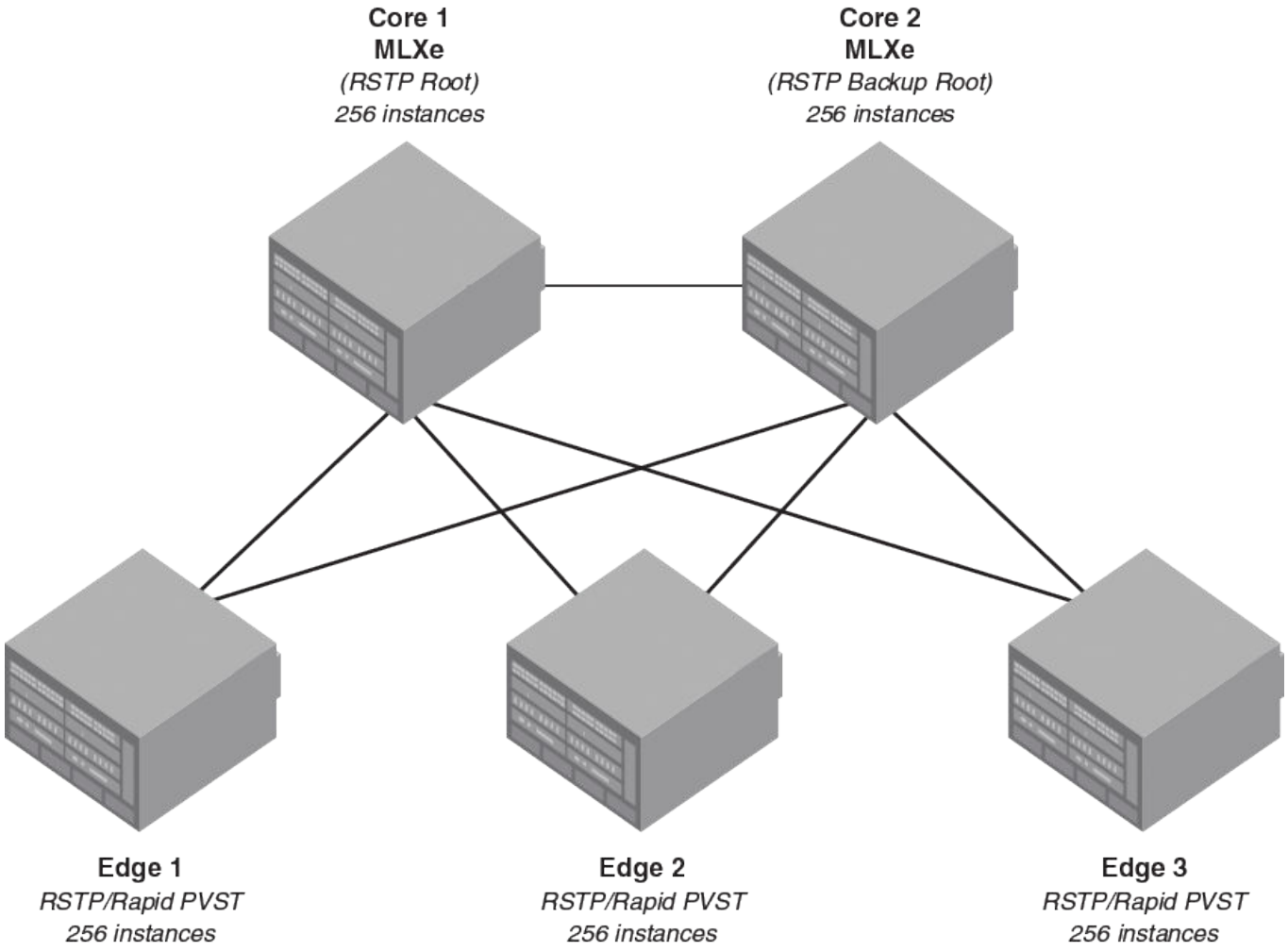
```
device(config)# no logging enable rstp tc-rx
```

Syntax: [no] logging enable rstp tc-rx

RSTP scaling recommendations and best practices

RSTP scaling recommendations and best practices are described in the following sections.

FIGURE 87 RSTP deployment example



Consider the RSTP deployment example as shown in the figure. The topology consists of two MLXe devices installed with MR2 management modules that are deployed as core devices configured with a maximum of 256 RSTP instances. Each core device is connected to multiple edge devices. An edge device can be either MLXe-MR2 device or any vendor switch that is configured with Per VLAN STP/RSTP instances or Cisco® PVST/PVST+/Rapid-PVST/Rapid-PVST+ instances. The Core 1 device is configured as RSTP root node for all VLANs with the lowest RSTP priority. The Core 2 device is configured as RSTP backup root node with the next lowest RSTP topology in the entire topology.

NOTE

RSTP scaling upto 256 instances is supported on NetIron OS 5.9.00 and later software versions.

For best results with 256 RSTP instances on the NetIron OS device, Extreme recommends the following best practices.

- Use only BR-MLX-10GX8-X, NI-MLX-10GX8-M, NI-MLX-10GX8-D, BR-MLX-100GX2-X, BR-MLX-40GX4-M, BR-MLX-100GX1-X, BR-MLX-10GX20-X2, BR-MLX-10GX20-M, BR-MLX-100GX2-CFP2-X2, BR-MLX-100GX2-CFP2-M, BR-MLX-10GX4-IPSEC-M, and later modules for deployment of RSTP scaling. The 48x1G, 24x1G, 20x1G, and lower generation modules are not supported.
- Use RSTP with default timers. Configuring aggressive values for RSTP timers such as 'hello-time' may delay the convergence due to faster protocol timeouts.
- Configure a maximum of 256 VLANs with RSTP with maximum of 128 ports in a VLAN and a maximum of 256 VLANs for each port.

  NOTE
  RSTP may not work with 256 VLANs with 128 ports configured on each VLAN due to limit on the system resources such as memory, message queues and others. For achieving the desired effective virtual ports, topology groups are recommended to be used with RSTP.

- Configure the **wait-for-all-cards** command at the global configuration mode on the core nodes so that the UP port events are received only when all the line cards are booted up. This configuration helps in avoiding unnecessary protocol convergence and temporary loops, during node reload scenarios, due to line cards booting up in a different order.
- Configure the **rstp admin-pt2pt-mac** command for RSTP enabled non-edge ports used for interconnecting nodes which enables rapid forwarding for the ports and avoids extra MAC address flushes.
- Configure the **rstp admin-edge** command for RSTP edge ports connected to the destination nodes such as traffic generators, PCs, or VoIP devices for rapid forwarding of these ports.
- Configure BUM (Broadcast, Unknown Unicast and Multicast) rate limiting on all RSTP non-edge ports such that only 2000 to 4000 packets could reach the CPU. This avoids too many BUM packets hitting on CPU making it available for protocol convergence.
- Configure the **rl-cpu-copy** command on all RSTP non-edge ports to limit the number of packets reaching CPU for source address learning. This helps in lesser traffic loss during MAC address flush operation during RSTP convergence.
- Configure the ports of the interconnected links between core nodes in a multi-slot LAG. These ports should belong to a separate line card than the ports connected to the edge nodes. This helps the BPDUs from the root node to reach the backup root node faster for better convergence than being queued with the BPDUs from the edge nodes.
- Configure multiple links between any two nodes (core to core or core to edge) in a single LAG for load balancing and faster convergence. Any redundant links configured between two nodes cause delay in convergence.
- Configure the lowest port path cost for the link between the two core devices such that this link always remains in the forwarding state even if any of the edge node with lower RSTP bridge priority tries to act as the root bridge.
- For better RSTP convergence, configuring loop detection is not recommended when there are a large number of VLANS or VLAN groups.
- Configuring the spanning-tree root-protect command on the RSTP non-edge ports is not recommended as this may delay convergence in a few deployments.
- Ports of different bandwidths should not be configured and deployed in a LAG.
- Configure bidirectional forwarding detection (BFD) with Tx and Rx interval of 250 ms timeout with multiplier 3 with maximum of 40 sessions per line card and maximum of 250 sessions across the system. This configuration helps to avoid BFD flaps when RSTP is converged and stable.
- Configure LACP with default (long) timers. Short timers may increase the convergence time due to LACP flaps.
- Configuring STP or MSTP instances with more than 128 instances of RSTP is not supported.
- Configuring more than 128 instances of RSTP over MCT VLANs is not supported.

- Configuring MRP or ERP or VSRP, or OAM protocols such as CFM or UDLD along with more than 128 instances of the RSTP is not supported.

Displaying RSTP information

You can display a summary or details of the RSTP information.

To display a summary of RSTP, use the following command.

```
device(config)#show rstp vlan 10
VLAN 10 - RSTP instance 0
----------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:
Bridge           Bridge Bridge Bridge Force    tx
Identifier       MaxAge Hello  FwdDly Version  Hold
hex              sec    sec    sec             cnt
0001000480a04000 20     2      15     Default  3
RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
0001000480a04000 0         0001000480a04000 Root  20  2   15
RSTP (IEEE 802.1w) Port Parameters:
      <--- Config Params -->|<-------------- Current state ----------------->
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/3   128 20000     T   F    DISABLED   DISABLED   0         0000000000000000
1/13  128 20000     T   F    DISABLED   DISABLED   0         0000000000000000
```

NOTE

After deploying or undeploying an MCT cluster, the syslog message for the final state change of the RSTP instance is not correctly updated and displayed on the console.

To display a summary of ports blocked by RSTP, use the following command.

```
device# show rstp blocked vlan 20

VLAN 20 - RSTP instance 0
----------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:

Bridge           Bridge Bridge Bridge Force    tx
Identifier       MaxAge Hello  FwdDly Version  Hold
hex              sec    sec    sec             cnt
80000024389e2d20 20     2      15     Default  3

RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
80000024388f6b20 2000      80000024388f6b20 3/5   20  2   15

RSTP (IEEE 802.1w) Port Parameters:

      <--- Config Params -->|<-------------- Current state ----------------->
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/6   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b20
3/7   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b20
3/8   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b20
```

Syntax: show rstp [**blocked**] [**vlan vlan-id**]

The **blocked** parameter displays blocked ports only, for VLANs enabled with RSTP. When the blocked parameter is not specified, all RSTP port states are displayed.

The **vlan***vlan-id* parameter displays RSTP information for the specified port-based VLAN.

The **show RSTP display** command shows the information listed in Table 41.

TABLE 41 CLI display of RSTP summary

This field...	Displays...
VLAN ID	The port-based VLAN that owns the STP instance and the number of RSTP instances on that VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all RSTP information is for VLAN 1.
Bridge IEEE RSTP Parameters	
Bridge Identifier	The ID of the bridge.
Bridge Max Age	The configured max age for this bridge. The default is 20.
Bridge Hello	The configured hello time for this bridge.The default is 2.
Bridge FwdDly	The configured forward delay time for this bridge. The default is 15.
Force-Version	The configured force version value. One of the following value is displayed: • 0 - The bridge has been forced to operate in an STP compatibility mode. • 2 - The bridge has been forced to operate in an RSTP mode. (This is the default.)
txHoldCnt	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Root Bridge Parameters:	
Root Bridge Identifier	ID of the Root bridge that is associated with this bridge
Root Path Cost	The cost to reach the root bridge from this bridge. If the bridge is the root bridge, then this parameter shows a value of zero.
Designated Bridge Identifier	The bridge from where the root information was received.It can be from the root bridge itself, but it could also be from another bridge.
Root Port	The port on which the root information was received. This is the port that is connected to the Designated Bridge.
Max Age	The max age is derived from the Root port. An RSTP-enabled bridge uses this value, along with the hello and message age parameters to compute the effective age of an RST BPDU. The message age parameter is generated by the Designated port and transmitted in the RST BPDU. RST BPDUs transmitted by a Designated port of the root bridge contains a message value of zero. Effective age is the amount of time the Root port, Alternate port, or Backup port retains the information it received from its peer Designated port. Effective age is reset every time a port receives an RST BPDU from its Designated port. If a Root port does not receive an RST BPDU from its peer Designated port for a duration more than the effective age, the Root port ages out the existing information and recomputes the topology. If the port is operating in 802.1D compatible mode, then max age functionality is the same as in 802.1D (STP).
Hello	The hello value derived from the Root port. It is the number of seconds between two Hello packets.
Fwd Dly	The number of seconds a non-edge Designated port waits until it can apply any of the following transitions, if the RST BPDU it receives does not have an agreed flag: • Discarding state to learning state • Learning state to forwarding state

TABLE 41 CLI display of RSTP summary (continued)

This field...	Displays...
	When a non-edge port receives the RST BPDU it goes into forwarding state within 4 seconds or after two hello timers expire on the port. Fwd Dly is also the number of seconds that a Root port waits for an RST BPDU with a proposal flag before it applies the state transitions listed above. If the port is operating in 802.1D compatible mode, then forward delay functionality is the same as in 802.1D (STP).
RSTP (IEEE 802.1W) Port Parameters	
Port Num	The port number shown in a slot#/port# format.
Pri	The configured priority of the port. The default is 128 or 0x80.
Port Path Cost	The configured path cost on a link connected to this port.
P2P Mac	Indicates if the point-to-point-mac parameter is configured to be a point-to-point link: • T - The link is configured as a point-to-point link. • F - The link is not configured as a point-to-point link. This is the default.
Edge port	Indicates if the port is configured as an operational Edge port: • T - The port is configured as an Edge port. • F - The port is not configured as an Edge port. This is the default.
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to Bridges and bridge port roles on page 281 for definitions of the roles.
State	The port's current RSTP state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled Refer to Bridge port states on page 285 and Edge port and non-Edge port states on page 286.
Designated Cost	The best root path cost that this port received, including the best root path cost that it can transmit.
Designated Bridge	The ID of the bridge that sent the best RST BPDU that was received on this port.

To display detailed information about RSTP, using the following command.

```
device(config)#show rstp detail
VLAN 10 - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:
BridgeId 0001000480a04000, RootBridgeId 0001000480a04000
Control ports - ethernet 1/3 ethernet 1/13
```

```
ForceVersion 2, MigrateTime 3, TxHoldCount 3
RSTP (IEEE 802.1w) Port Parameters:
Port 1/3 - Role: DISABLED - State: DISABLED
Port 1/13 - Role: DISABLED - State: DISABLED
```

Syntax: show rstp detail [**vlan vlan-id**]

The **vlan***vlan-id* parameter displays RSTP information for the specified port-based VLAN.

The **show RSTP detail** command shows the following information.

This field...	Displays...
VLAN ID	ID of the VLAN that owns the instance of RSTP and the number of RSTP instances on that VLAN.
Bridge ID	ID of the bridge.
Control ports	Ports assigned to the VLAN
forceVersion	the configured version of the bridge: • 0 - The bridge has been forced to operate in an STP compatible mode. • 2 - The bridge has been forced to operate in an RSTP mode.
MigrateTime	The number of seconds the bridge took to migrate from STP to RSTP mode.
txHoldCount	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Port	ID of the port in slot#/port# format.
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to Bridges and bridge port roles on page 281 for definitions of the roles.
State	The port's current RSTP state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled Refer to Bridge port states on page 285 and Edge port and non-Edge port states on page 286.

Configuring RSTP under an ESI VLAN

RSTP can also be configured under a VLAN that is part of a user-configured ESI. For example, to enable RSTP on a VLAN that is part of an ESI, configure the following commands.

```
device(config)# esi customer1 encapsulation cvlan
device(config-esi-customer1)# vlan 100
device(config-esi-customer1-vlan-100)# rstp
```

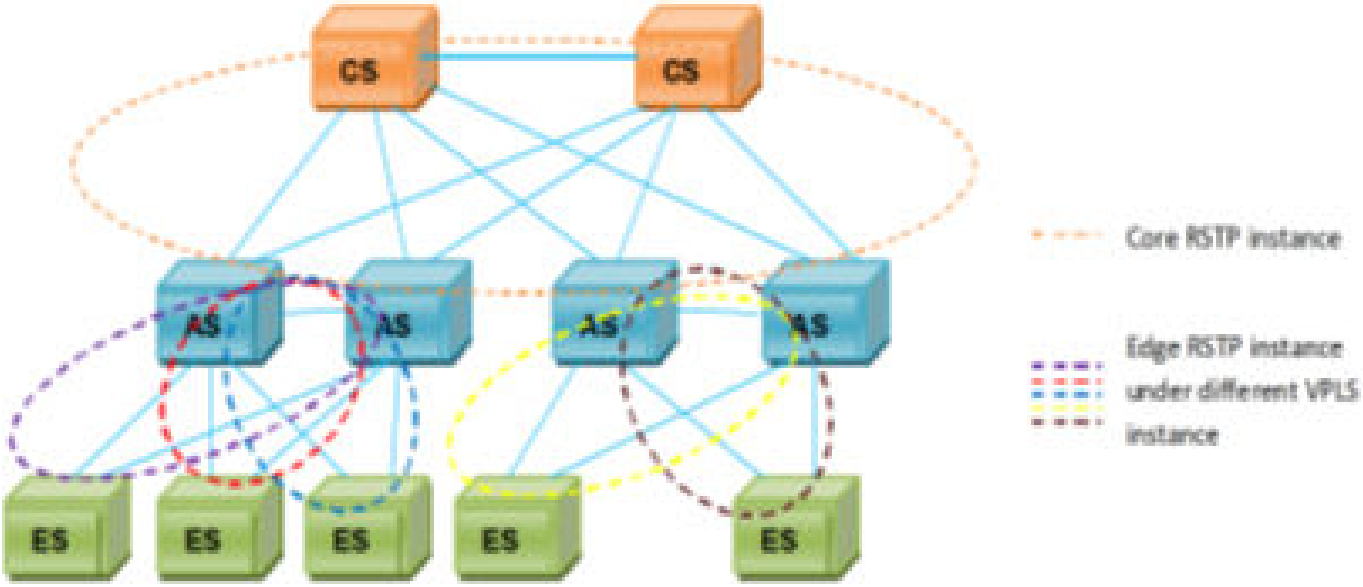
RSTP support for PB and PBB

A PBB network is comprised of a set of Backbone Core Bridges (BCBs) and Backbone Edge Bridges (BEBs). BEBs are interconnected by some or all of the S-VLANs supported by a PB network. Each BEB provides interfaces that encapsulate customer frames, thus allowing customer MAC addresses (C-MAC) and VLANs (S-VLAN) to be independent of backbone MAC addresses (B-MAC) and VLANs (B-VLAN) used to relay those frames across the backbone.

In CES 2000 Series and CER 2000 Series, RSTP over PBB is supported in previous releases implemented using the ESI framework.

Consider a scenario where all BEBs (AS - Access Switch) are connected to two BCBs (CS - Core Switch) to provide dual home for all the BEBs, and all PBs (ES - Edge Switch) are connected to two BEBs to provide dual home for all the PBs in the network. With the dual homing of the ASs to the CSs and ESs to the ASs, all failures are protected. This dual homing support creates potential loops in a PB network and PBB network. To achieve the functionality of dual homing of AS and ES, the active topology of a PB network and PBB network area should be isolated by running different instance of RSTP.

FIGURE 88 RSTP isolation in dual homing of AS and ES



The network shown in Figure 89 has two Core Switches (CSs) to provide resiliency in the core. The CS functions as a Backbone Core Bridge (BCB). All Access Switches (AS) in the network dual home to the two CSs. The AS functions as a Backbone Edge Bridge (BEB). The CSs do not have any service interfaces. The function of the CS is to switch traffic between the ASs. As a BCB, a CS will switch on the outer PBB B-tag and will not perform any PBB encapsulation.

The AS and CS switches in the network will form a single RSTP region. With the dual homing of the ASs to the CSs, all failures are protected against. ES dual homing will help to protect the failure of AS.

Core RSTP

RSTP will run in the PBB core for loop detection and avoidance. All ASs and CSs will participate in the RSTP and one CS will be selected as the root for the core RSTP instance. The assumption is there will be only one RSTP instance running in the PBB core and traffic flow will be through one CS which is the root bridge.

A backbone service provider can either use RSTP or MSTP. Preferably MSTP shall be used, since this will allow the service provider to use different active paths for different B-VLANs.

To avoid a potential loop in the core PBB network, RSTP will be enabled. The regular VLAN corresponds to VPLS B-VLAN in AS/BEB bridges.

Edge RSTP

Dual homing ES to two ASs would require RSTP to avoid loops. An AS will have several RSTP instances provisioned on the ES facing side. Here one AS will be acting as the root and the Edge RSTP is completely separate and independent from the Core RSTP.

The Edge RSTP instance should be enabled on the VPLS instance on AS/ES.

For dual homing of the ES, the ASs can be connected using either of the following two methods.

1. Two AS/BEB switches connected via the S-tagged endpoint.

2. Two AS/BEB switches connected via the IB-tagged endpoint.

   In each case, the RSTP behavior is different, which is explained below.

   Figure 90 shows that BEB-1, BEB-2 and PB could be of same S-VLAN or different S-VLANs. Since RSTP is enabled under the VPLS instance, all the VPLS VLANs which belong to that VPLS instance will be considered as part of same RSTP instance. BPDUs will be transmitted on all the VPLS endpoints with the VLAN tag associated with that end point.

   FIGURE 89 RSTP convergence when two AS connected via S tagged endpoint

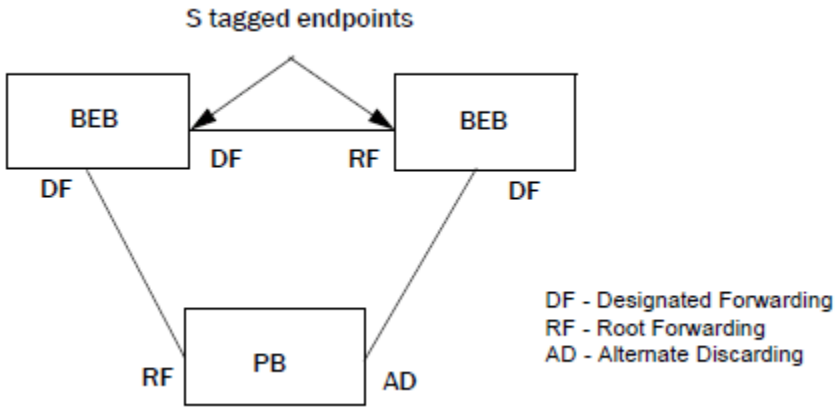
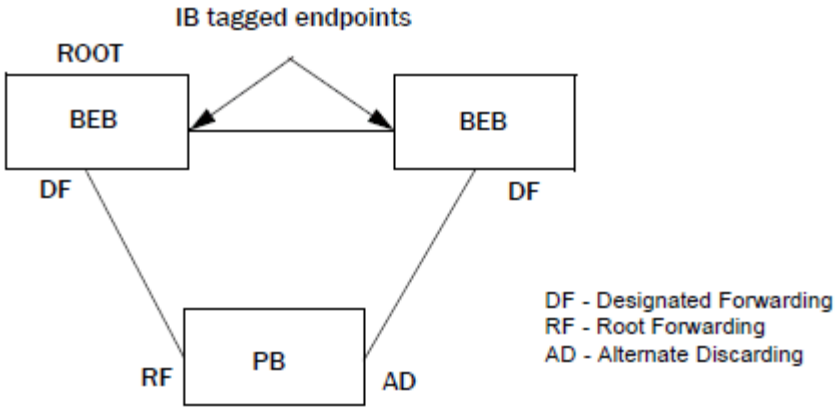


   Figure 91 shows that BEB-1 and BEB-2 are connected via an IB tagged endpoint. The topology convergence is the same as in Figure 90. The difference comes in the BPDU transmitted out of the IB tagged endpoint that will be a tunneled packet which will have a PBB header. In this topology the IB- tagged end point should be always in the forwarding state by configuring either BEB-1 or BEB-2 as a ROOT bridge.

   FIGURE 90 RSTP convergence when two AS connected via IB tagged endpoint



BPDU behavior on VPLS endpoints

1. By default, the BPDU generated from the VPLS end point will be with destination MAC of 00-00-00-00-00-08.

2. Upon receiving a BPDU with destination MAC 00-00-00-00-00-00, the RSTP will start responding with that MAC on next transmitted BPDUs. (This is applicable for the VPLS VLAN acting as a C-VLAN or S-VLAN.)
3. The VPLS B-VLAN end point will not respond for a tunneled BPDU with the STP MAC 00-00-00-00-00-00. It will be considered as a tunneled BPDU from customer bridges and be tunneled across S-VLAN as well as C-VLAN.
4. The VPLS B-VLAN end point will be consuming the BPDU. Only when RSTP is enabled on the IB tagged end point and received tunneled BPDU, it has a PBB-STP MAC. If RSTP is not enabled, the received BPDU with PBB-STP MAC will be send to S-VLAN

Limitations

- Total RSTP instances is limited to 128, including regular VLAN and VPLS instances.
- RSTP Single is not supported in VPLS VLAN.
- The is no MIB support for PBB RSTP.
- Do not use the same VLAN ID for a regular VLAN and VPLS instance in the network. Enabling RSTP on a regular VLAN and VPLS instance which has a VPLS VLAN with the same VLAN ID as the regular VLAN ID can lead to an undesirable topology convergence.
- It is not possible to enable RSTP on a PBB VPLS instance if that instance has multiple VLANS configured with same member ports.
- RSTP interoperability between MLX Series and CER 2000 Series devices is not supported if both are acting as a BEB.
- Running RSTP on a VPLS Instance will not avoid the pure Layer 2 forwarding loop created by the regular B-VLAN in the BEBs. Run RSTP on regular B-VLAN in BEBs.
- Switchover and hitless upgrade are not supported on PBB RSTP.
- When changing the RSTP parameters on a regular B-VLAN, the parameters also need to be changed on the VPLS instance. The expectation is to have the same port states for B-VLAN (regular) end points and VPLS ISID end points.

Configuration commands

Use the following commands to configure RSTP on a PBB VPLS instance.

Syntax: [no] rstp [**forward-delay** *value*] | [**hello-time** *value*] | [**max-age** *time*] | [**force-version** *value*] | [**priority** *value*]

The **forward-delay** parameter specifies how long a port waits before it forwards an RST.

The **hello-time** parameter specifies the interval between two hello packets. The values range from 1 to 10 seconds. The default is 2 seconds.

The **max-age** parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. Acceptable values range from 6 to 40 seconds. The default is 20 seconds. The value of **max-age** must be greater than the value of **forward-delay** to ensure that the downstream bridges do not age out faster than the upstream bridges.

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify either of the following values:

0 - The STP compatibility mode. Only STP (or legacy) BPDUs will be sent.

2 - The default. RST BPDUs will be sent unless a legacy bridge is detected. If a legacy bridge is detected, STP BPDUs will be sent instead.

The **priority** parameter specifies the priority of the bridge. You can enter a value from 0 to 65535. A lower numerical value means a the bridge has a higher priority. Thus, the highest priority is 0. The default is 32768.

The **[no]** version disables the feature and returns settings to default.

Syntax: [no] rstp etherne t *slot/portnum* **path-cost** *value* | **priority** *value* | [**admin-edge-port**] | [**admin-pt2pt-mac**] | [**force-migration-check**]

The **ethernet***slot/portnum* parameter specifies the interface used.

The **path-cost** parameter specifies the cost of the port's path to the root bridge. RSTP prefers the path with the lowest cost. You can specify a value from 1 to 20,000,000.

The **priority***value* parameter specifies the preference that RSTP gives to this port relative to other ports for forwarding traffic out of the topology. You can specify a value from 0 to 240, in increments of 16. If you enter a value that is not divisible by four, the software rounds to the nearest value that is divisible by four. The default is 128. A higher numerical value means a lower priority; thus, the highest priority is 8.

Set the **admin-edge-port** to enabled or disabled. If set to enabled, then the port becomes an edge port in the domain.

Set the **admin-pt2pt-mac** to enabled or disabled. If set to enabled, then a port is connected to another port through a point-to-point link. The point-to-point link increases the speed of convergence. This parameter, however, does not auto-detect whether or not the link is a physical point-to-point link.

The **force-migration-check** parameter forces the specified port to sent one RST BPDU. If only STP BPDUs are received in response to the sent RST BPDU, then the port will go return to sending STP BPDUs.

Show commands

The **show rstp** command output displays VPLS instance ID if RSTP is running in VPLS VLAN.

device(config)# show rstp

Syntax: show rstp [**vlan** *vlan-id*] [**vpls id** *vpls-id*]

```
device(config)# show rstp
VPLS Instance ID 1  - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:
Bridge             Bridge Bridge Bridge Force   tx
Identifier        MaxAge Hello  FwdDly Version Hold
hex               sec    sec    sec            cnt
0001000480a04000  20     2      15     Default 3
RootBridge        RootPath DesignatedBridge Root Max  Hel Fwd
Identifier       Cost     Identifier       Port Age  lo  Dly
hex            hex                          sec  sec sec
0001000480a04000 0        0001000480a04000 Root 20   2   15
RSTP (IEEE 802.1w) Port Parameters:
<--- Config Params -->|<-------------- Current state ----------------->
Port Pri  PortPath P2P Edge Role     State    Designa- Designated
Num       Cost     Mac Port                   tedcost  bridge
1/3  128  20000    T   F    DISABLED DISABLED 0        0000000000000000
1/13 128  20000    T   F    DISABLED DISABLED 0        0000000000000000
```

The **show rstp detail** command displays VPLS instance ID if RSTP is running in VPLS VLAN.

```
device(config)# show rstp detail vpls id 1
```

Syntax: show rstp detail [**vlan** *vlan-id*] [**vpls id** *vpls-id*]

```
NetIron(config)#show rstp detail
VPLS Instance ID - 1  - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:
BridgeId 0001000480a04000, RootBridgeId 0001000480a04000
Control ports - ethernet 1/3 ethernet 1/13
ForceVersion 2, MigrateTime 3, TxHoldCount 3
RSTP (IEEE 802.1w) Port Parameters:
Port 1/3 - Role: DISABLED - State: DISABLED
Port 1/13 - Role: DISABLED - State: DISABLED
```

The **show mpls vpls detail** command displays the VPLS instance ID if RSTP is running in VPLS VLAN.

Syntax: show mpls vpls detail

```
NetIron #show mpls vpls id 1
VPLS as, Id 1, Max mac entries: 2048
 PBB
   Bridge Destination MAC Address: None (NHT Index: 0)
 Total vlans: 2, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
 IFL-ID: n/a
 Enabled L2 Protocol:RSTP
 Vlan 100: Topo ID 1
   Tagged: ethe 1/1

   Port    Protocol  State
   1/1     RSTP      DISABLED
   1/2     RSTP      FORWARDING
  Vlan 200
   Tagged: ethe 1/1
      CPU-Protection: OFF
   Local Switching: Enabled
   Extended Counter: ON
```

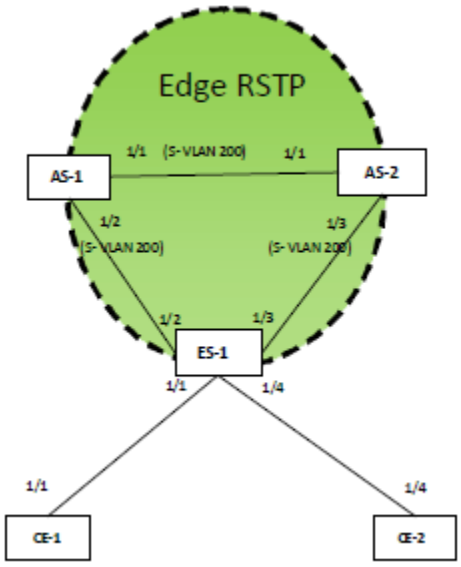
Use case scenarios

Use case 1: Edge RSTP - AS-1 is connected to AS-2 and ES-1 via S-tagged endpoints of same S-VLAN

The following deployment scenario is a case where RSTP is deployed for a single S-VLAN in a PB network. VPLS VLAN 200 (S-VLAN) is responsible for carrying traffic to the PB network. In this case, AS-1 is configured as ROOT Bridge, VPLS VLAN 200 is configured on AS-1, AS-2, and ES-1 which connects the three bridges together. RSTP is running on the VPLS Instance on AS-1, AS-2 and ES-1.

The following describes the steps to configure the nodes in the topology.

FIGURE 91 Edge RSTP topology 1



Configuring AS-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/2
```

RSTP Configuration on vpls instance 1

```
device_AS-1(config-mpls-vpls-pb-svlan)#rstp
device_AS-1(config-mpls-vpls-pb-svlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/3
```

RSTP Configuration on vpls instance 1

```
device_AS-2 (config-mpls-vpls-pb-svlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 200
device_ES-1(config-mpls-vpls-pb-vlan-vlan-200)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 300 on the customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag eth 1/1 eth 1/4
```

RSTP Configuration on VPLS instance

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 300 and add port 1/1 to it.

```
device_CE-1(config)#vlan 300
device_CE-1(config-vlan-300)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 300 and add port 1/4 to it.

```
device_CE-2(config)#vlan 300
device_CE-2(config-vlan-300)#tagged ethernet 1/4
```

If RSTP needs to be enabled in CE bridges, the following configuration should be applied.

Configuring CE-1 and CE-2

RSTP Configuration

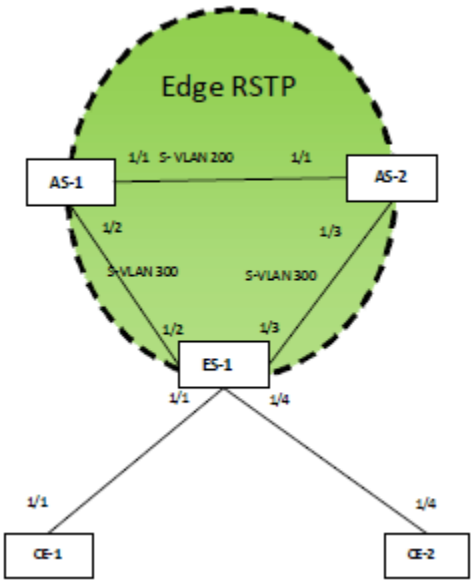
```
device_CE-1(config)#vlan 300
device_CE-1(config-vlan-300)#rstp
```

Use case 2: Edge RSTP - AS-1 is connected to AS-2 and ES-1 via S-tagged endpoints of different S-VLAN

The following deployment scenario is a case where RSTP is deployed on two different S-VLANs of the same VPLS instance in a PB network. Here AS-1 is configured as a ROOT Bridge and RSTP is running on the VPLS Instance on AS-1, AS-2, and ES-1. VPLS VLAN 200 configured on AS-1 and AS-2 acts as an S-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 configured on AS-1 and AS-2 acts as another S-VLAN which connects AS-1 and AS-2 to the ES-1. AS-1 and AS-2 has the VPLS VLAN 200 and 300 configured under same VPLS instance. VPLS VLAN 300 (S-VLAN) is configured ES-1 connects to AS-1 and AS-2.

The following discussion describes procedure required to configure the nodes in the topology.

FIGURE 92 Edge RSTP topology 2



Configuring AS-1

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-svlan)#rstp
device_AS-1(config-mpls-vpls-pb-svlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
```

```
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-2(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-svlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure the C-VLAN 400 on customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 400
device_ES-1(config-mpls-vpls-pb-vlan-vlan-400)#tag eth 1/1 eth 1/4
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

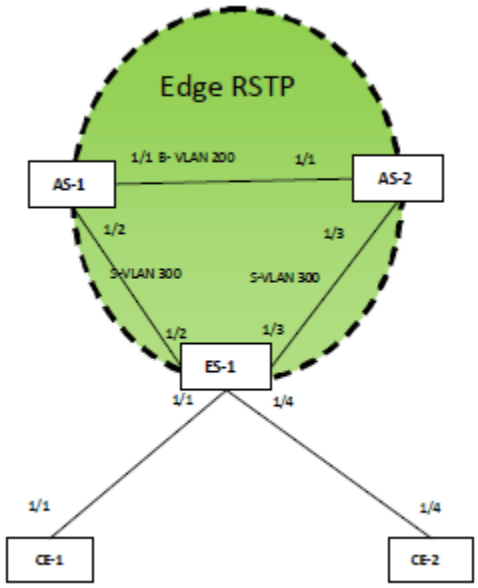
Use case 3: Edge RSTP - AS-1 is connected to AS-2 via IB-tagged endpoint and both the AS on ES facing side with same S-VLAN

The following deployment scenario is a case where RSTP is deployed on a VPLS instance which has a S-VLAN configured to the ES facing side and B-VLAN configured which connects 2 ASs.

AS-1 is configured as a ROOT Bridge and RSTP is running on the VPLS Instance on AS-1, AS-2 and ES-1. VPLS VLAN 200 is configured in AS-1 and AS-2 acts as B-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 is configured in AS-1 and AS-2 acts as S-VLAN which connects AS-1 and AS-2 to the ES-1. In AS-1 and AS-2 VPLS VLAN 200 and 300 are configured under the same VPLS instance. VPLS VLAN 300 (S-VLAN) is configured on ES-1, which connects to AS-1 and AS-2.

The following discussion describes how to configure the nodes in the topology.

FIGURE 93 Edge RSTP topology 3



Configuring AS-1

Tag type configuration

Configure port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
```

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/2
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 200
device_AS-1(config-vlan-200)#tagged ethernet 1/1
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-vlan 1
device_AS-1(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-1(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan)#rstp
device_AS-1(config-mpls-vpls-pb-vlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure the port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
:
```

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/3
```

B-VLAN Configuration

```
device_AS-2(config)#vlan 200
device_AS-2(config-vlan-200)#tagged ethernet 1/1
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pbb-vlan 1
device_AS-2(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-2(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_AS-2(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/3
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
```

```
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 400 on a customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 400
device_ES-1(config-mpls-vpls-pb-vlan-vlan-400)#tag eth 1/1 eth 1/4
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

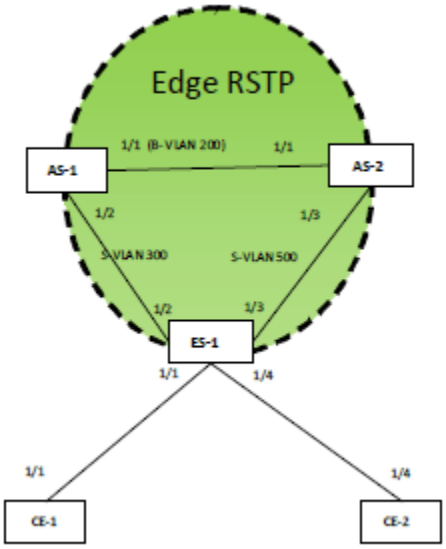
Use case 4: Edge RSTP - AS-1 is connected to AS-2 via IB-tagged endpoint and both the AS on ES facing side with different S-VLAN

The following deployment scenario is a case where RSTP is deployed on a VPLS instance which has a S-VLAN configured to the ES facing side and B-VLAN configured which connects 2 ASs.

AS-1 is configured as a ROOT Bridge. VPLS VLAN 200 is configured in AS-1 and AS-2 acts as B-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 is configured in AS-1 connects AS-1 to ES-1 and VPLS VLSN 500 on AS-2 connects AS-2 to ES-1. In AS-1 VPLS VLAN 200 and 300 are configured under same VPLS instance and AS-2 VPLS VLAN 200 and 500 are configured under the same VPLS Instance. VPLS VLAN 300 (S-VLAN) configured on ES-1 which connects to AS-1 and VPLS VLAN 500 connects to AS-2.

The following discussion describes how to configure the nodes in the topology.

FIGURE 94 Edge RSTP topology 4



Configuring AS-1

Tag type configuration

Configure the port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
```

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/2
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 200
device_AS-1(config-vlan-200)#tagged ethernet 1/1
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-vlan 1
device_AS-1(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-1(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan)#rstp
device_AS-1(config-mpls-vpls-pb-vlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure a port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
```

Configure a port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/3
```

B-VLAN Configuration

```
device_AS-2(config)#vlan 200
device_AS-2(config-vlan-200)#tagged ethernet 1/1
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pbb-vlan 1
device_AS-2(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-2(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan-pbb)#vlan 500
device_AS-2(config-mpls-vpls-pb-vlan-vlan-500)#tag ethernet 1/3
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure a port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 300.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 500
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 400 on customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 400
device_ES-1(config-mpls-vpls-pb-vlan-vlan-400)#tag eth 1/1 eth 1/4
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

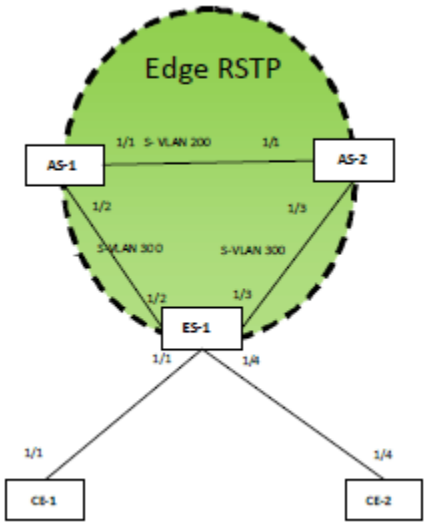
Use case 5: Edge RSTP- Interoperability with CES 2000 Series and CER 2000 Series

In this scenario AS-1 is connected to AS-2 and ES-1 via S-tagged endpoint of different S-VLAN, and ES-1 is a CES 2000 Series and CER 2000 Series device.

The following scenario is a case where RSTP is deployed on two different S-VLANs of the same VPLS instance in a PB network. AS-1 is configured as a ROOT Bridge and RSTP is running on the VPLS Instance on AS-1, AS-2, and ES-1. VPLS VLAN 200 is configured on AS-1 and AS-2 acts as a S-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 is configured on AS-1 and AS-2 acts as another S-VLAN which connects AS-1 and AS-2 to the ES-1. AS-1 and AS-2 has VPLS VLAN 200 and 300 configured under same VPLS instance. VPLS VLAN 300 (S-VLAN) is configured with ES-1 and connects to AS-1 and AS-2.

The following discussion describes how to configure the nodes in the topology.

FIGURE 95 Edge RSTP topology 5



Configuring AS-1

Tag type configuration

Configure a port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-svlan)#rstp
device_AS-1(config-mpls-vpls-pb-svlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure a port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
```

```
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-2(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-svlan)#rstp
```

Configuring ES-1

Port-type configuration

```
device_(config)# interface ethernet 1/2
device_(config-if-e1000-1/2)# port-type provider-network
device_(config-if-e1000-1/2)# enable
device_(config)# interface ethernet 1/3
device_(config-if-e1000-1/3)# port-type provider-network
device_(config-if-e1000-1/3)# enable
```

Port-type configuration

```
device_(config)# interface ethernet 1/1
device_(config-if-e1000-1/1)# port-type customer-edge
device_(config-if-e1000-1/1)# enable
device_(config)# interface ethernet 1/4
device_(config-if-e1000-1/4)# port-type customer-edge
device_(config-if-e1000-1/4)# enable
device_(config)# esi provider encapsulation svlan
device_ (config-esi-provider)# vlan 300
device_(config-esi-provider-vlan-300)# tagged ethernet 1/2
device_ (config-esi-provider-vlan-300)# tagged ethernet 1/3
device_ (config-esi-provider-vlan-300)# exit
device_ (config-esi-provider)# exit
device_(config)# esi customer encapsulation cvlan
device_(config-esi- customer )# vlan 400
device_(config-esi- customer -vlan-400)# tagged ethernet 1/1
device_(config-esi- customer -vlan-400)# tagged ethernet 1/4
device_(config-esi- customer -vlan-400)# exit
device_(config-esi- customer)# exit
```

RSTP Configuration

```
device_(config)# esi customer encapsulation  cvlan
device_(config-esi- customer )# vlan 400
device_(config-esi- customer )#rstp
device_(config)# esi provider encapsulation  svlan
device_(config-esi-provider)# vlan 300
device_(config-esi-provider-vlan-300)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

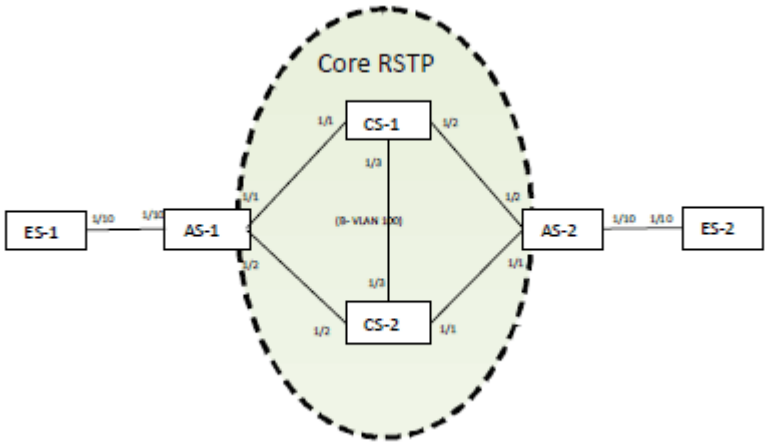
```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

Use case 6: Core RSTP

The following deployment scenario is a case where RSTP is deployed for a single B-VLAN in a PBB network. In CS-1 and CS-2 a regular VLAN 100 is configured as B-VLAN which carries the traffic to the PBB core network. AS-1 and AS-2 have a regular VLAN and VPLS VLAN 100 which carries PBB traffic towards the B-VLAN. CS-1 is configured as the ROOT Bridge and RSTP is running over regular VLAN 100 in CS-1 and CS-2. In AS-1 and AS-2 RSTP runs over the regular VLAN 100 corresponds to the VPLS VLAN B-VLAN 100.

The following discussion describes how to configure the nodes in the topology.

FIGURE 96 Core RSTP in PBB network



AS-1 Configuration

> **NOTE**
> The AS-2 configuration is similar to the AS-1 configuration.

To carry PBB traffic, configure a VPLS instance. The B-VLAN used here is 100. For PB traffic, the S-VLAN used is 200 and C-VLAN 300.

Tag type configuration

```
device_AS-1(config)#tag-type 88e8 eth 1/1
device_AS-1(config)#tag-type 88e8 eth 1/2
device_AS-1(config)#tag-type 9100 eth 1/10
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 100
device_AS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-vlan-100)#rstp
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan 1
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb)#vlan 100 isid 101010
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-bvlan)#vlan 200
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-200)#tagged ethernet 1/10
```

CS-1 Configuration

NOTE
The configuration of CS-2 is similar to the configuration of CS-1, except for the RSTP priority configuration.

Port type configuration

```
device_CS-1(config)#tag-type 88e8 eth 1/1
device_CS-1(config)#tag-type 88e8 eth 1/2
```

B-VLAN Configuration

```
device_CS-1(config)#vlan 100
device_CS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

RSTP Configuration

```
device_CS-1(config-vlan-100)#rstp
device_CS-1(config-vlan-100)#rstp priority 100
```

ES-1 Configuration

NOTE
The configuration of ES-2 is similar to the configuration of ES-1.

Port type configuration

```
device_ES-1(config)#tag-type 9100 eth 1/10
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-svlan 1
device_ES-1(config-mpls-vpls-pb-svlan)#vlan 200
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/10
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-svlan#rstp
```

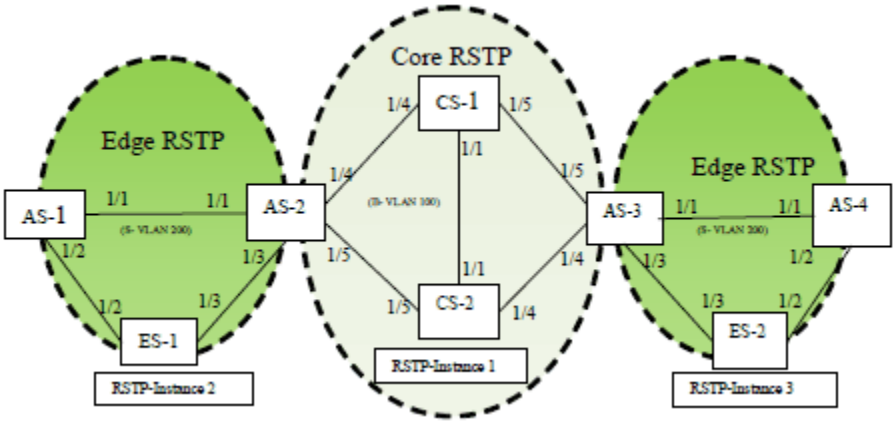
Use case 7: Core and Edge RSTP

RSTP will run in the PBB core for loop detection and avoidance in the B-VLAN. All ASs and CSs in B-VLAN will participate in the same RSTP instance. One CS will act as root for the core RSTP instance. All AS will have a different RSTP instance on the ES facing side and one of the AS will be acting as a ROOT.

The following deployment scenario is a case where RSTP is deployed for a B-VLAN and S-VLAN in a PBB network. In CS-1 and CS-2 a regular VLAN 100 is configured as a B-VLAN. AS-1 and AS-2 has a VPLS VLAN 100 which acts as a B-VLAN. VPLS VLAN 200 on ES-1 and ES-2 which acts as an S-VLAN for the PB network. CS-1 is configured as the ROOT Bridge for core RSTP. AS-1 and AS-4 are the ROOT bridges for RSTP on the ES facing side.

The following discussion describes how to configure the nodes in the topology.

FIGURE 97 RSTP deployment in PB and PBB network



Configuring AS-1

> **NOTE**
> The configuration of AS-4 is similar to the configuration of AS-1.

Tag type configuration

Configure aport tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#rstp
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#rstp priority 100
```

Configuring AS-2

> **NOTE**
> The configuration of AS-3 is similar to the configuration of AS-2.

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
device_AS-2(config)#tag-type 88e8 eth 1/5
device_AS-2(config)#tag-type 88e8 eth 1/6
```

B-VLAN configuration

```
device_AS-1(config)#vlan 100
device_AS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

RSTP configuration

```
device_AS-1(config-vlan-100)#rstp
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/3
device_AS-2(config-mpls-vpls-pb-svlan-vlan-100)#rstp
```

B-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pbb-bvlan-pbb)#vlan 100 isid 101010
device_AS-2(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tag ethernet 1/4 ethernet 1/5
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#rstp
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#rstp
```

Configuring ES-1

NOTE
The configuration of ES-2 is similar to the configuration of ES-1.

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-svlan 1
device_ES-1(config-mpls-vpls-pb-svlan)#pbb
device_ES-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/2 ethernet 1/3
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#rstp
```

CS-1 Configuration:

NOTE

The configuration of CS-2 is similar to the configuration of CS-1, except for the RSTP priority configuration.

Port type configuration

```
device_CS-1(config)#tag-type 88e8 eth 1/1
device_CS-1(config)#tag-type 88e8 eth 1/2
device_CS-1(config)#tag-type 88e8 eth 1/2
```

B-VLAN Configuration

```
device_CS-1(config)#vlan 100
device_CS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/4  ethernet 1/5
```

RSTP Configuration

```
device_CS-1(config-vlan-100)#rstp
device_CS-1(config-vlan-100)#rstp priority 100
```

NOTE

There can be scenarios where the S-VLAN used on ES-1 is different from one used on ES-2. This requires the configuration of different S-VLANs on AS-1 and AS-2.

Metro Ring Protocol

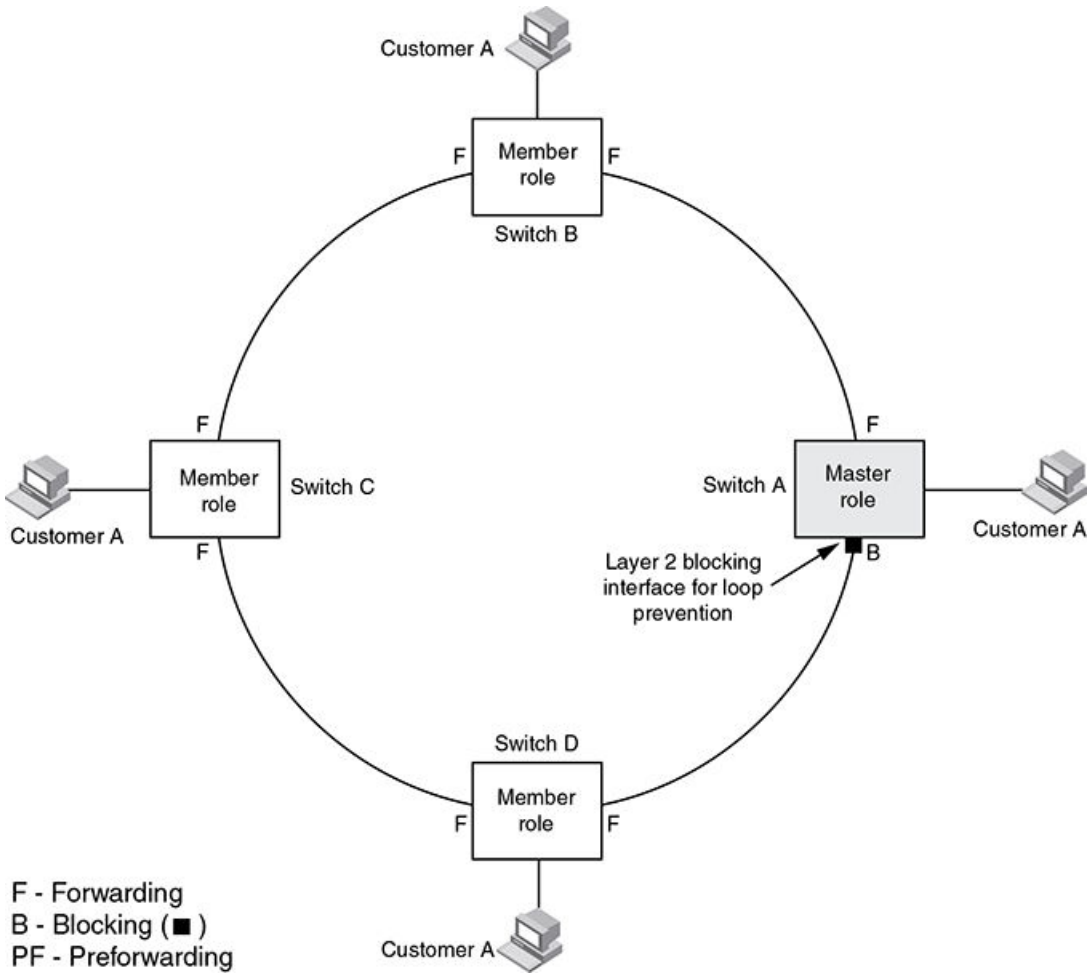
Metro Ring Protocol

Extreme Metro Ring protocol (MRP) is a proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in ring topologies. It is an alternative to Spanning Tree Protocol (STP) and is very useful in Metropolitan Area Networks (MAN) where using 802.1D STP has the following limitations:

- 802.1D recommends a maximum bridge diameter of seven nodes with standard timers. MRP is capable of many more nodes than this.
- 802.1D has a slow reconvergence time that could be seconds or even minutes. MRP can detect and heal a break in the ring in under one second.

Figure 99 shows a simple metro ring.

FIGURE 98 MRP - normal state



The ring in this example consists of four Extreme device nodes that support MRP. Each node has two ring interfaces and the interfaces are all in one port-based VLAN. There are customer networks utilizing the nodes and Layer 2 traffic is forwarded to and from the customer networks through the ring. Each customer interface can be in the same VLAN as the ring or in a separate VLAN under control of MRP as part of a topology group.

For each discrete ring, one node is configured in the master role for the MRP ring. One of the two ring interfaces on the master node is configured as the primary interface, the other is the secondary interface. The primary interface originates Ring Health Packets (RHPs) which are used to monitor the health of the ring. An RHP is forwarded on the ring to the next interface until it reaches the secondary interface of the master node. On receipt of an RHP, the secondary interface transitions into blocking mode to prevent a Layer 2 loop.

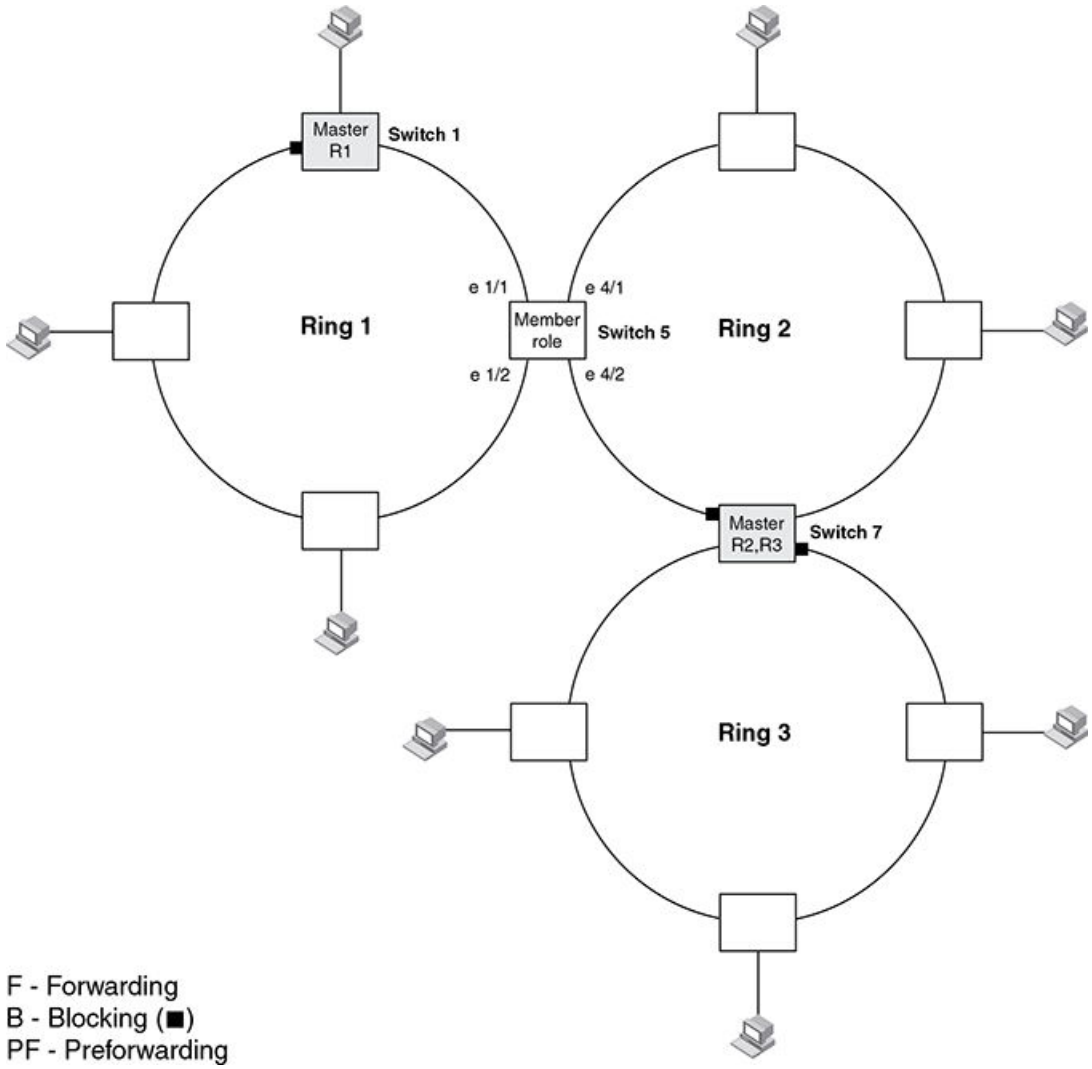
NOTE

When you configure MRP, it is recommended that you disable the secondary ring interface on the master node before beginning or changing the ring configuration. Disabling an interface prevents a Layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once you have completed the MRP configuration and enabled it on all the nodes, you should re-enable the secondary ring interface.

MRP rings without shared interfaces (MRP Phase 1)

MRP Phase 1 allows you to configure multiple MRP rings, as shown in Figure 100, but the rings cannot share the same interfaces. For example, you cannot configure ring 1 and ring 2 to share interfaces ethernet 1/1 and 1/2 on Switch 5. Each ring must remain an independent ring and RHP packets are processed within each ring.

FIGURE 99 MRP without shared interfaces

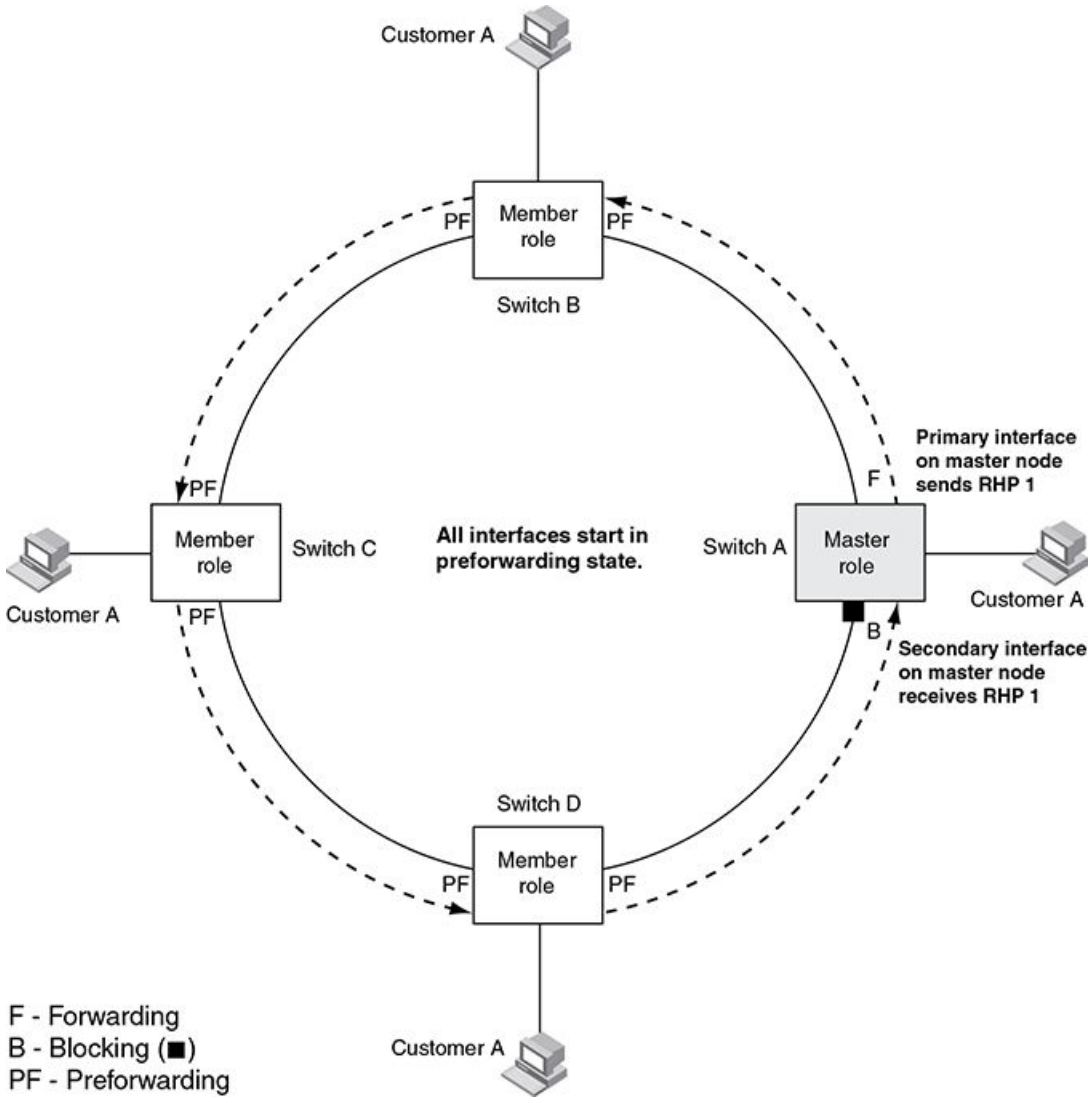


In this example, two nodes are each configured with two MRP rings. Any node in a ring can be the master for its ring. A node also can be the master for more than one ring.

Ring initialization

Figure 101 shows the initial state of the ring, when MRP is first enabled on the ring's switches. On the master the primary interface starts in forwarding mode and the secondary interface starts in blocking mode. All ring interfaces on member nodes begin in the preforwarding state (PF).

FIGURE 100 MRP ring - initial state



An RHP is an MRP protocol packet used to monitor the health of the ring. The source address is the MAC address of the master node and the destination MAC address is a protocol address for MRP. The Master node generates RHPs and sends them on the ring. The state of a ring interface is influenced by the RHPs.

A ring interface can have one of the following MRP states:

- **Preforwarding (PF)** - The interface will forward RHPs and learn MAC addresses but won't forward data for the ring. All ring interfaces start in this state when you enable MRP except the master node. A blocking interface transitions to preforwarding when the preforwarding timer expires and no RHP's have been received.

- **Forwarding (F)** - The interface will forward RHP's and data for the ring. On member switches an interface transitions from preforwarding to forwarding when the preforwarding time expires or the interface receives an RHP with the forwarding bit set. A break in the ring is indicated if the secondary interface on the master fails to receive an RHP within the preforwarding timer and the interface transitions from blocking to forwarding to heal the ring.
- **Blocking (B)** - The interface can process RHPs, but cannot forward data for the ring. Only the secondary interface on the master node can be blocking.

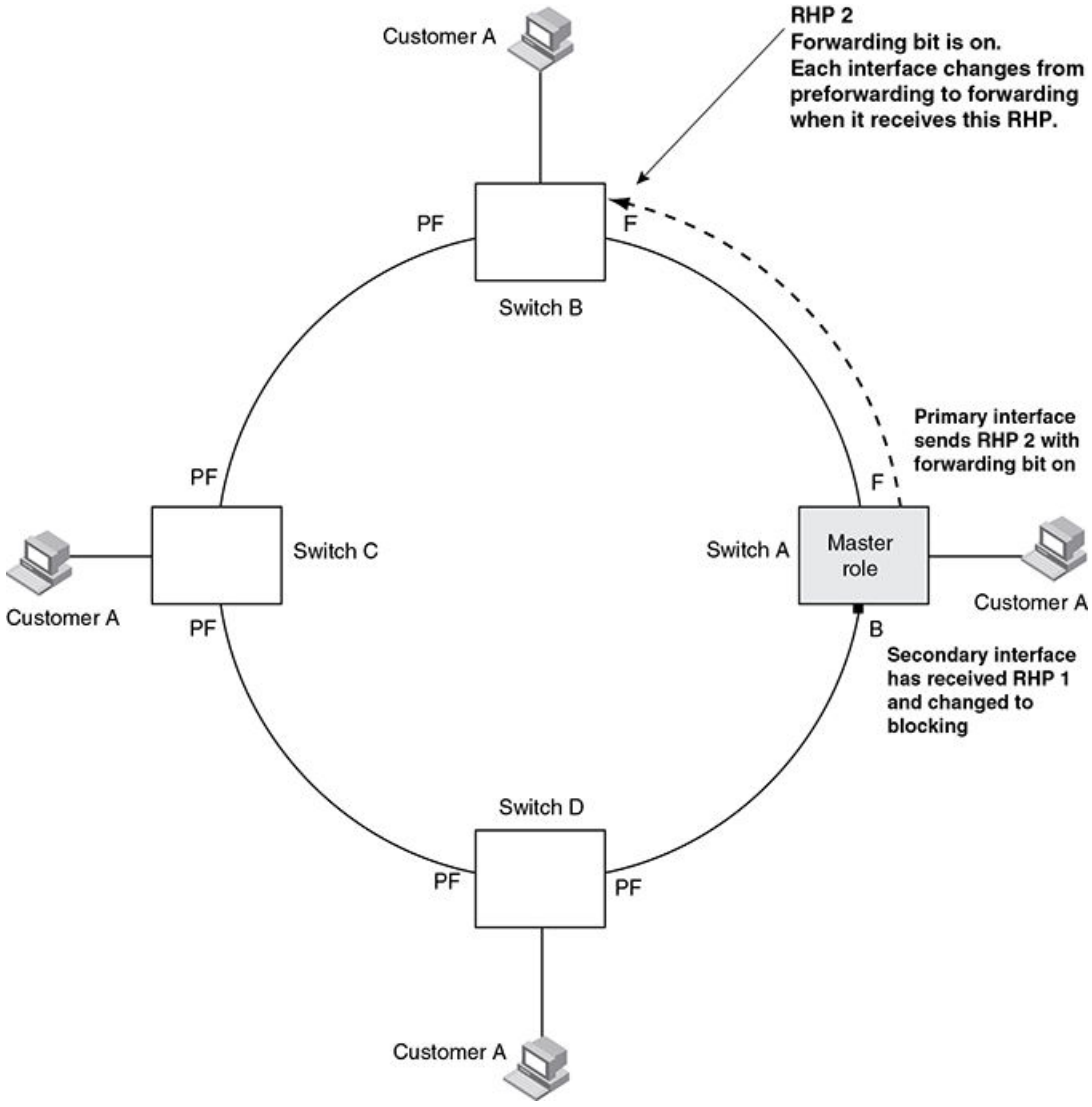
  NOTE
  The configured preforwarding time defines the number of milliseconds the interface will remain in a state before changing to the next state without receiving an RHP.

When MRP is enabled, all interfaces begin in the preforwarding state and the primary interface on the master node immediately sends an RHP (RHP 1 in Figure 101) onto the ring. The secondary interface on the master node listens for the RHP:

- If the secondary interface receives the RHP, all links in the ring are up and the interface changes its state to blocking. The primary interface then sends another RHP (RHP 2 in Figure 102) with its forwarding bit set on. As each of the member interfaces receives the RHP, the interfaces change their state to forwarding. Typically, this occurs in sub-second time. The ring very quickly enters the fully initialized state.
- If the secondary interface does not receive the RHP by the time the preforwarding time expires, a break has occurred in the ring. The secondary interface changes its state to forwarding. The ring is not intact, but data is still forwarded among the nodes using the links that are up.

Figure 102 shows an example.

FIGURE 101 MRP ring - from preforwarding to forwarding

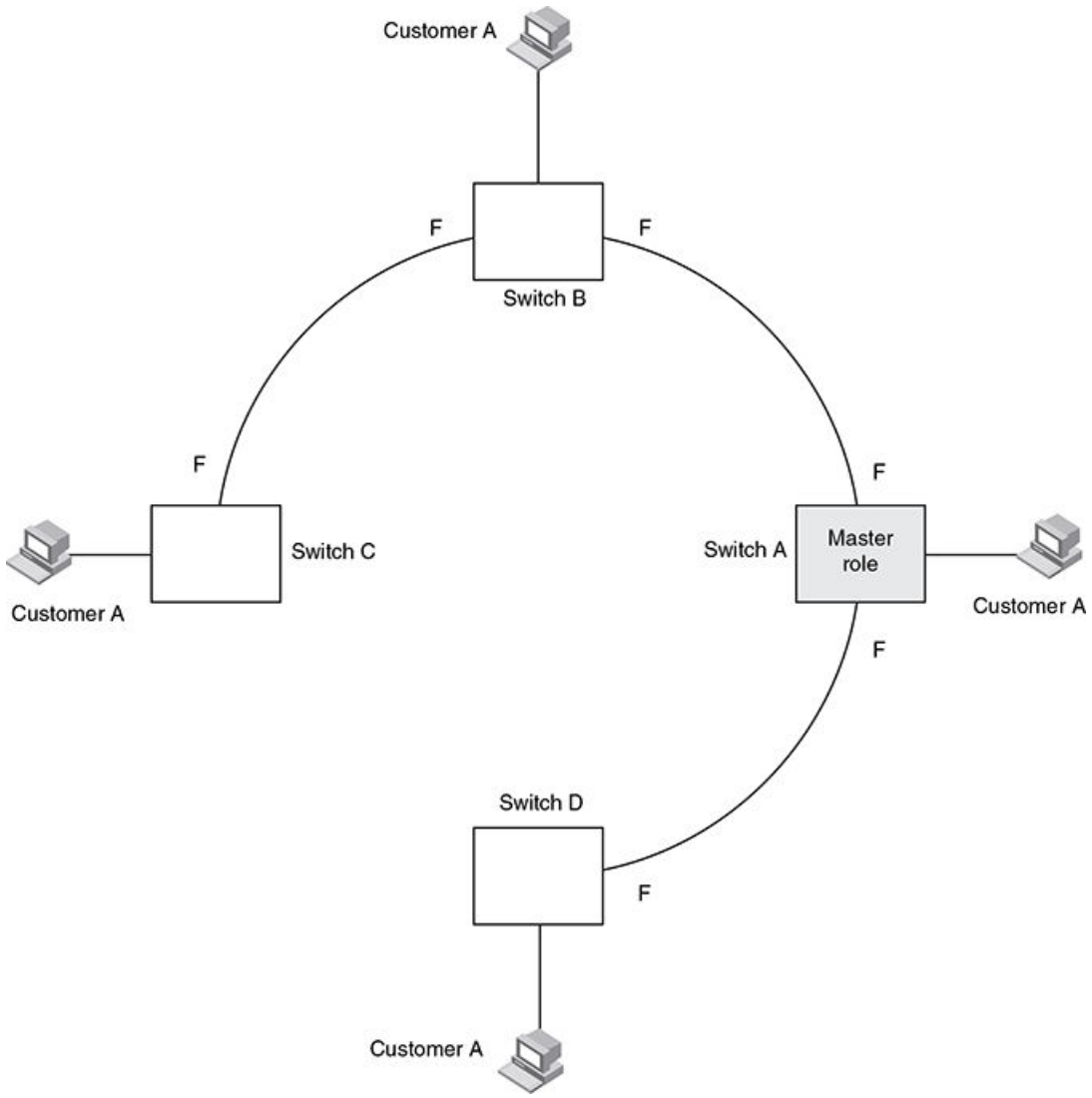


Each RHP also has a sequence number. MRP can use the sequence number to determine the round-trip time for RHPs in the ring. Refer to MRP diagnostics on page 368.

How ring breaks are detected and healed

Figure 103 shows ring interface states following a link break. MRP quickly heals the ring and preserves connectivity among the customer networks.

FIGURE 102 MRP ring - ring break



If a break in the ring occurs, MRP heals the ring by changing the states of some of the ring interfaces:

- **Blocking interface** - When the secondary interface on the master node transitions to a blocking state it sets a timer defined by the preforwarding time configured. If the timer expires before the interface receives a ring RHP, the interface changes state to preforwarding. Once the secondary interface state is preforwarding:
  - If the interface receives an RHP, the interface changes back to the blocking state and resets the timer.
  - If the interface does not receive an RHP for its ring before the preforwarding time expires, the interface changes to the forwarding state, as shown in Figure 103.
- **Forwarding interfaces** - All member interfaces remain in the forwarding state unless the physical interface is in an error condition.

When the link is repaired, the associated MRP interfaces come up in the preforwarding state allowing RHPs to be forwarded around the ring and finally reach the secondary interface on the master node:

- If an RHP reaches the master node's secondary interface, the ring is intact, the secondary interface changes to blocking. The master node sets the forwarding bit on in the next RHP. When the restored interfaces receive this RHP, they immediately change state to forwarding.
- If an RHP does not reach the master node's secondary interface, the ring is still broken. The master node does not send an RHP with the forwarding bit on. In this case, the restored interfaces remain in the preforwarding state until the preforwarding timer expires, then change to the forwarding state.

MRP alarm RHP enhancement

Prior to the enhancement detection of ring breaks were completely timer based. If the ring master fails to receive RHPs for a period of 3 "hello times" (by default the hello time is 100 ms) this indicates that the ring is broken in some manner. This initiates a topology change as described in the previous section. The convergence time associated with such an event could take several hundred milliseconds.

This enhancement enables ring nodes to rapidly notify the master of link failures. To understand the mechanism we introduce the concept of downstream switches in the ring and how member switches determine the primary and secondary ring interfaces. Remember that a primary ring interface sends RHPs and a secondary ring interface receives RHPs.

To fully understand the mechanism the reader needs to be aware of the concept of shared interfaces and interface owner ID's which are a function of MRP phase 2.

A downstream switch is defined as the next switch that will receive the ring RHP originated from the master primary interface for a particular ring. In Figure 104 Switch B is downstream from the master, Switch C is downstream from Switch B and so on and so forth. In addition it should be noted that a member switch identifies which ring interface is secondary for each discrete ring by virtue of the receipt of RHPs for that ring. In a topology with shared interfaces a single physical interface can therefore be a primary ring interface for one ring and a secondary ring interface for another ring. It should be noted that the output of the 'show metro' command as well as the configuration will change if the primary and secondary ring interfaces of the master are swapped. This keeps the identification of interface roles consistent with the flow of RHPs for discrete ring instances.

When a link is detected to be down on a member switch secondary ring interface due to a link failure an alarm RHP, which is an RHP with the alarm bit set, is sent from the primary ring interface towards the ring master, notifying the master of the failure.

The destination MAC address in the alarm is the ring MAC address. The MAC address will be in the format 0304.8000.00xx where 'xx' is the ring number in hexadecimal.

For example ring 100 = 0304.8000.0064. This ensures that the packet is hardware forwarded all the way to the master. When the master in the ring receives this alarm the secondary interface is immediately transitioned from blocking to forwarding.

NOTE

In the event of a shared interface failing the alarm RHP packet is only sent by the owner ring of the failed interface. If all rings configured on a shared interface were to generate alarms then the respective master switches for each ring would start forwarding on both interfaces creating a loop condition. By restricting alarm generation to the owner ring we ensure that only one master switch is notified to ensure that the ring heals. The owner ring ID should be the highest priority ring configured on the shared interface.

Operation of the alarm RHP enhancement is shown in Figure 104 and described below:

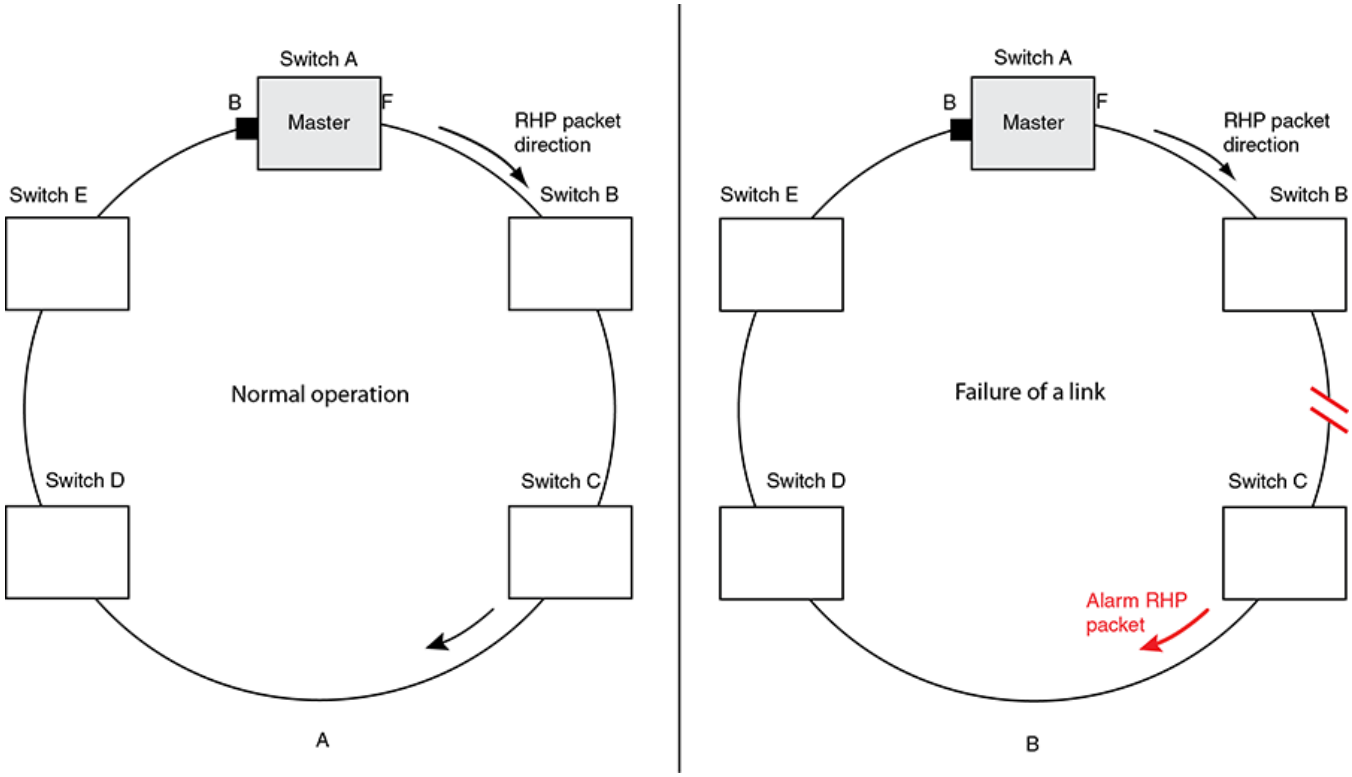
When the link between Switch B and Switch C fails, the downstream switch detects the failure of the link associated with its secondary ring interface and generates an alarm. The following is the complete sequence of events that occurs.

1. The downstream Switch C detects a link down event on the link to its upstream neighbor Switch B.

2. Switch C sends a single RHP packet with the alarm bit set. The RHP packet is sent in the same direction of flow as that of the normal RHP packets.
3. Switch A receives the alarm on the secondary ring interface that was sent by Switch C. It is now aware that the ring is broken even though the preforwarding timer for blocking to preforwarding may not have expired.
4. Switch A immediately transitions its secondary interface from blocking to forwarding to heal the ring.
5. RHP packets continue to be sent on the primary interface by Switch A to detect when the ring has been healed.

   From a user perspective there is no other difference in the behavior of the ring other than the rapid convergence due to link failures. There is no CLI command required to enable this feature.

FIGURE 103 An MRP ring under normal operation (A) and after detection of a failure in the ring (B)



Topology change notification for multicast traffic

Figure 105 shows a Layer 2 aggregation network which runs on NetIron OS MRP. In this scenario, switch A acts as the MRP master and also the Internet Group Management Protocol (IGMP) querier. When a link failure is detected on the port 1/1 of switch A, the port 1/2 of switch A will transition from blocking state to forwarding state allowing the ring to re-converge in sub-second time. With the transition of port 1/2 to forwarding state, the IGMP querier sends the IGMP reports through the port 1/2 and the IGMP receivers will send the join message to the querier causing the multicast traffic to re-converge immediately.

FIGURE 104 MRP ring with switch A as the querier

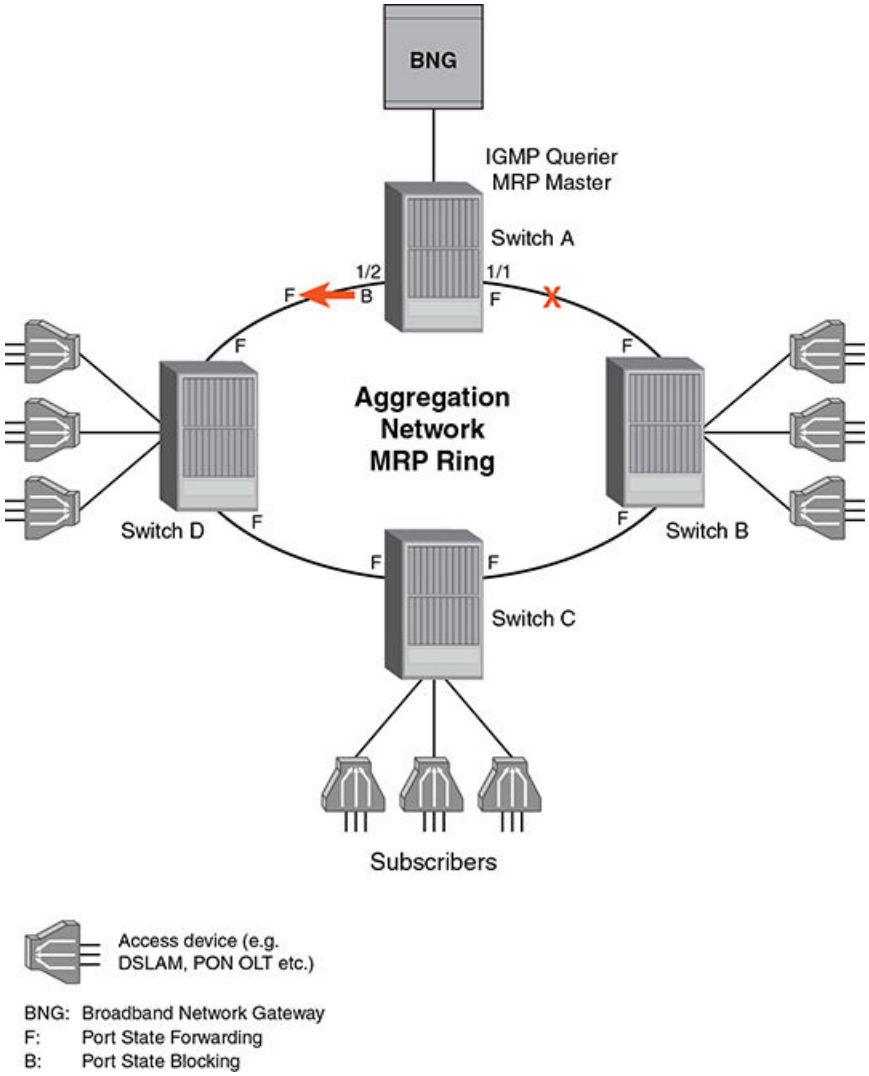
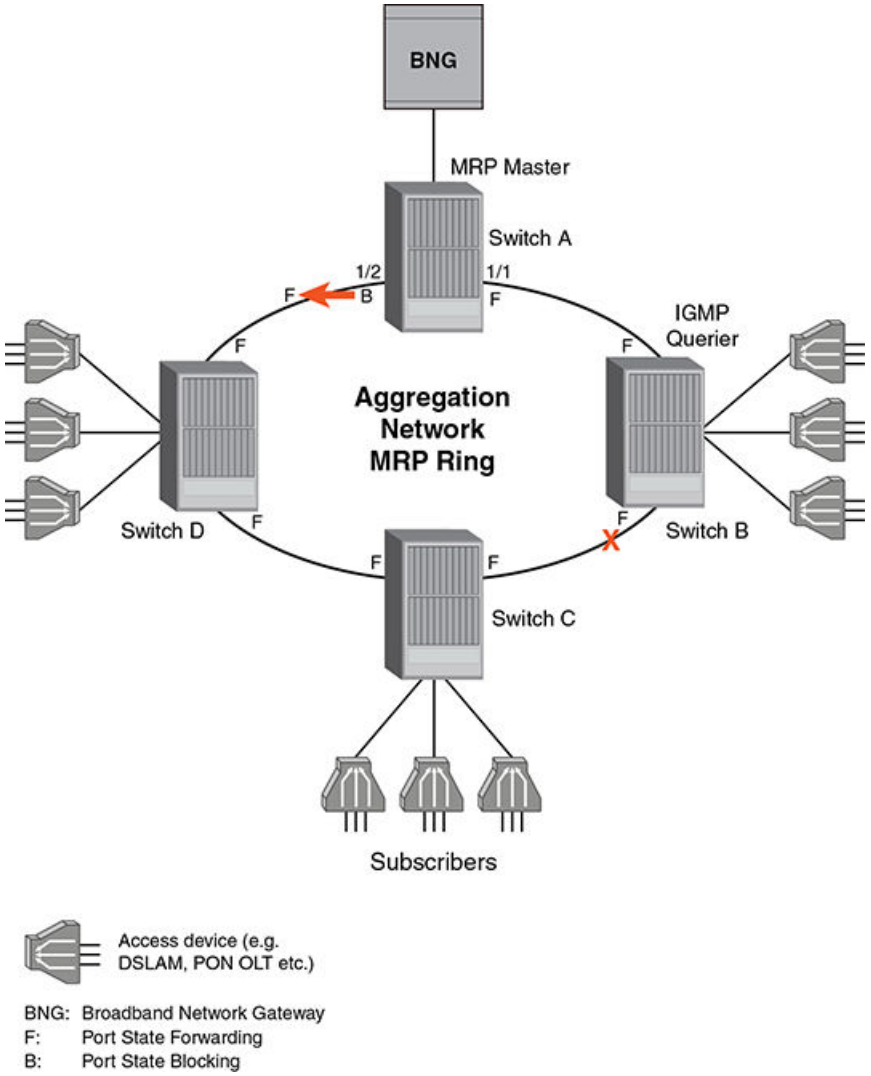


Figure 106 shows a Layer 2 aggregation network which runs on NetIron OS MRP. In this scenario, switch A acts as the MRP master and switch B acts as the IGMP querier. When a link failure is detected on the switch B, the port 1/2 of switch A will transition from blocking state to forwarding state causing the multicast traffic to re-converge. The failover time of the multicast traffic is determined by the IGMP query interval and the IGMP group membership time on the querier. These timers can be set by **ip igmp query-interval** and **ip igmp group-membership-time** commands.

FIGURE 105 MRP ring with switch B as the querier



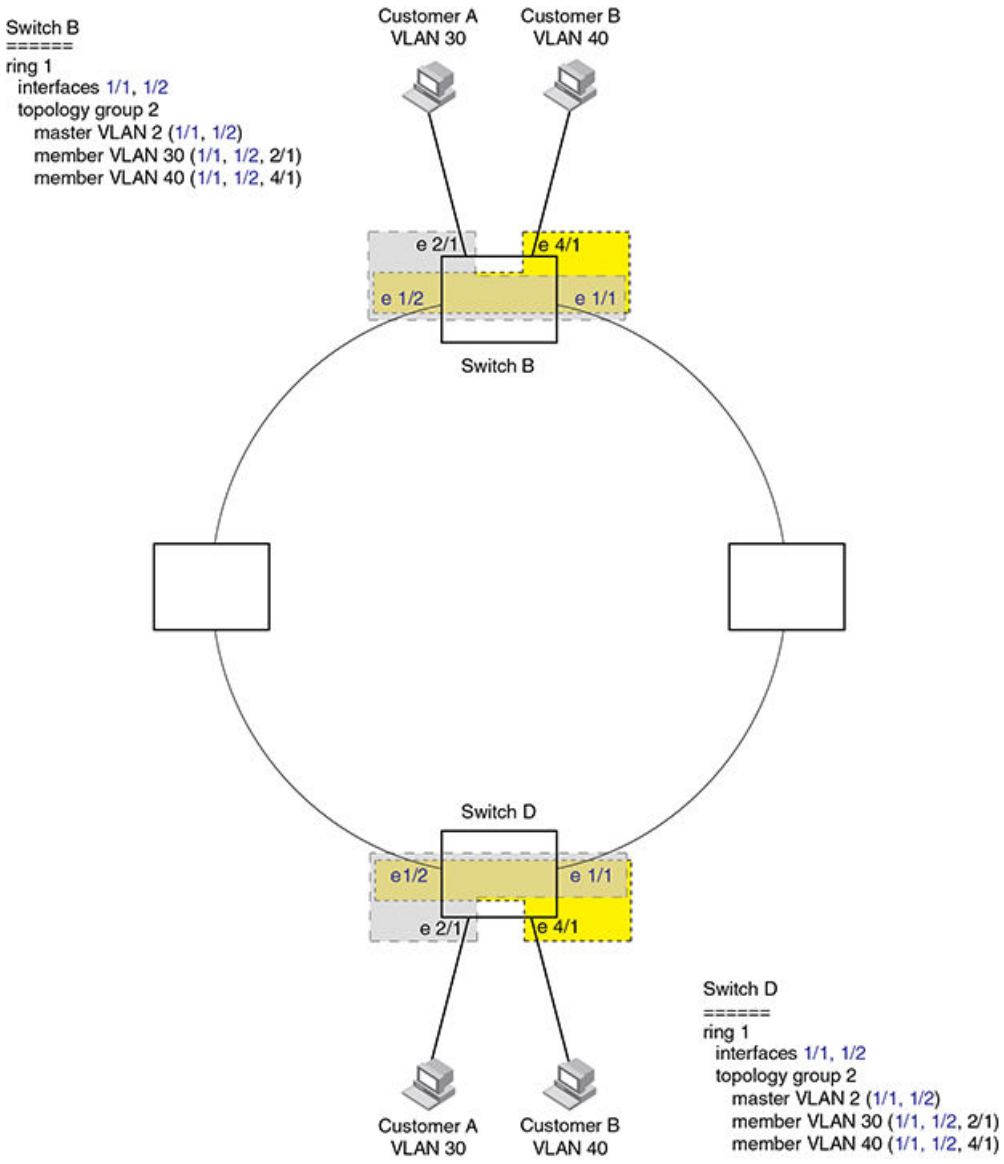
In the scenarios shown in Figure 105 and Figure 106, a topology change notification is sent to the MRP ring master, which forces an advertisement of the IGMP query to the entire network upon receiving the notification. This reduces the failover time for the multicast streams over a ring topology without the need of lowering the IGMP query interval or IGMP group membership time.

Master VLANs and customer VLANs in a topology group

The reader is referred to *Topology Groups* chapter for further information on topology group concepts and operation.

All the ring interfaces must be placed in to the master VLAN for the topology group. Customers configured with member VLANs inherit the configuration of the topology group master VLAN and have equivalent Layer 2 connectivity across the ring. Figure 107 shows an example.

FIGURE 106 MRP ring - ring VLAN and customer VLAN



In this example each customer has their own VLAN. Customer A has VLAN 30 and customer B has VLAN 40.

Customer A host attached to Switch D on an interface in VLAN 30 can reach the customer A host attached to Switch B on an interface in VLAN 30 through the ring at Layer 2. The same mechanism is used to connect customer B hosts on VLAN 40.

Customer A and customer B traffic is separated by using different VLANs.

You can configure MRP separately on each customer VLAN. However, this is impractical if you have many customers. To simplify configuration when you have many customers (and many VLANs), you can use a topology group.

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as MRP. A topology group contains of a master VLAN and member VLANs. The master VLAN contains all the configuration parameters for the Layer 2 protocol (STP, MRP, or VSRP). The member VLANs use the Layer 2 configuration of the master VLAN.

In Figure 107, VLAN 2 is the master VLAN and contains the MRP configuration parameters for ring 1. VLAN 30 and VLAN 40 are member VLANs in the topology group. Since a topology group is used, a single instance of MRP provides redundancy and loop prevention for both the customer VLANs.

If you use a topology group:

- The master VLAN must contain the ring interfaces.
- The ring interfaces must be tagged as they will be used for multiple VLANs.
- The member VLAN for a customer must contain the two ring interfaces and the interfaces for the customer.

Refer to MRP CLI example on page 369 for the configuration commands required to implement the MRP configuration shown in Figure 107.

Configuring MRP

To configure MRP, perform the following tasks for each discrete ring:

- On the switch identified as the ring master disable the secondary ring interface. This manually prevents a Layer 2 loop from occurring during configuration.
- Configure each switch for MRP one at a time following the planned flow of RHP's
- On each switch in the path add an MRP ring to a port-based vlan. When you add a ring, the command changes to the configuration level for the ring, where you can do the following:
  - On the master node configure the master ring role.
  - Specify the two MRP interfaces for the ring.
  - Option: Specify a name for the ring. Extreme recommends that you have a naming convention for your MRP rings and consistently apply names for all the rings in the topology.
  - Option: Change the hello time and the preforwarding time. These parameters control how quickly failover occurs if the master fails to receive RHPs for the ring.
  - Enable the ring.
- Re-enable the interface you disabled in step one. MRP will prevent loops when enabled on all devices in the ring.

When using topology groups the ring configuration must be added to the master-vlan for the group. For further information refer to *Topology Groups* chapter.

Configuration considerations

- When you configure MRP, Extreme recommends that you disable one of the ring interfaces before beginning the ring configuration. Disabling an interface prevents a Layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once MRP is configured and enabled on all the nodes, you can re-enable the interface.
- The above configurations can be configured as MRP masters or MRP members (for different rings).
- If you configure MRP on a device running Layer 3 software, then restart the device running Layer 2 software, the MRP configuration gets deleted.

Adding an MRP ring to a VLAN

NOTE
If you plan to use a topology group make sure you configure MRP on the topology group's master VLAN.

To add a MRP ring to a VLAN, enter commands such as the following.

```
device(config)# vlan 2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name CustomerA
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
```

These commands configure an MRP ring in VLAN 2 with a ring ID of 1, a ring name of Customer A. If the node is the master then the following command is used to specify the node as the master for the ring.

device(config-vlan-2-mrp-1)# master

The ring interfaces are 1/1 and 1/2. The first interface listed will be allocated as the primary interface and the second will be allocated as the secondary interface. The primary interface initiates RHPs. The ring takes effect in VLAN 2.

Syntax: [no] metro-ring ring-id

The *ring-id* parameter specifies the ring ID 1 - 255. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] name string

The *string* parameter specifies a name for the ring. The name is optional, but it can be up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] master

Configures this node as the master node for the ring. Enter this command only on one node in the ring. The node is a member (non-master) node by default.

Syntax: [no] ring-interface ethernet primary-if ethernet secondary-if

The **ethernet** *primary-if* parameter specifies the primary interface. On the master node, the primary interface originates RHPs. Ring control traffic will flow out of this interface by default. On member nodes the order in which you enter the interfaces does not matter as the secondary interface is determined by the receipt of RHPs from the master meaning the other interface defined in config becomes the primary. Once the ring is enabled the configuration entries on a member switch will reflect the ring direction no matter what order they are originally entered.

The **ethernet** *secondary-if* parameter specifies the secondary interface.

> NOTE
> The CES 2000 Series and CER 2000 Series devices do not support selection of a secondary interface based on reception of RHPs. As a result, the primary and secondary interfaces must be configured correctly.

Syntax: [no] enable

The **enable** command enables the ring.

Changing the hello and preforwarding times

You can also change the RHP hello time and preforwarding time. To do so, enter commands such as the following.

```
device(config-vlan-2-mrp-1)# hello-time 200
device(config-vlan-2-mrp-1)# preforwarding-time 400
```

These commands change the hello time to 200 ms and change the preforwarding time to 400 ms.

Syntax: [no] hello-time ms

Syntax: [no] preforwarding-time ms

The *ms* specifies the number of milliseconds.

The hello time can be from 100 - 1000 (one second). The default hello time is 100 ms.

The preforwarding time can be from 200 - 5000 ms, and must be at least twice the value of the hello time and must be a multiple of the hello time. The default preforwarding time is 300 ms.

A change to the hello time or preforwarding time takes effect as soon as you enter the command.

> **NOTE**
> You can use MRP ring diagnostics to determine whether you need to change the hello time and preforwarding time. Refer to MRP diagnostics on page 368.

Changing the scale timer

You are able to decrease MRP convergence time by changing the MRP scale timer tick from 100 ms to 50 ms. To do so, enter the following command:

```
device(config)# scale-timer mrp
```

Syntax: [no] scale-timer mrp

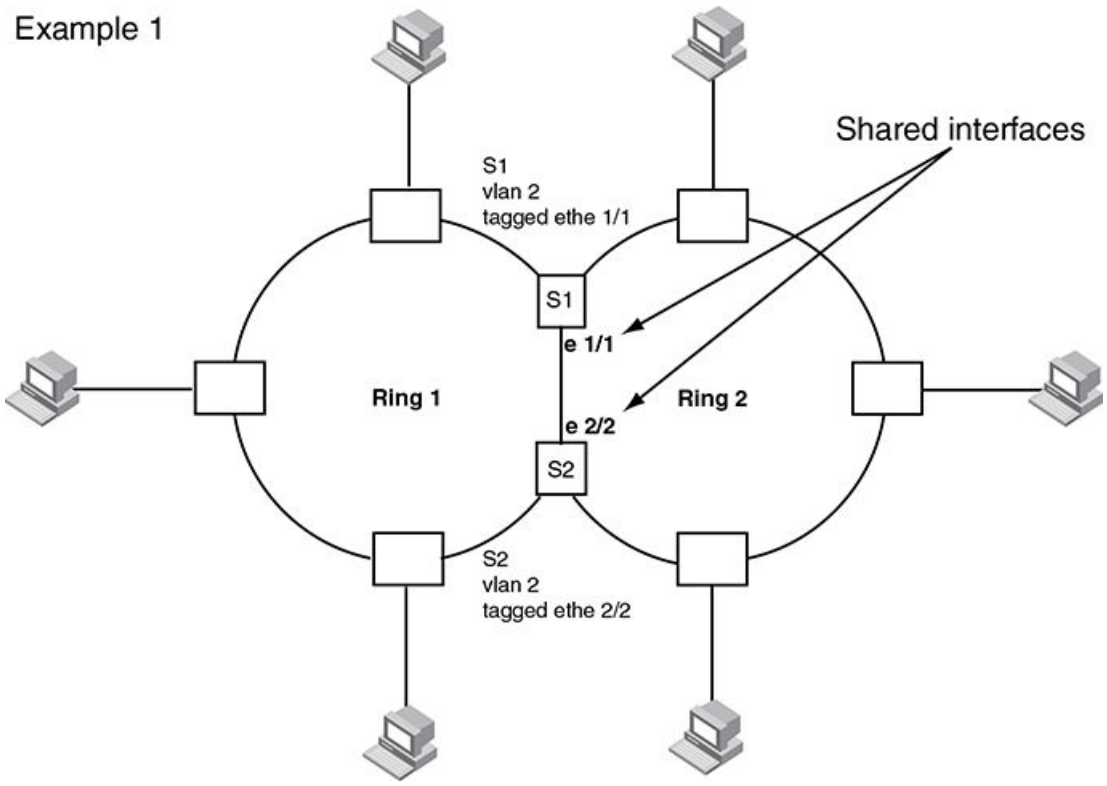
Note: This command accepts no values and is put in place as it is shown above.

Note: Changing the scale timer affects the operation of MRP. Refer to MRP timers on page 366 for further information.

MRP rings with shared interfaces

With MRP phase 2, multiple rings can be configured to share the same interface as long as the interfaces belong to the same VLAN. Figure 108 shows an example of two rings that share the same interfaces on S1 and S2.

FIGURE 107 Example 1 multiple rings sharing interfaces - phase 2



On each node that will participate in ring 1, you configure the ring ID and the ring interfaces that will be used. You repeat the configuration steps for all nodes in ring 2. In a multiple ring configuration, a ring's ID determines its priority. The lower the ring ID, the higher the priority of a ring with ring ID 1 being the highest possible priority.

A key concept with MRP phase 2 is the ability to extend a single VLAN across the whole topology even when multiple rings are required. Consider the example in Figure 109 where we have three MRP rings and a customer who needs to create neighbor relationships between all three routers depicted. The routers all have interfaces configured in a single subnet and need IP connectivity between each other.

If each ring had an independent VLAN then we would have to have a mechanism to move IP packets from a single IP subnet between different Layer 2 topologies. By using MRP phase 2 we have multiple rings all associated with a single Layer 2 topology allowing a common subnet to be distributed in the manner shown in the example. Whilst this looks nothing like a standard spanning tree network it should be treated in the same way from the perspective of a Layer 2 topology, multiple paths where certain paths must be blocked to prevent loops at Layer 2.

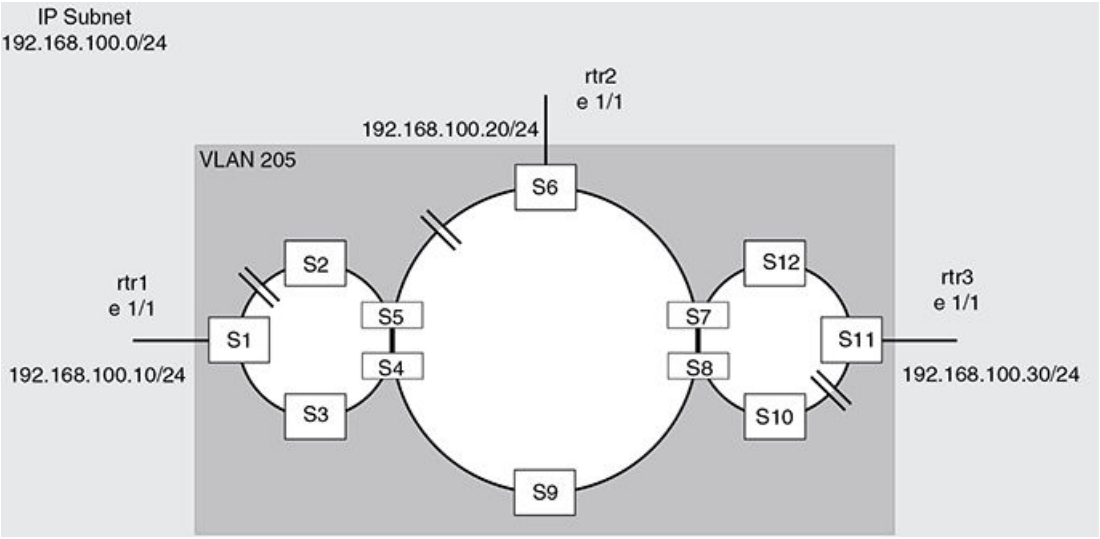
In addition it should be noted that the concept of multiple rings being associated with a single VLAN describes the extent to which broadcasts will be propagated at Layer 2 for that VLAN. In other words a broadcast will be propagated to all ring interfaces in the Layer 2 topology. The use of topology groups allows multiple VLANs to effectively reuse a single Layer 2 topology while maintaining a level of separation.

The obvious issue with this approach is that there must be a mechanism to prevent loops on the rings and this is the job of MRP.

It is very easy to focus on the ring topology rather than the underlying Layer 2 topology described by multiple rings. Design decisions are driven by the same factors as a standard spanning tree network replacing root bridges with ring masters. Traffic patterns at Layer 2 are determined by which ring interfaces are forwarding and which are blocking and this in turn should drive design decisions for ring master

placement as well as the direction of RHP flow from the ring masters. Traffic patterns in standard operation as well as failure mode can be determined prior to implementation allowing for appropriate capacity management on all links.

FIGURE 108 Multiple rings with one VLAN spanning them



Ring interface ownership

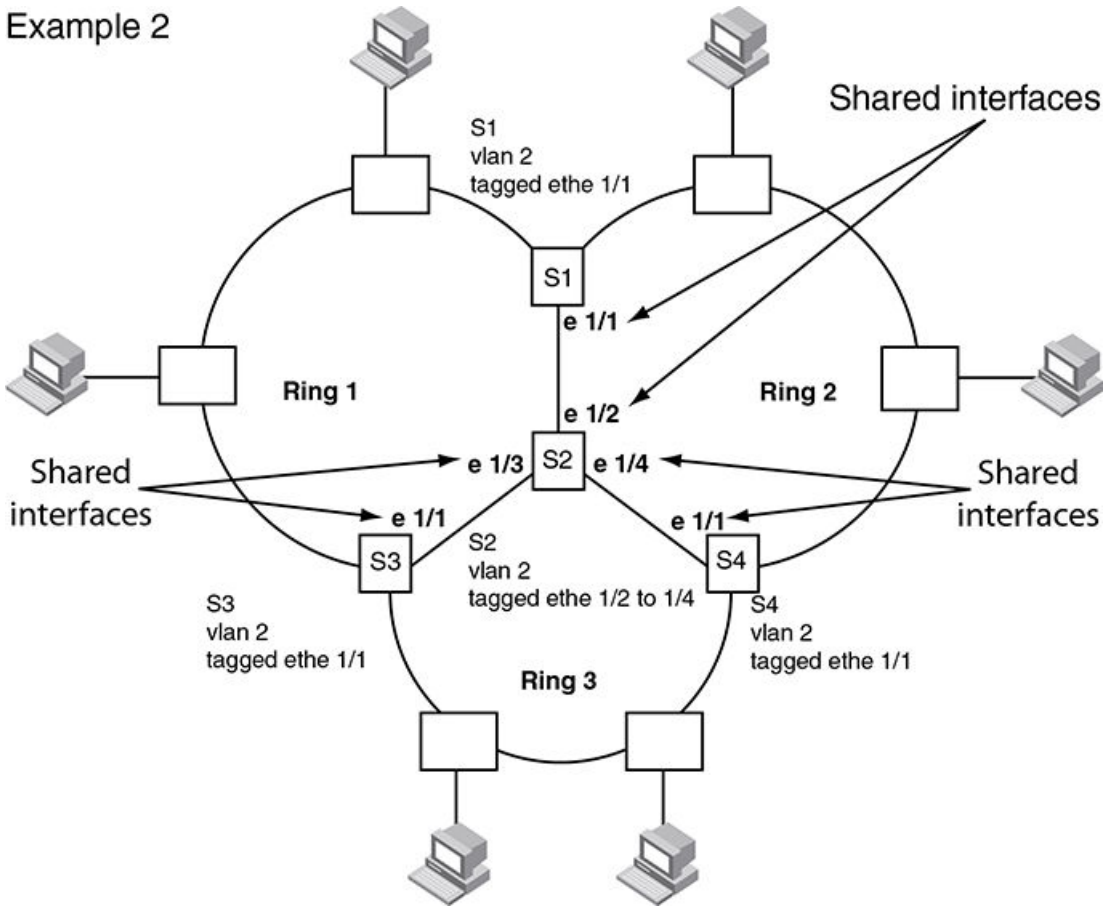
On a shared interface the highest priority ring will be the owner of the interface. In Figure 110 interface ethernet 1/1 on S1 will be owned by ring 1 and marked as a regular interface while in ring 2, the same interface is marked as a tunnel interface in the output of the **show metro** command.

On S2 interface ethernet 1/2 is again owned by ring 1 and marked as a regular interface.

In Figure 110 the same principles of interface ownership apply. All shared interfaces on ring 1 nodes are shown as owned by ring 1 and marked as regular interfaces. Ring 2 will show shared interfaces as tunnel interfaces.

On S2 ethernet 1/4 and on S4, ethernet 1/1 interfaces will be owned by ring 2, as the highest priority ring on the interface, and ring 3 will show these interfaces as tunnel interfaces.

FIGURE 109 Example 2 multiple rings sharing interfaces - phase 2



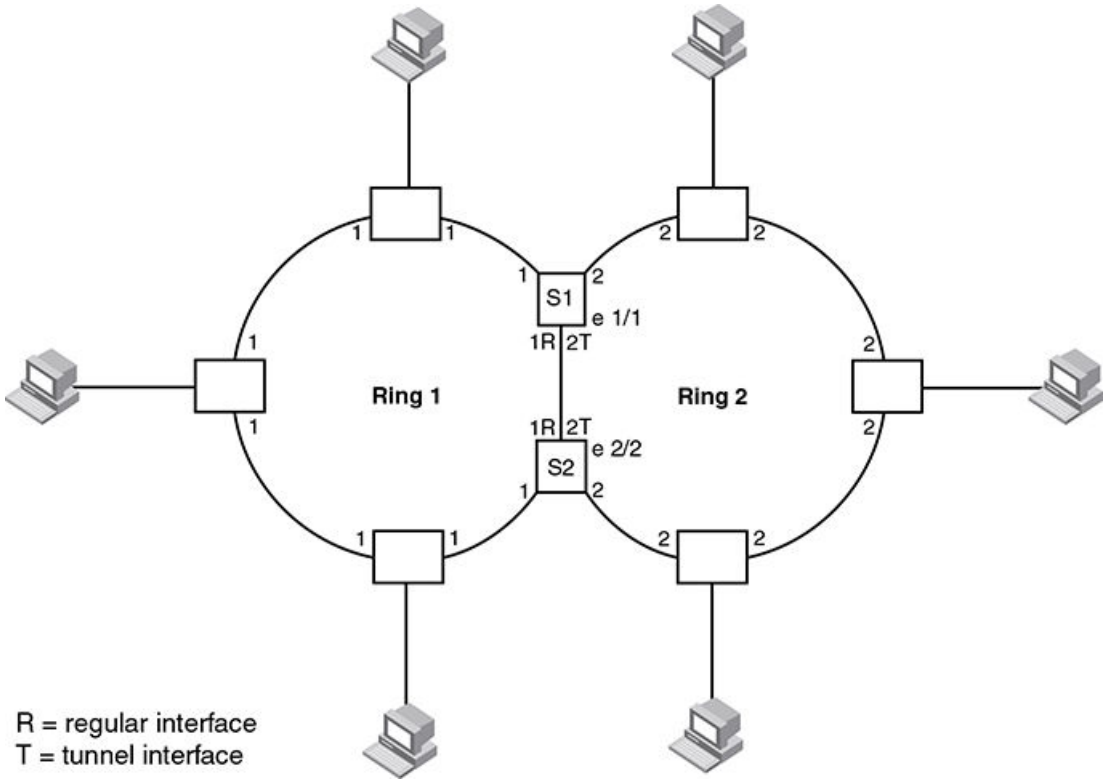
Ring interface IDs and types

For example, in Figure 111, all interfaces configured for ring 1 have a priority of 1. Interface e 1/1 on S1 and e 2/2 on S2 have a priority of 1 since 1 is the highest priority ring that shares the interface.

All interfaces on ring 2, except for e 1/1 on S1 and e 2/2 on S2 have a priority of 2.

If a node has shared interfaces then the ring interfaces that belong to the ring with the highest priority are regular interfaces for that ring and all lower priority ring interfaces are marked as tunnel interfaces. The highest priority ring configured becomes the priority for the interface.

FIGURE 110 Interface IDs and types



In Figure 111, nodes S1 and S2 have interfaces that belong to rings 1 and 2. Interface e 1/1 on S1 and e 2/2 on S2 are regular interfaces for ring 1, but they are tunnel interfaces for ring 2.

Selection of the master node for a ring

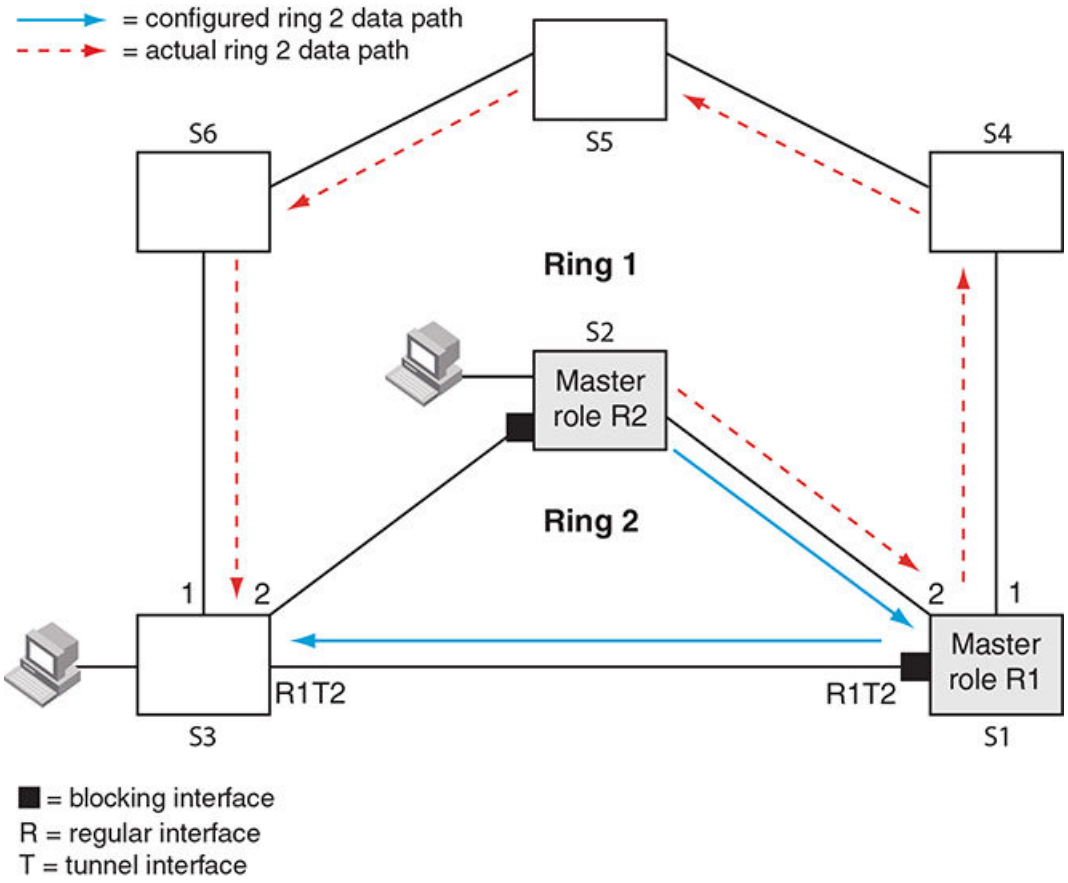
Configuring MRP rings with shared interfaces limits the nodes that can be designated as the master node for any particular ring.

- Any node on the ring that does not have any shared interfaces can be designated as the ring's master.
- You can only designate a node that has shared interfaces as master for a ring where all interfaces for the ring are marked as regular interfaces.
- On a node with shared interfaces, where you configure the role as master, the secondary ring interface should not be a shared interface. If you designate a shared interface as secondary it will be blocking under normal operation and allow RHP's but no data for lower priority rings. This can create unexpected traffic flows on the rings.

In Ring interface IDs and types on page 358 any of the nodes on ring 1, even S1 or S2, can be a master node as all of the ring interfaces, even the shared interfaces between S1 and S2, are marked as regular interfaces for ring 1.

However for ring 2, neither S1 nor S2 can be a master node since the shared interfaces between S1 and S2 are marked as tunnel interfaces for ring 2.

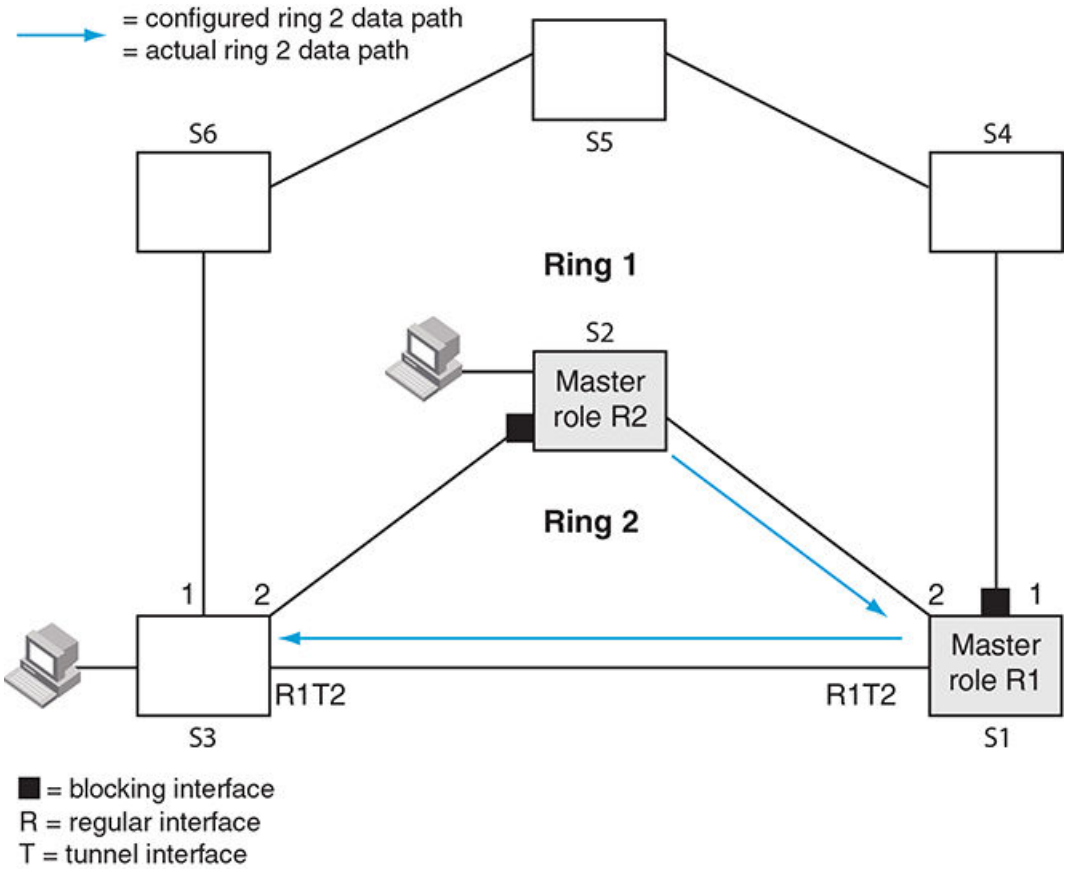
FIGURE 111 Unexpected switching path with shared interface



In Figure 112 ring 2 was configured with shared ring interfaces on S1 and S3 as depicted. S1 was configured as the master for ring 1 and the shared interface was defined as the secondary interface and subsequently blocks data. The designer intended the switching path between a host on S2 and another host on S3 to be via S1 shared interface, however due to the shared interface being blocked the actual switching path becomes S1 to S4,S5,S6 and finally S3.

Ring 2 is still operational but is not behaving in the manner which the design called for. By configuring the secondary interface on the regular port for ring 1 we obtain the expected result as shown in Figure 113.

FIGURE 112 Expected switching path with shared interface



RHP processing in rings with shared interfaces

Interfaces on an MRP ring have one of the following states:

- **Blocking (B)** - The interface can process RHPs, but cannot forward data for the ring. Only the secondary interface on the Master node can be blocking. If the interface receives RHP's for lower priority rings these RHPs will be discarded by this interface. This prevents RHP's from lower priority rings from looping in the topology.
- **Preforwarding (PF)** - The interface will forward RHPs but won't forward data for the ring. All ring interfaces start in this state when you enable MRP. A blocking interface transitions to preforwarding when the preforwarding timer expires.
- **Forwarding (F)** - The interface will forward RHP's and data for the ring. On member switches an interface transitions from preforwarding to forwarding when the preforwarding time expires or the interface receives an RHP with the forwarding bit set. A break in the ring is indicated if the secondary interface on the master fails to receive an RHP within the preforwarding timer and the interface transitions from blocking to forwarding to heal the ring. The preforwarding time is the number of milliseconds the interface will remain in the preforwarding state before changing to the Forwarding state, even without receiving an RHP.

The primary interface of the master node initiates RHP packets and sends them onto the ring. When the packet reaches a forwarding interface, MRP checks to see if the receiving interface is a regular interface or a tunnel interface:

- If the interface is a regular interface, the RHP packet is forwarded to the next interface. Forwarding of the packet continues on the ring until the secondary interface of the master node receives the packet and processes it. For the configured ring the receipt of an RHP with the same ring ID indicates the ring is healthy. RHPs for lower priority rings will be discarded without further processing at this point.

- If the interface is a tunnel interface, MRP checks the priority of the RHP packet and compares it to the priority of the tunnel interface:
  - If the RHP packet's priority is less than or equal to the interface's priority, the packet is forwarded through ring interfaces with higher priority which are in the forwarding state.
  - If the priority of the RHP packet is greater than the priority of the interface, the RHP packet is dropped. For example, if an RHP with a ring ID of 1 arrives at a tunnel interface owned by ring 2 the RHP will be dropped. If an RHP with a ring ID of 2 or 3 arrives at a tunnel interface owned by ring 2 the RHP will be forwarded.

  NOTE
  It is important to understand the key concept of RHPs leaking from lower priority rings to higher priority rings. Always remember that tunnel interfaces check the ring ID of an RHP before forwarding. Higher priority ring ID RHPs will be dropped.

How ring breaks are detected and healed between shared interfaces

If the link between shared interfaces breaks, the secondary interface on the highest priority ring master node changes to a preforwarding state, refer to Flow when a link breaks on page 364. Any RHP from lower priority rings can traverse this interface and thus maintain the integrity of the lower priority rings. When the secondary interface changes state to forwarding the lower priority ring RHP's continue to traverse the interface.

This behavior allows the ring 2 RHP's to continue around ring 1 and back to ring 2 until it reaches the secondary interface on ring 2's master node which changes to blocking mode since it receives its own RHP.

NOTE
On the ring member node, the primary and secondary interface is decided by the RHP flow from the ring master. The secondary interface is always the RHP receiver for its ring RHP's, the primary interface is always the sender of its rings RHP's. If there is no active ring master in the topology, then the running configuration on the member node will show exactly what was configured. This may change on introduction of an active ring master.

Normal flow

Figure 114 and Figure 115 show how RHP packets are forwarded in rings with shared interfaces. Figure 114 shows the flow of ring 1 RHPs while Figure 115 shows how ring 2 RHPs flow.

Interface e 2/1 is the primary interface of the ring 1 master node. The primary interface forwards an RHP packet on the ring. Since all the interfaces on ring 1 are regular interfaces, the RHP packet is forwarded until it reaches interface e 2/2, the secondary interface of the ring 1 master. Receipt of this RHP indicates a healthy ring 1 and interface e2/2 then changes to or maintains its state of blocking.

No copies of the ring 1 RHPs are forwarded on ring 2 tunnel interfaces or ring 2 regular interfaces in accordance with the rule that a higher priority RHP is not permitted to traverse a lower priority ring interface.

FIGURE 113 RHP flow on rings with shared interfaces showing ring 1 RHP flow

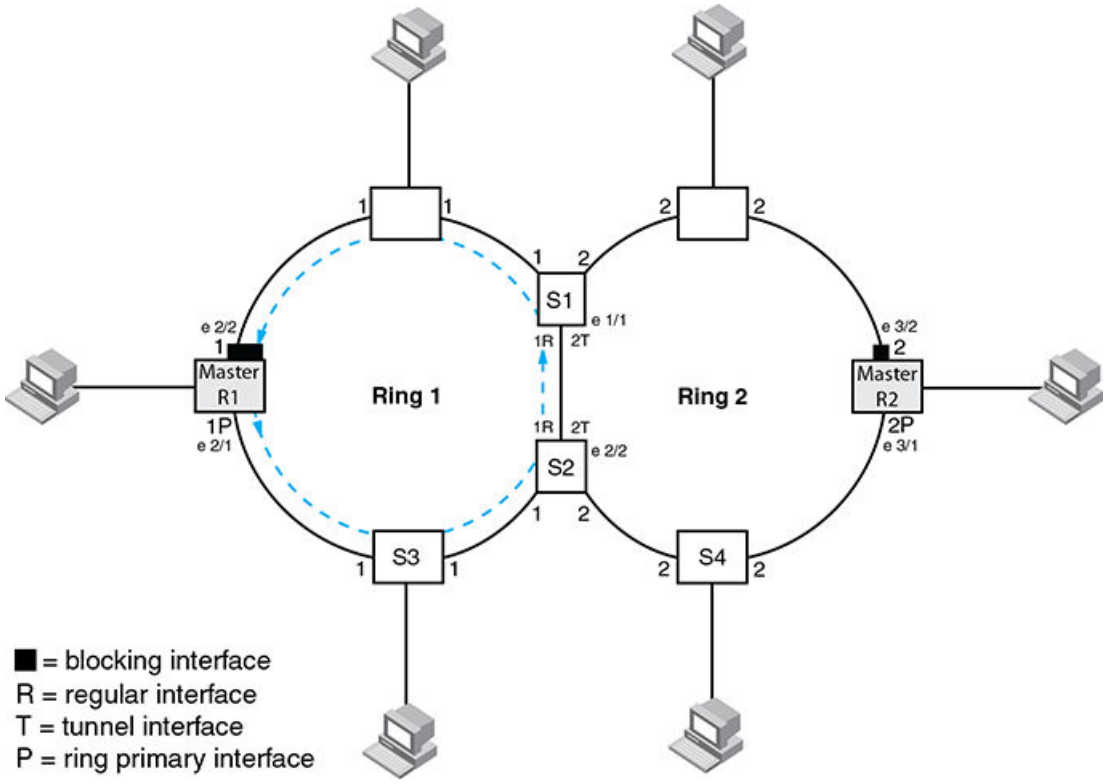
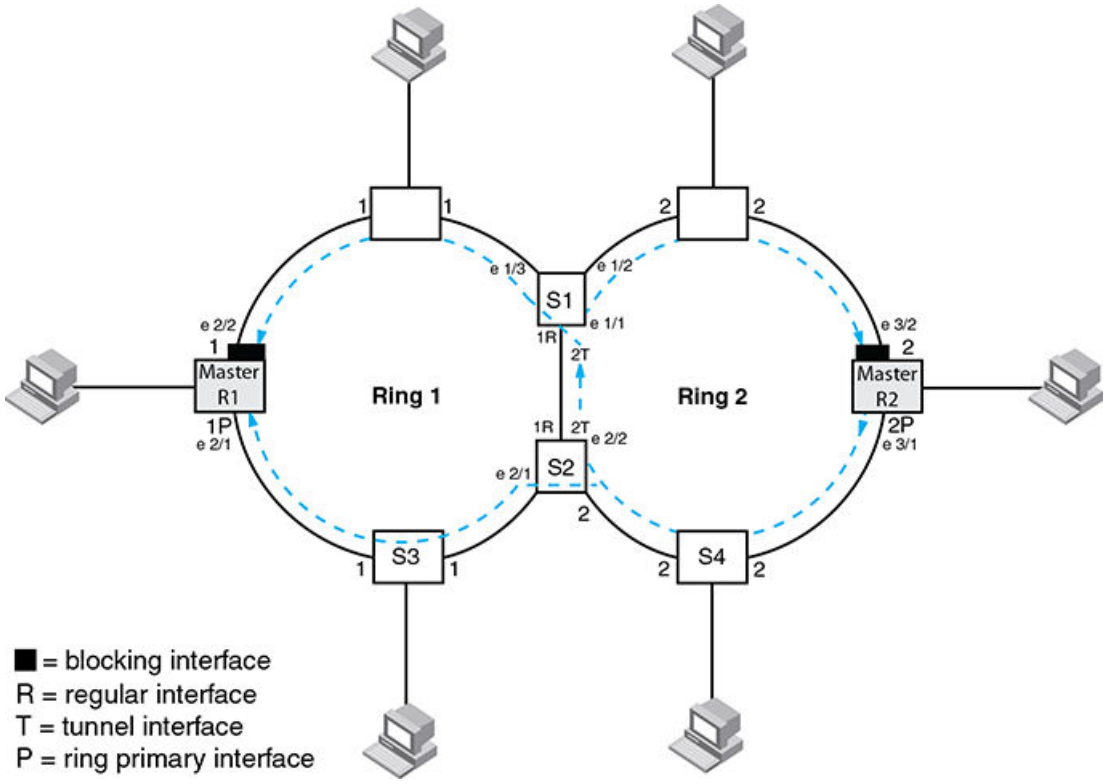


FIGURE 114 RHP flow on rings with shared interfaces showing ring 2 RHP flow



Referring to Figure 115 interface 3/1, is the primary interface of the ring 2 master node. It sends an RHP packet on the ring. Since all interfaces on S4 are regular interfaces, the RHP packet is forwarded on those interfaces.

When the RHP reaches S2:

- A copy of the RHP is sent out of regular interface e 2/1 onto ring 1. This is in accordance with the rule that a lower priority RHP can traverse a higher priority ring interface. This RHP is forwarded until it reaches the ring 1 master where it is discarded.
- A copy of the RHP is forwarded out of the ring 2 tunnel interface on e 2/2

The RHP is received by S1 on e 1/1 and then:

- A copy of the RHP is sent out of regular interface e 1/2 on ring 2
- A copy of the RHP is sent out of regulate interface e 1/3 on ring 1. This is in accordance with the rule that a lower priority RHP can traverse a higher priority ring interface. This RHP is forwarded until it reaches the ring 1 master where it is discarded.

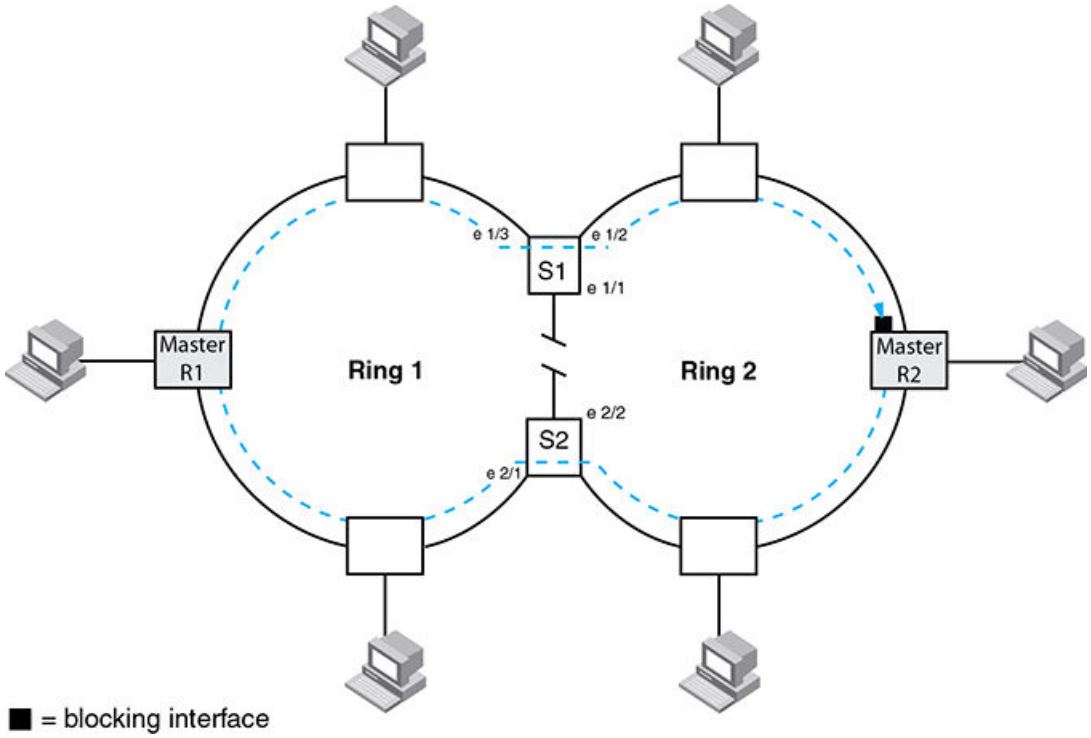
Flow when a link breaks

Referring to Figure 116 if the link between S1 and S2 fails, the secondary interface on the ring 1 master node changes to a forwarding state.

The RHPs from the master for ring 2 reach S2 and a copy of the RHP is forwarded out of e 2/1. This RHP traverses the ring 1 master and continues around ring 1 until it reaches S1. After S1 the RHP is back on ring 2 and is finally received by the master for ring 2 which keeps its secondary interface in blocking mode.

It should now be clear how the flow of lower priority RHPs over the higher priority ring ensure that both ring masters do not transition to forwarding and create a loop condition.

FIGURE 115 Flow of RHP packets when a link for shared interfaces breaks



Ring 2 RHPs follow this path until the link is restored. Once the link is restored the ring 1 master will transition its secondary ring interface to blocking and the ring 2 RHP flow is as shown in Normal flow on page 362.

NOTE

There should always be a layer 2 protocol configured in the default vlan when MRP is configured with all dual mode ports.

Configuring MRP with shared interfaces

MRP Phase 2 allows you to enter commands such as the following when configuring MRP.

```
device(config)# vlan 2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name CustomerA
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2-mrp-1)# metro-ring 2
device(config-vlan-2-mrp-2)# name CustomerB
device(config-vlan-2-mrp-2)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
```

Syntax: [no] metro-ring ring-id

The *ring-id* parameter specifies the ring ID, which can be from 1 - 255. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] name string

The *string* parameter specifies a name for the ring. The name is optional, but it can have up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] ring-interface ethernet

The **ethernet** *primary-if* parameter specifies the primary interface. On the master node, the primary interface is the one that originates RHPs. Ring control traffic and layer 2 data traffic will flow in the outward direction from this interface by default. On member nodes, the direction of traffic flow depends on the traffic direction selected by the master node. Therefore, on a member node, the order in which you enter the interfaces does not matter.

The **ethernet***secondary-if* parameter specifies the secondary interface.

Syntax: [no] enable

The **enable** command enables the ring.

MRP timers

To effectively tune MRP timers it is crucial to understand the association between the hello time and the preforwarding time.

Flushing the MAC table following an MRP event

After an MRP event switches in the ring, flush the MAC tables and relearn to ensure correct forwarding paths. Notification to flush is carried out by sending topology change Ring Health Packets (RHPs).

Hello time

This timer specifies the interval at which RHP's are generated by the ring master. It should be noted that this interval is applied not only to standard RHP's but also to topology change notification RHP's. For example: Setting the hello time to its maximum value of 15,000 ms would mean that the three topology change notification RHP's that are sent following a ring break being detected or a ring heal event would result in MAC table flushes three times at 15 second intervals. On a busy network this would cause unnecessary impact.

Preforwarding time

The preforwarding time defines the amount of time an interface will take to move from blocking to preforwarding without RHP's being received. It also defines the amount of time an interface will take to move from preforwarding to forwarding without RHP's being received.

The preforwarding time must be at least 2 x hello time and must be a multiple of the hello time.

The preforwarding time for a lower priority ring must be greater than or equal to the highest higher priority ring.

For example: Setting the preforwarding time to its maximum value of 30,000 ms will mean that a break in the ring (assuming no alarm RHP's are generated) will take one minute to heal.

Setting hello and preforwarding timers appropriately

When setting timers both the hello time and the preforwarding time should be considered to ensure that the appropriate recovery time is applied on the network.

Consider a break in the network that does not generate alarm RHP's

Example 1(default values):

Preforwarding time = 300ms

Hello time = 100ms

Time to forwarding = 2 x preforwarding time = 600ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 900ms = 0.9secs

Example 2:

Preforwarding time = 10000ms

Hello time = 100ms

Time to forwarding = 2 x preforwarding time = 20000ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 20300ms = 20.3secs

Example 3:

Preforwarding time = 10000ms

Hello time = 5000ms

Time to forwarding = 2 x preforwarding time = 20000ms

Post recovery mac table flush time = 3 x hello time = 15000ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 35000ms = 35secs

It can therefore be seen that the hello time should not be changed on the network unless there is evidence of regular misses on the ring. Time to traverse the ring can be determined by running MRP diagnostics.

Effect of the scale timer

Changing the scale timer has a significant effect on the operation of MRP and should be considered for very high performance low latency networks where a very rapid failure detection and recovery mode is required. Achieving this rapid detection and recovery requires very stable high speed environments to prevent a high level of unnecessary topology changes in the environment.

The effect of setting the scale timer is that the time taken to move from blocking to preforwarding and preforwarding to forwarding is (preforwarding value - the hello time). This is a significant change to the operation of MRP in the default state which has been described in the previous section.

Note: When setting the timer at the CLI the actual value used will be exactly half of the input value. The examples that follow assume the corrected value.

Consider a break in the network that does not generate alarm RHP's

Example 1(default values):

Preforwarding time = 300ms

Hello time = 100ms

Time to forwarding = preforwarding time - hello time = 200ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 500ms = 0.5secs

Example 2:

Preforwarding time = 100ms

Hello time = 50ms

Time to forwarding = preforwarding time - hello time = 50ms

Post recovery mac table flush time = 3 x hello time = 150ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 200ms = 0.2secs

Example 3:

Preforwarding time = 10000ms

Hello time = 5000ms

Time to forwarding = preforwarding time - hello time = 5000ms

Post recovery mac table flush time = 3 x hello time = 15000ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 20000ms = 20sec

MRP diagnostics

The MRP diagnostics feature calculates how long it takes for RHP packets to travel through the ring. When you enable MRP diagnostics, the software tracks RHP packets according to their sequence numbers and calculates how long it takes an RHP packet to travel one time through the entire ring. When you display the diagnostics, the CLI shows the average round-trip time for the RHP packets sent since you enabled diagnostics. The calculated results have a granularity of 1 microsecond.

Enabling MRP diagnostics

To enable MRP diagnostics for a ring, enter the following command on the Master node, at the configuration level for the ring.

```
device(config-vlan-2-mrp-1)#diagnostics
```

Syntax: [no] diagnostics

NOTE
When using the 'show metro' command, the member node of a ring does not display correctly since the MRP RHPs are hardware forwarded (or software forwarded on the linecard), these statistics are only reflective of the MRP RHPs that made it to the management processor. In most cases, these would be TC RHPs since the MP needs to flush MACs in that case.

NOTE
This command is valid only on the master node.

Displaying MRP diagnostics

To display MRP diagnostics results, enter the following command on the Master node.

```
device(config)# show metro 2 diag
Metro Ring 2 - CustomerA
```

```
=============
diagnostics results
Ring       Diag       RHP average     Recommended     Recommended
id         state      time(microsec)  hello time(ms)  Prefwing time(ms)
2          enabled    125             100             300
Diag frame sent     Diag frame lost
1230                0
```

Syntax: show metro ring-id diag

If the recommended hello time and preforwarding time are different from the actual settings and you want to change them, refer to Configuring MRP on page 353.

Displaying MRP information

You can display the following MRP information:

- Topology group ID associated with the MRP ring
- Ring configuration information and statistics

Displaying topology group information

To display topology group information, enter the following command.

Syntax: show topology-group [**group-id**]

Displaying ring information

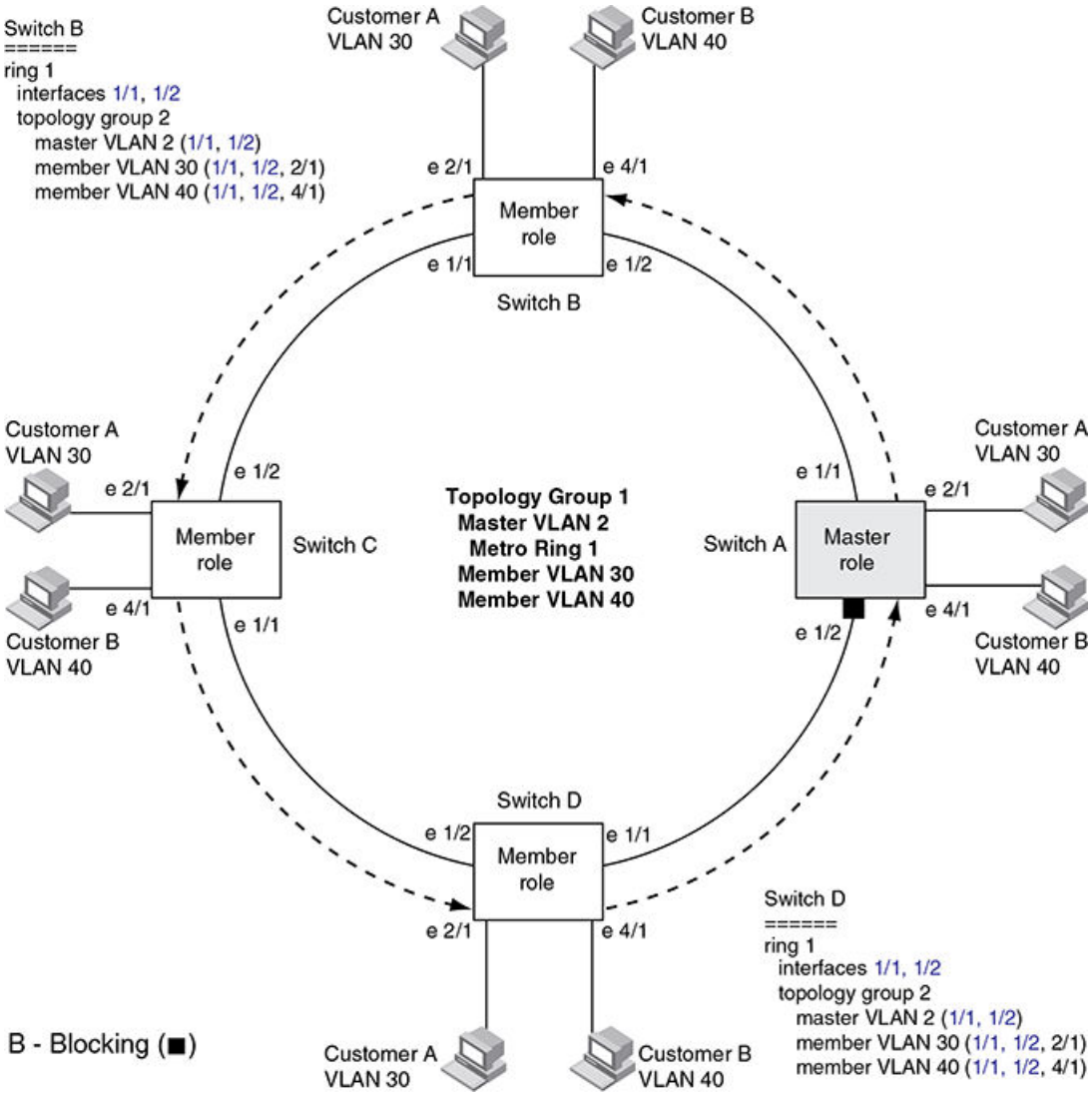
To display ring information, enter the following command.

```
device(config)# show metro-ring 10
Metro Ring 10 - VLAN Type REGULAR
==============
Ring       State      Ring       Master     Topo       Hello      Prefwing
id                    role       vlan       group      time(ms)   time(ms)
10         enabled    member     7          1          100        300
Ring interfaces Interface role  Interface state interface type
ethernet 1/1    primary         forwarding      regular
ethernet 30/1   secondary       forwarding      regular
RHPs sent       RHPs rcvd       TC rcvd    TC sent    State changes
0               0               69         0          6
```

Syntax: show metro-ring *ring-id*

MRP CLI example

The following examples show the CLI commands required to implement the MRP configuration shown in Figure 117.



NOTE

For simplicity, the figure shows the vlans on only two switches. The CLI examples implement the ring on all four switches.

Commands on Switch A (master node)

The following commands configure a vlan for the ring. The ring vlan must contain both of the node's interfaces with the ring. Add these interfaces as tagged interfaces, since the interfaces also must be in each of the customer vlans configured on the node.

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# master
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2-mrp-1)# exit
device(config-vlan-2)# exit
```

The following commands configure the customer vlans. The customer vlans must contain both the ring interfaces as well as the customer interfaces.

```
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(config-vlan-40)# exit
```

The following commands configure topology group 1 on vlan 2. The master vlan is the one that contains the MRP configuration. The member vlans use the MRP parameters of the master vlan. The control interfaces (the ones shared by the master vlan and member vlan) also share MRP state.

```
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
device(config-topo-group-1)# member-vlan 30
device(config-topo-group-1)# member-vlan 40
```

Commands on Switch B

The commands for configuring switches B, C, and D are similar to the commands for configuring Switch A, with two differences: the nodes are not configured to be the ring master. Omitting the **master** command is required for non-master nodes.

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2)# exit
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(config-vlan-40)# exit
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
device(config-topo-group-1)# member-vlan 30
device(config-topo-group-1)# member-vlan 40
```

Commands on Switch C

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2)# exit
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(config-vlan-40)# exit
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
```

```
device(config-topo-group-1)# member-vlan 30
device(config-topo-group-1)# member-vlan 40
```

Commands on Switch D

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2)# exit
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(config-vlan-40)# exit
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
device(config-topo-group-1)# member-vlan 30
device(config-topo-group-1)# member-vlan 40
```

Configuring MRP under an ESI VLAN

MRP can also be configured under a vlan that is part of a user-configured ESI. Configuring MRP in this scenario is exactly the same as explained before.

```
device(config)# esi customer1 encapsulation cvlan
device(config-esi-customer1)# vlan 100
device(config-esi-customer1-vlan-100)# tag ethernet 1/1 to 1/2
device(config-esi-customer1-vlan-100)# metro-ring 1
device(config-esi-customer1-vlan-100-mrp-1)# name "Metro A"
device(config-esi-customer1-vlan-100-mrp-1)# master
device(config-esi-customer1-vlan-100-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-esi-customer1-vlan-100-mrp-1)# enable
device(config-esi-customer1-vlan-100-mrp-1)# exit
device(config-esi-customer1-vlan-100)# exit
```

Configuration considerations

The configuration considerations are as follows:

- MRP can be configured for vlans with encapsulation type B-VLAN, S-VLAN or C-VLAN.
- When MRP is configured for vlans under an ESI, the MRP members must be part of the same ESI.

Ethernet Ring Protection Protocol

Ethernet Ring Protection Overview

Ethernet Ring Protection (ERP), a non-proprietary 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.

You can enable one instance of ERP on a device. Changes to a master VLAN apply to the member VLANs.

> 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.

Ethernet Ring Protection components

An ERP deployment consists of the following components:

- Roles assigned to devices, called Ethernet Ring Nodes (ERN)
- 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. See Figure 118.

FIGURE 116 ERP with IEEE 802.1ag support

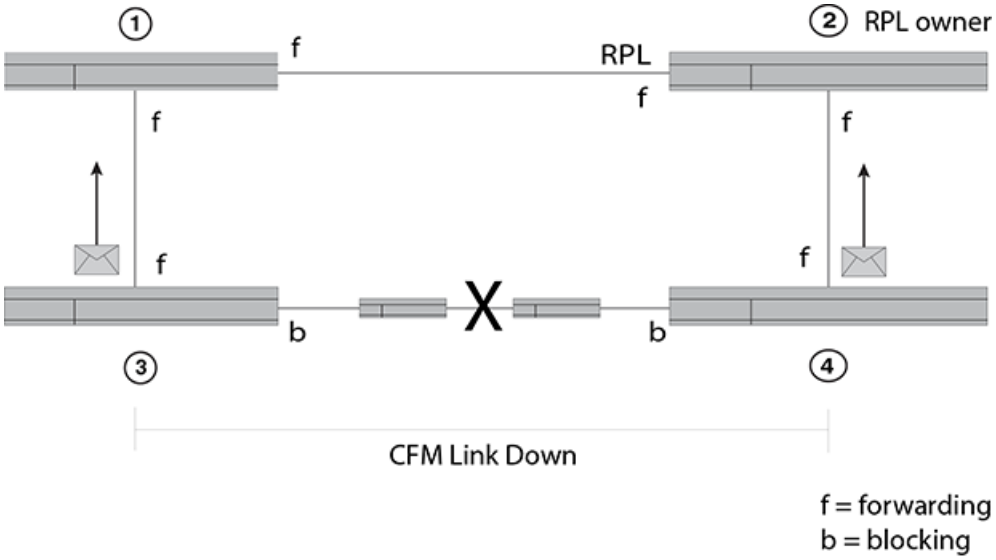


Figure 118 shows a segment with ERNs 3 and 4 and two non-participating switches located on the same network segment between them. When ERNs 3 and 4 stopped receiving CCMs, the following actions occurred on ERNs 3 and 4:

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. The figure below shows the general R-APS packet structure. For details about the packet structures, see ITU-T G.8032.

Figure to be added here for R-APS packet structure.

The destination MAC address (Dst Mac) is the first element in the packet and is of the form 00-00-00-00-00-<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 if using 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. ERP operational states shows the initialization process for an ERN.

Message exchange and actions during ERN initialization version 2

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** -- This wait-to-block 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 clearing a MS command, the WTB timer prevents the formation of a closed loop due to 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 happen.

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.

Figure 119 shows the first step of initialization beginning from ERN 4, a non-RPL node. The actions of each ERN are:

- 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.

FIGURE 117 Initializing an ERN topology - I

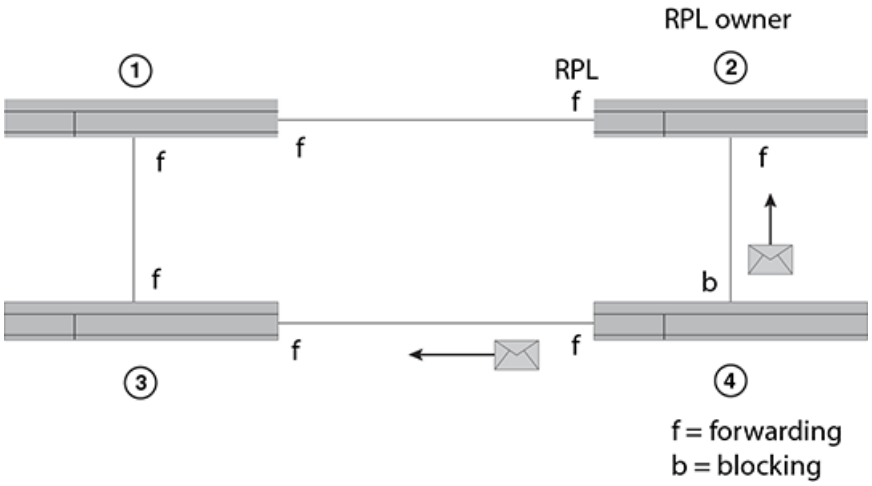


Figure 120 shows the next sequence of events. Next, ERN 1 initializes. The actions of each ERN are:

- 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.

FIGURE 118 Initializing an ERP topology - II

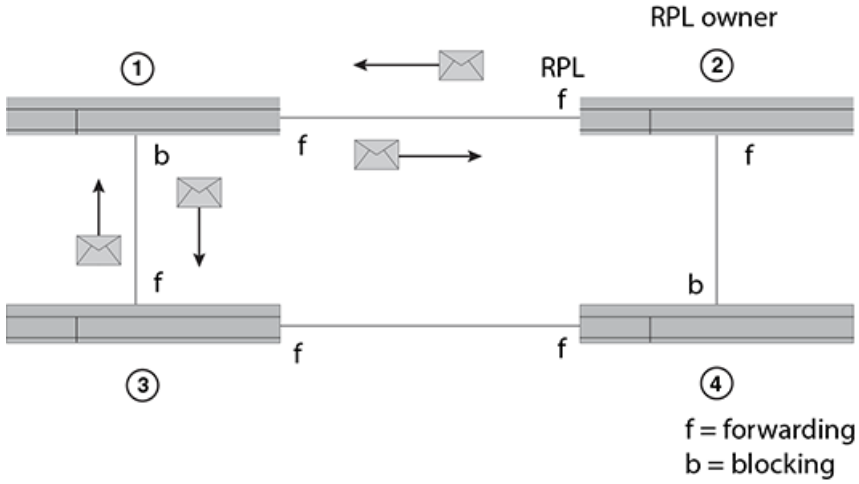


Figure 121 shows the next sequence of events. The actions of each ERN are:

- 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.

FIGURE 119 Initializing an ERP topology - III

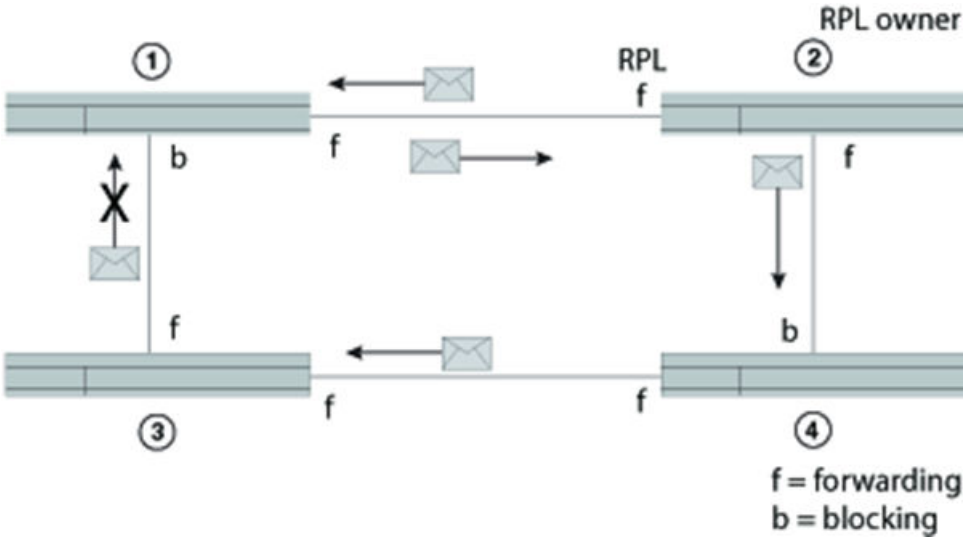


Figure 122 shows the next sequence of events. The actions of each ERN are:

- 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.

FIGURE 120 Initializing an ERP topology - IV

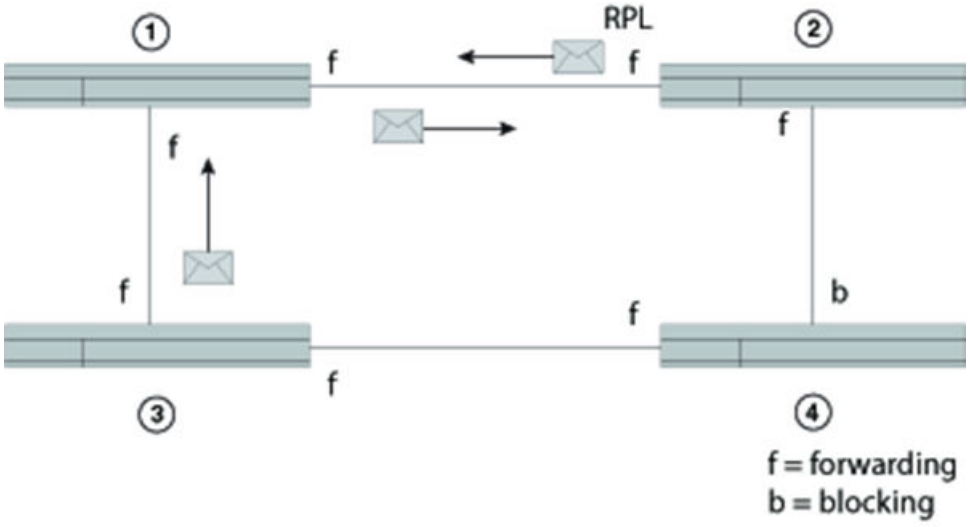


Figure 123 shows the next sequence of events. Next ERN 2 initializes. The actions of each ERN are:

- 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.

FIGURE 121 Initializing an ERP topology - V

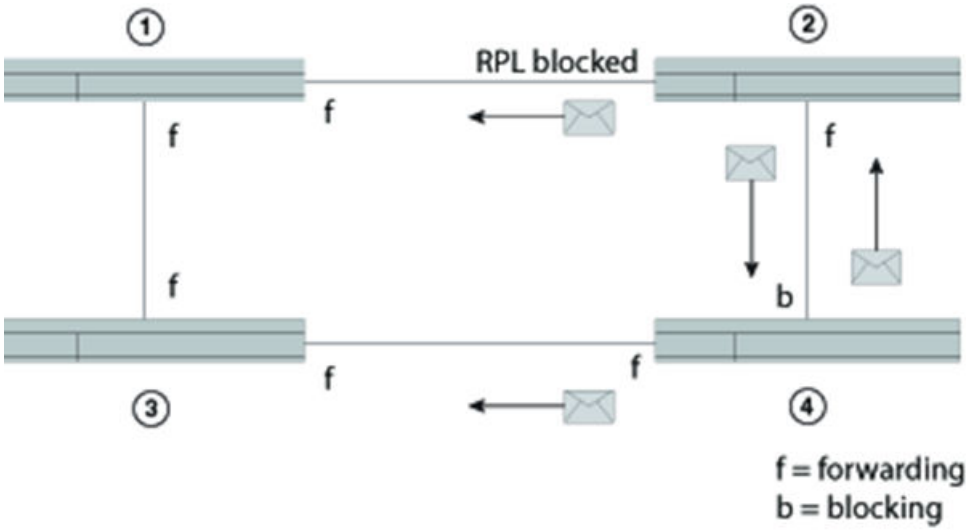
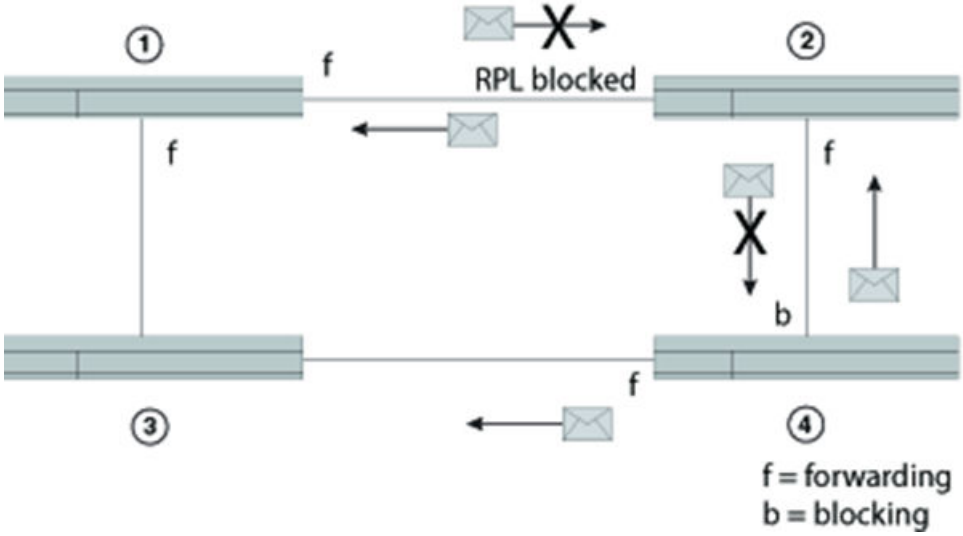


Figure 124 shows the next sequence of events. The actions of each ERN are:

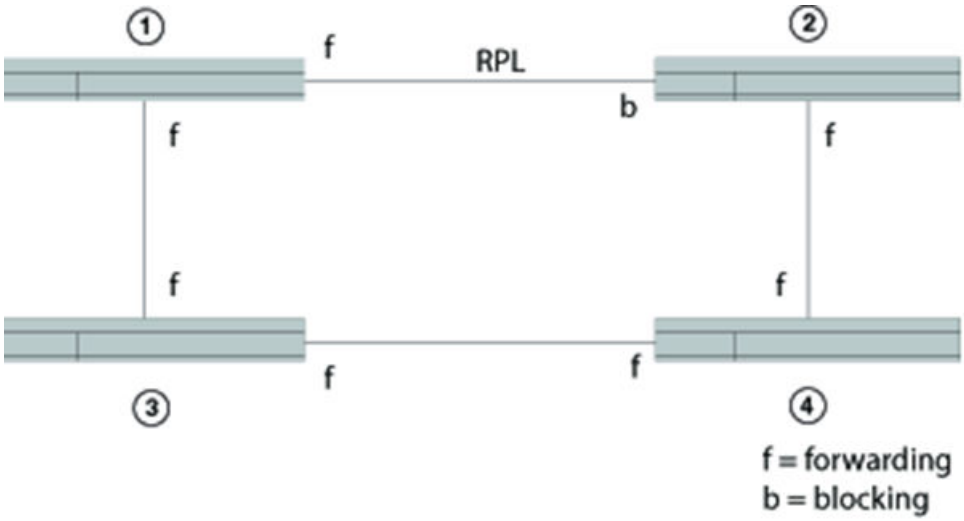
- After the WTB timer expires, ERN 2 (RPL owner in the Pending state) transmits R-APS (NR, RB), and then ERN 2 enters the Idle state.
- ERN 1, still in the Pending state, forwards R-APS (NR, RB) and enters the Idle state.
- ERN 3 takes no action.
- ERN 4 from the Pending state and stops transmitting R-APS (NR).

FIGURE 122 Initializing an ERP topology - VI



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. See Figure 125.

FIGURE 123 Initializing an ERP topology - VII



Signal fail

Signal fail and signal fail 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 signal failure 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.

Figure 126 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 124 ERP topology

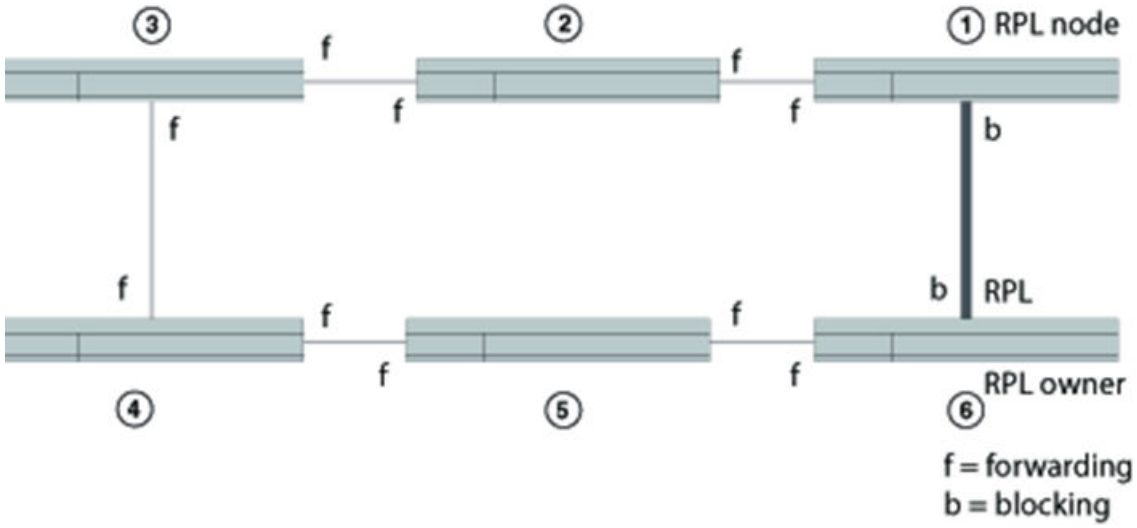
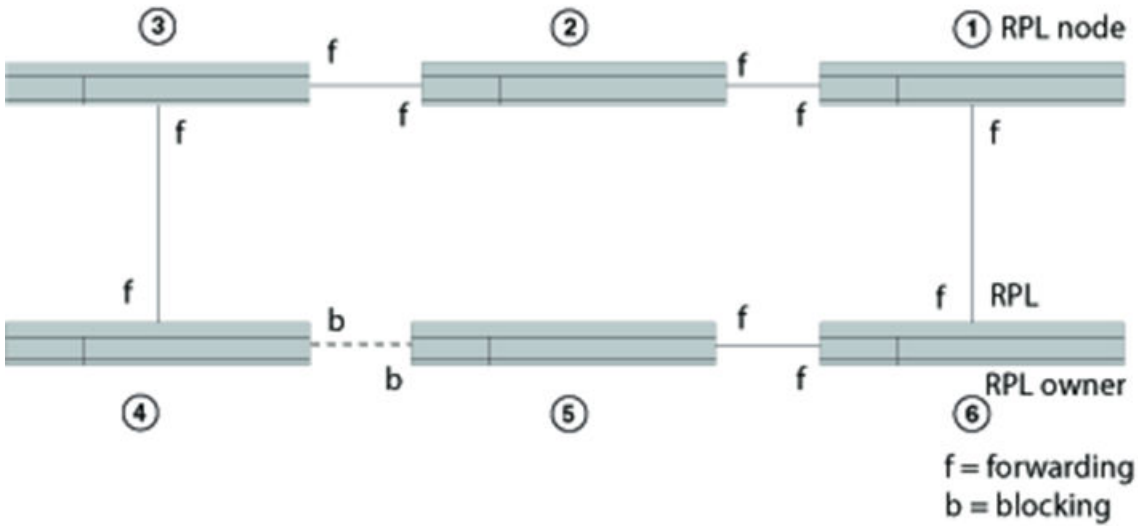


Figure 127 shows the same Ethernet ring topology after a failure at the forwarding port of ERN 4 when a signal fail triggered, and ring protection was needed. ERN 6 unblocked the RPL port and the RPL node changed the blocking port to the forwarding state.

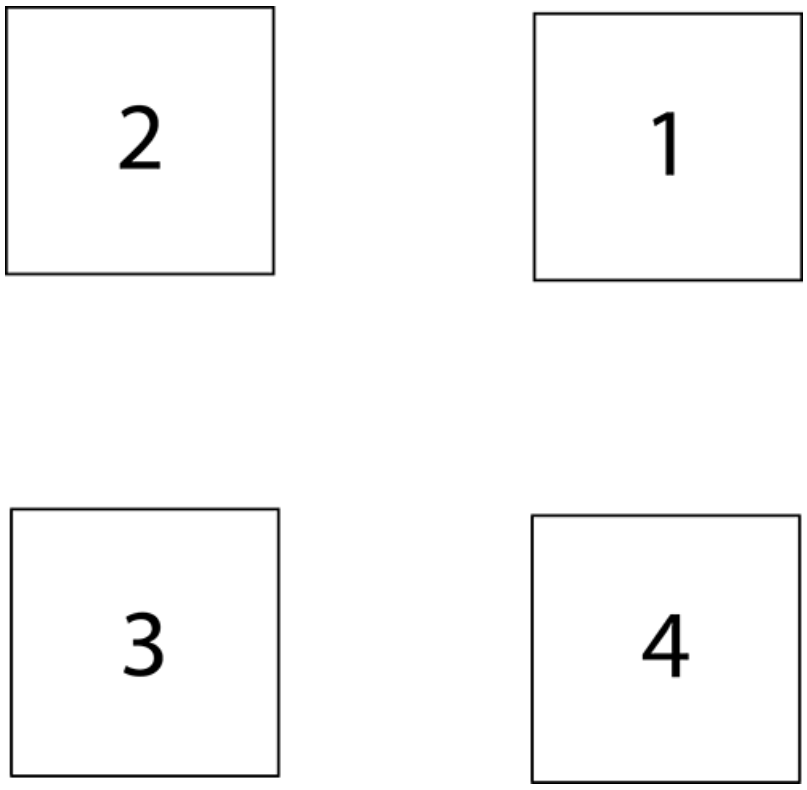
FIGURE 125 ERP topology in a Protected state



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. See Figure 128. The thick line between ERNs 1 and 2 indicate that dual-end blocking is enabled.

FIGURE 126 Manual Switch example



The node, which 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 with the **no** command.

> 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.

Figure 128 shows a manual switch on ERN 3, which is a non-RPL node. In order to clear the MS condition, the operator must enter the manual switch command from ERN 3. The sequence of messages and actions is as shown in Manual switch.

MS on Non-RPL node

Non-RPL node with error (ERN 3)	RPL owner (ERN 1) and RPL node (ERN2)	Other Non-RPL node (ERN 4)
From the Idle state, ERN3: 1. Blocks the MS port 2. Sends the RAPS (MS) 3. Flushes the FDB		

Non-RPL node with error (ERN 3)	RPL owner (ERN 1) and RPL node (ERN2)	Other Non-RPL node (ERN 4)
4. Enters the manual switch (MS) state		
		From the Idle state, ERN 4: 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. Manual switch shows the sequence of events during the MS recovery process.

MS recovery process

Non-RPL node with error (ERN 3)	RPL owner (ERN 1)	RPL node (ERN2) with dual-end blocking enabled	Non-RPL node (ERN 4)
From the MS state, ERN 3: 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, ERN 1: 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, ERN 2: 1. Receives the R-APS (NR) 2. Forwards the R-APS (NR) 3. Enters the Pending state	From the MS state, ERN 2: 1. Receives the R-APS (NR) 2. Forwards the R-APS (NR) 3. Enters the Pending state
From the Pending state, ERN 3: 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, ERN 2: 4. Blocks the RPL 5. Forwards the R-APS (NR, RB) 6. Flushes the FDB 7. Enters the Idle state	From the Pending state, ERN 4: 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.

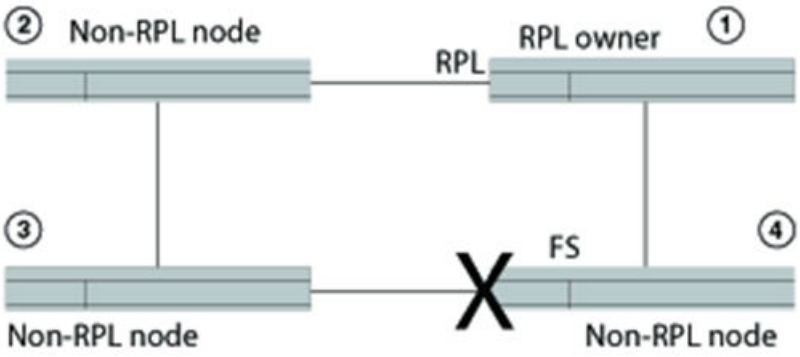
An FS request can be accepted no matter what state the topology is in. Since the local FS and R-APS (FS) are higher priority than SF; an SF occur i ng later than FS will not trigger the SF process. In addition, because the local FS and R-APS (FS) are higher priority than SF, when a node receives a R-APS (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.

Figure 129 shows a port failure on ERN 4.

FIGURE 127 Single forced switch scenario



Forced switch shows the sequential order of events triggered as a result of an operator-initiated forced switch command entered from ERN 4.

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. Forced switch shows the forced switch recovery process in sequential order.

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	From FS state, ERN 3: 1. Forwards R-APS	

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
	2. Starts the guard timer 3. Enters the Pending state	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 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

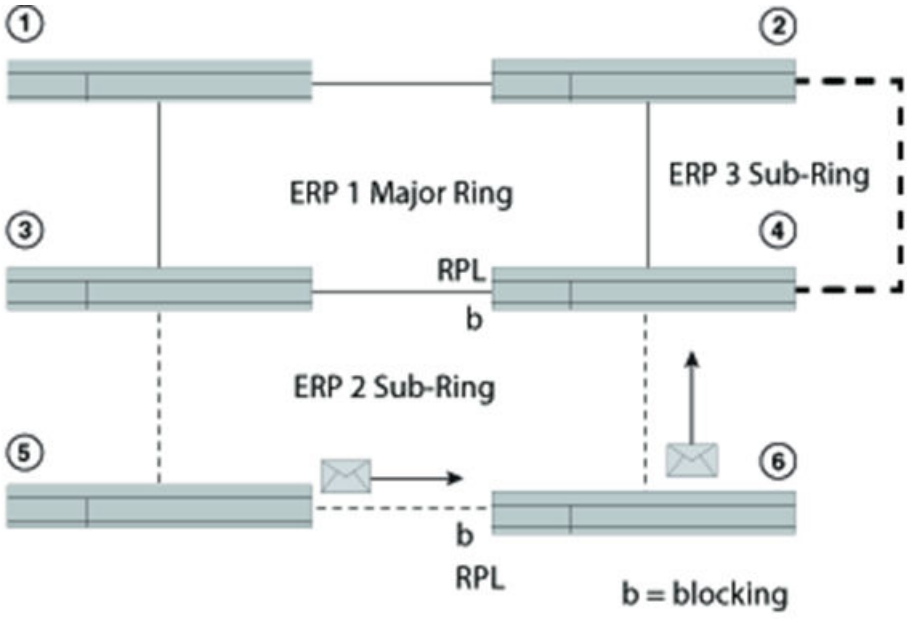
In non-revertive mode, the traffic channel is allowed to use the RPL, if it is not failed, after a switch condition clears. In the recovery from a Protection state, the RPL owner generates no response regarding the reception of NR messages. When other healthy nodes receive the NR message, there is no 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.

The dotted lines in Figure 130 show two of the many potential sub-rings that you can configure.

FIGURE 128 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.

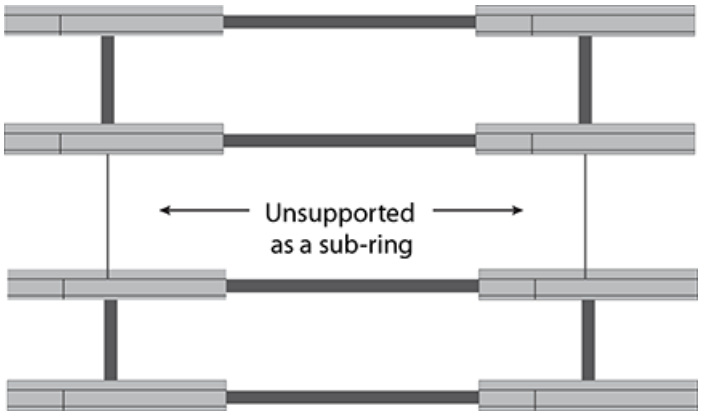
NOTE
ERP Virtual channel support is no longer supported.

A sub-ring in segments interconnecting major rings is not supported. Figure 131 shows a major ring and two segments not supported as a sub-ring.

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.

FIGURE 129 Unsupported sub-ring in segments



FDB flush optimization

The FDB stores the node ID and BPR sent in the R-APS messages. When an ERN receives a new R-APS message, it compares the received node ID and BPR to the node ID and BPR in its memory. If the pair vary from the previously stored pair, the ERN deletes the previous pair and stores the new pair. The device then triggers a FDB flush unless the DNF (No Not Flush) is set in the message.

FDB optimization is achieved with the following features:

- Non-revertive mode alleviates the need to flush the FDB after a link failure with link protection (SF) condition
- Dual-end blocking decreases attempted messages and traffic to the RPL blocking port
- Interconnected ring support to decrease the latency for messaging
- Do Not Flush (DNF) messages

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 only 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

Sample configuration

The following example is of an ERP configuration consisting of four devices: an RPL owner, an RPL node, and two non-RPL nodes.

> NOTE
> Before configuring any ERP settings, configure the VLAN and ports.

Device 1 RPL owner

> NOTE
> Optionally, you can configure the non-revertive mode feature. This setting can only be set on the RPL owner.

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#rpl vlan 2 e1/2
(config-erp-1)#rpl-owner
(config-erp-1)#enable
```

Device 2 RPL node

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#rpl vlan 2 e 1/1
(config-erp-1)#enable
```

Device 3 Non-RPL node

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#enable
```

Device 4 Non-RPL node

(config)# erp 1

(config-erp-1)#right-interface vlan 2 e 1/2

(config-erp-1)#left-interface vlan 2 e 1/1

(config-erp-1)#enable

Configuring ERP with IEEE 802.1ag

To configure and initialize ERP using APS and IEEE 802.1ag 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 (MEP) 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

ERP commands

This section lists ERP configuration commands.

Assigning ERP IDs

You must assign an ERP ID. This ID number is used to:

- 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*

For example, to assign the number 10 to the ERP, enter:

```
(config)# erp 10
```

Naming an Ethernet Ring Node

From within the ERP configuration shell, you can optionally name an ERN with a meaningful name. The name must be 31 alphanumeric characters or fewer; and the name can use the "underscore" and "dash" special characters.

Syntax: [no] name *erp_name*

For example, to assign the name "to_extreme1" to an ERN with ID number 10, enter:

device `(config)# erp 10`

device `(config-erp-10)# name "to_extreme1"`

Use the **no** 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 (00-00-00-00-00- <01 or ERP ID>) in R-APS messages. By default 00-00-00-00-00-01 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.

For more information about feature support for version 1 and 2, see Setting the ITU-T G.8032 version number on page 399.

Syntax: [**no**] **raps-default-mac**

Configuring R-APS MEL value

The R-APS Maintenance Entity Group Level (MEL) value can be configured. The R-APS MEL value is carried in ERP PDUs. The default R-APS MEL value is 7.

Syntax: [**no**] **raps-mel mel value**

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 transmission of RAPS (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.

Syntax: [**no**] **raps-propagate-tc**

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, in the major ring, you must first configure two interfaces. In a sub-ring, at least one interface must first be configured before enabling the ERP instance.

Syntax: [no] enable

Example of a non-RPL node configuration in a major ring:

(config)# erp 1

(config-erp-1)#right-interface vlan 2 e 1/2

(config-erp-1)#left-interface vlan 2 e 1/1

```
(config-erp-1)#enable
```

Configuring interfaces

Each ERN in a major ring must have explicitly defined left and right interfaces so that ERP can function properly. ERNs in a sub-ring must have at least one interface defined so that ERP can function properly.

For proper operation you must configure the interfaces following the same manner on each ERN, such as left/ right, left/ right, and so on.

Syntax: [no] left-interface [**vlan** *vlan-id* | **esi esi_name vlan vlan_id**] **e** *slot/ number*

Syntax: [no] right-interface [**vlan** *vlan-id* | **esi esi_name vlan vlan_id**] **e** *slot/ number*]

Use the **no** command to remove the configuration of each interface.

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, you next must set the RPL. To set the RPL you need to specify the VLAN and Ethernet slot and port.

> NOTE
> When you assign the role of RPL owner, you must also configure the RPL.

Syntax: [no] rpl-owner

Syntax: [no] rpl [**vlan** *vlan-id* | **esi esi_name vlan vlan_id**] **e** *slot/number*

Enabling sub-rings for multi-ring and ladder topologies

In multi-ring and ladder topologies, you can enable the multi-ring feature.

Interconnected rings consists of one major ring and at least one sub-ring within the same VLAN or different VLANs. 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 need to be configured in a sub-ring. All ERP features are available in both major and sub-rings.

R-APS PDUs only flow in the nodes with same ring ID. The R-APS PDU can be forwarded through the port in sub-ring blocking state.

Syntax: [no] sub-ring [**parent-ring-id erp-id**]

The **parent-ring-id** is used when there are different VLANs on the major ring and sub-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 major ring connected to the sub-ring.

Use the **no** 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.

Achieving sub-50ms ring protection switch time

The G.8032v2 ERP implementation has been enhanced to achieve sub-50ms ring protection switch time. This enhancement involves optimizations to reduce the number of MAC flushes, temporary flooding of traffic while MAC flush is in progress, and faster link failure detection using Connectivity Fault Management (CFM).

You will need to configure the following to achieve sub-50ms ring protection switch time.

Enter the following command to allow temporary flooding of traffic during MAC flush. Use the **no** version of this command to disable the flooding of traffic.

```
device(config-erp-1)#flooding-enable
```

Syntax: [no] flooding-enable

During topology change, there are multiple MAC flushes triggered by ERP protocol. Optimizations have been made to ERP protocol to reduce the number of MAC flushes to achieve faster convergence. These optimization can be enabled using the **fdb-flush-optimization** command. Use the **no** version of this command to disable the flush optimization.

```
device(config-erp-1)#fdb-flush-optimization
```

Syntax: [no] fdb-flush-optimization

IEEE 802.1ag can be used to monitor the ERP interfaces for signal failures.The **dot1ag-compliance** command allows MD and MA's configured as part of IEEE 802.1ag to be associated with an ERP instance. Use the **no** version of this command to disable the **dot1ag-compliance** command.

```
device(config-erp-1)#dot1ag-compliance domain-name erp ma-name ma-erp
```

Syntax: [no] dot1ag-compliance domain-name *domain-name* **ma-name** *ma-name*

The **domain-name** parameter specifies the maintenance domain name for 802.1ag CFM.

The **ma-name** parameter specifies the maintenance association name. This can be up to 21 characters long.

Configuration recommendations for dot1ag:

- Dot1ag configuration for ERP is recommended only for copper ports as link failure detection time is higher when compared to fiber ports.
- If dot1ag is configured, it is recommended to ensure broadcast traffic in the network does not cross 100% of link bandwidth utilization. If the broadcast traffic crosses 100% of the link bandwidth utilization, then CCM packets might be lost which can result in MEPs moving to failed state causing ERP state fluctuations.

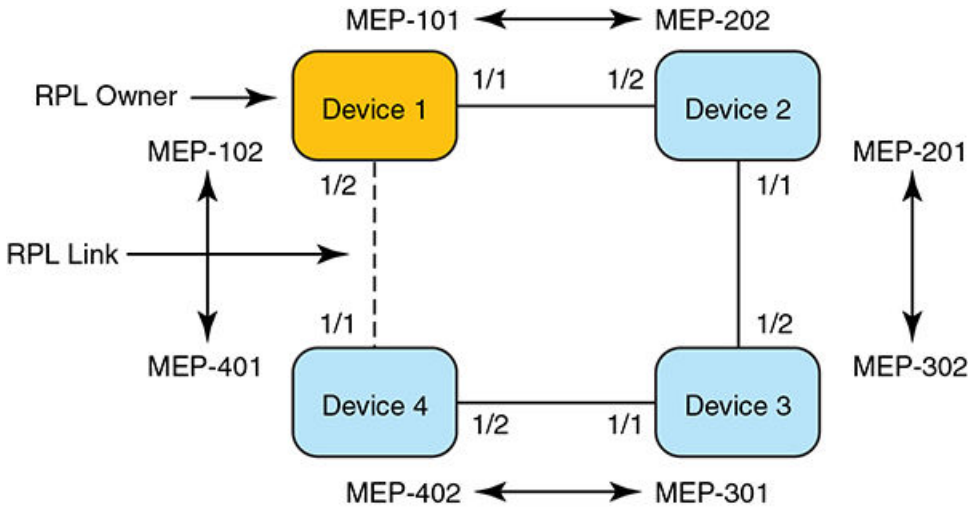
  NOTE
  Sub-50ms convergence time may not be achievable with LAG interfaces.

This command needs to be enabled under the link MA configuration of IEEE 802.1ag. This allows configuring MEPs to individual links of the LAG, and enables monitoring of each member link of that LAG. Use the **no** version of this command to disable the individual link monitoring.

```
device(config-cfm-md-erp-ma-ma-erp)#individual-link-monitoring
```

Syntax: [no] individual-link-monitoring

FIGURE 130 Network diagram for ERP



Configuration example

NOTE
The VLAN CPU protection needs to be enabled on the VLANs.

Device 1 Configuration steps

CFM configuration:

```
device#configure terminal
device(config)#cfm-enable
device(config-cfm)#domain-name erp id 1 level 1
device(config-cfm-md-erp)#ma-name ma-erp link-ma priority 7
device(config-cfm-md-erp-ma-ma-erp)#individual-link-monitoring
device(config-cfm-md-erp-ma-ma-erp)#mep 101 down port eth 1/1
device(config-cfm-md-erp-ma-ma-erp)#mep 102 down port eth 1/2
```

VLAN Configuration:

```
device(config)#vlan 100
device(config-vlan-100)#tag eth 1/1 eth 1/2
device(config-vlan-100)#vlan-cpu-protection
```

ERP Configuration:

```
device(config)#erp 1
device(config-erp-1)#left-interface vlan 100 eth 1/1
device(config-erp-1)#right-interface vlan 100 eth 1/2
device(config-erp-1)#rpl-owner
device(config-erp-1)#rpl vlan 100 eth 1/2
device(config-erp-1)#flooding-enable
device(config-erp-1)#fdb-flush-optimization
device(config-erp-1)#dot1ag-compliance domain-name erp ma-name ma-erp
device(config-erp-1)#enable
```

Device 2 Configuration steps

CFM Configuration:

```
device#configure terminal
device(config)#cfm-enable
device(config-cfm)#domain-name erp id 1 level 1
device(config-cfm-md-erp)#ma-name ma-erp link-ma priority 7
device(config-cfm-md-erp-ma-ma-erp)#individual-link-monitoring
device(config-cfm-md-erp-ma-ma-erp)#mep 201 down port eth 1/1
device(config-cfm-md-erp-ma-ma-erp)#mep 202 down port eth 1/2
```

VLAN Configuration:

```
device(config)#vlan 100
device(config-vlan-100)#tag eth 1/1 eth 1/2
device(config-vlan-100)#vlan-cpu-protection
```

ERP Configuration:

```
device(config)#erp 1
device(config-erp-1)#left-interface vlan 100 eth 1/1
device(config-erp-1)#right-interface vlan 100 eth 1/2
device(config-erp-1)#flooding-enable
device(config-erp-1)#fdb-flush-optimization
device(config-erp-1)#dot1ag-compliance domain-name erp ma-name ma-erp
device(config-erp-1)#enable
```

Configuring non-revertive mode

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.

Syntax: [no] non-revertive-mode

Use the **no** command to remove the non-revertive mode setting.

Configuring and clearing a forced switch

An operator can use the forced switch (FS) mechanism 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.

Syntax: [no] forced-switch [**vlan** *vlan-id* | **esi** *esi_name* **vlan** *vlan_id*] **e** *slot/port*]

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 that manually blocks a desired port in a ring. You need to explicitly specify the VLAN, Ethernet slot, and port from the desired device.

Syntax: [no] manual-switch [**vlan** *vlan* | **esi** *esi_name* **vlan** *vlan_id*] **e** *slot/port*]

Use the **no manual-switch** command to remove the manual switch mechanism.

Configuring dual-end blocking

You can configure dual-end blocking to optimize your ERP configuration. The RPL node must be adjacent to the RPL owner.

When you configure the RPL on an ERN that is adjacent to the RPL owner, you are enabling the dual-end blocking feature and changing the ERN's role to that of RPL node. You configure the RPL node with the **rpl** command. Before configuring dual-end blocking, you must verify that the RPL node is actually the correct peer and obtain the RPL link settings; an incorrect setting will cause incorrect port blocking.

> NOTE
> The RPL node must be a peer of the RPL owner, and the RPL must be configured on this peer; otherwise, the device will perform incorrect port blocking behavior.

Syntax: [no] rpl [**vlan** *vlan-id* | **esi** *esi_name* **vlan** *vlan_id slot/number*]

Configuring the guard timer

The guard timer 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.

Syntax: guard-time *time-value*

Configuring and clearing the wait to restore timer

For SF recovery situations, you can 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 command can be used to stop the WTR timer. While 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 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.

This timer can be stopped by issuing the **clear erp** command.

Syntax: wtr-time *time-value*

Testing the WTR timer

You can enter the **fast-wtr-time** command to test your configuration. Instead of having to wait 5 minutes for the timer to expire, you wait 5 seconds. This command changes the timer's unit of measure from minutes to seconds.

Syntax: [no] fast-wtr-time

Use the **no** command to return the unit of measure to minutes.

Configuring and clearing the WTB timer

The WTB timer ensures that clearing of a single FS command does not trigger the reblocking of the RPL when multiple FS situations co-exist in an Ethernet Ring. When recovering from an 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 5000ms (5 seconds) longer than the guard timer. You can configure this timer in 100 ms increments from 5100ms to 7000ms (7 seconds); the default value is 5500ms.

The WTB timer can be stopped through the CLI by entering the **clear erp***erp_id***wtb-timer** command.

Syntax: wtb-time *time-value*

Configuring a hold-off timer

The hold-off timer is used in each ERN to prevent unnecessary Signal Fail events due to port flapping. If you configure a non-zero hold-off timer value, when a link error occurs, the event will not be reported immediately. When the hold-off timer expires, ERP checks if the error still exists.

The hold-off timer is used in every ERN. When a new defect occurs (new SF), this event will not be 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 100ms 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.

Syntax: holdoff-time *time-value*

Configuring the message interval time

The message interval time of R-APS messages continuously sent within an ERP ring can be configured. You can configure the interval in 100ms increments from 100ms to 5000ms (5 seconds); the default value is 5000ms.

Syntax: message-interval *time-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. Setting the ITU-T G.8032 version number lists the feature and MAC ID differences between versions 1 and 2.

NOTE
The ERP **version** command does not have a shortened form. You must enter the complete command.

1. Signal Fail Signal Fail recovery
2. Always uses 01:19:A7:00:00:01 as the ERP ID in R-APS messages
3. Signal Fail Signal Fail recovery Manual Switch Forced Switch Non-revertive Interconnected rings RPL configuration on non-RPL owner
4. Allows use of the ERP ID for the last two bytes of the MAC ID (01:19:A7:00:00:erp-id)

Syntax: version *version_number*

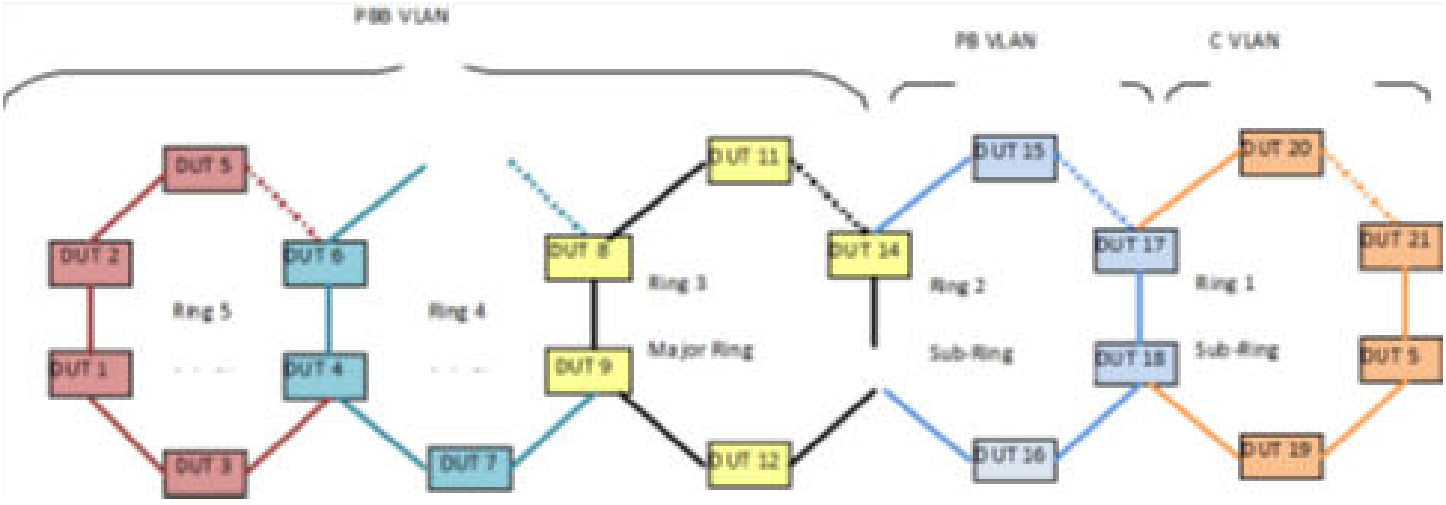
You can view the version by entering the **show erp** command. The version appears on the top line directly after the ERP ID.

ERP over ESI VLAN (CES 2000 Series and CER 2000 Series devices)

Figure 133 shows a diagram of one of the sample topologies that is used to explain deployment of ERP over PBB using the ESI model. In the diagram, Ring 3 is the major ring while all others are sub-rings. Each ESI VLAN will be running a separate instance of ERP. Any ring can act as a major ring. However, it is recommended that the B-VLAN be used as a part of Major ring.

When a network involves PB and PBB rings, one of the PBB ring must be configured as a major ring. In the case of a PB only network, a PB ring can be the major ring. There can be only one major ring in a given network.

FIGURE 131 ERP configuration over ESI network



In general scenarios of ERP, a multiple VLANs span across multiple rings forming a major ring and sub-rings. The major ring and sub-ring is determined by the ERP configuration and the ERP protocol operates keeping in focus the traffic flow.

Interconnection rings with different VLANs

In the ESI model, traffic flow is determined by the mapping of one ESI VLAN relationship to another ESI VLAN. In ERP over ESI VLAN (CES 2000 Series and CER 2000 Series devices) on page 399, the ESI CVLAN1 is a client for the ESI SVLAN1 instance. So traffic from Ring 1(CVLAN1) gets encapsulated with an S-TAG (SVLAN1) in Ring 2. Similarly, ESI SVLAN1 is a client for the ESI BVLAN1 instance, resulting in traffic from Ring 2 encapsulated with a PBB header in Ring 3. The traffic forwarding from Ring 3 towards Ring 4 and Ring 5 occurs as in any normal VLAN as all the rings are running on same VLAN (BVLAN1).

To support the ERP over ESI model, each ring needs to run its own ERP instance with an ESI VLAN. For Ring 1, the ERP instance needs to run on CVLAN1, for Ring 2 on SVLAN1 and for Rings 3, 4 and 5 on BVLAN1. Some of the nodes, such as DUT17 and DUT18, connect one ERP ring to another ERP ring. These are called interconnection nodes. Each interconnection node in ERP over ESI VLAN (CES 2000 Series and CER 2000 Series devices) on page 399, runs two instances of ERP. For example, in the case of DUT17 one ERP instance is run for Ring 1 on CVLAN1 and another ERP instance for Ring 2 on SVLAN1.

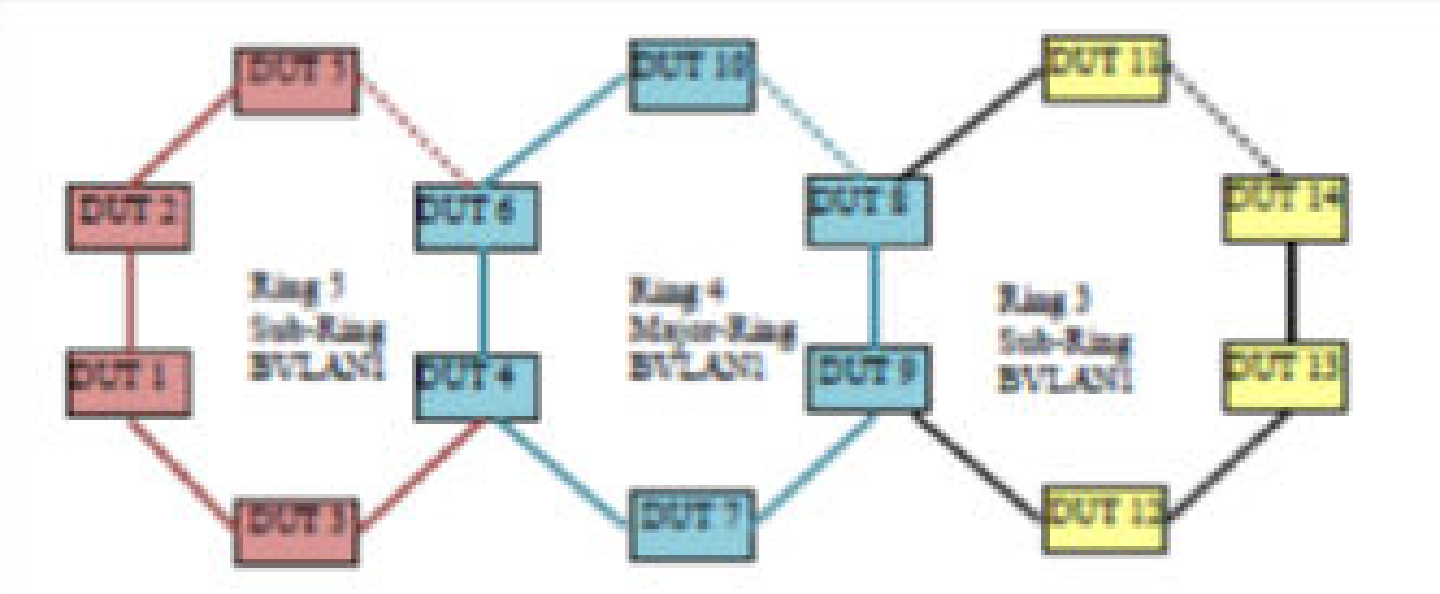
The interconnection nodes play a vital role in traffic forwarding and should be aware of ERP instances associated with each other within the node. In ERP over ESI VLAN (CES 2000 Series and CER 2000 Series devices) on page 399, the DUT17 interconnection node needs to be aware of Ring 1 being a sub-ring of the Ring 2 ERP instance. This association is important as it is required to perform a FDB flush in Ring-2 on receiving R-APS (SF) messages from Ring-1 .

Interconnection rings with same VLANs

The PBB case such as in Figure 134, where one B-VLAN span across multiple rings, the ring mapping will still be maintained as it helps push the events from sub-rings to major ring.

The PB case will be also handled similarly.

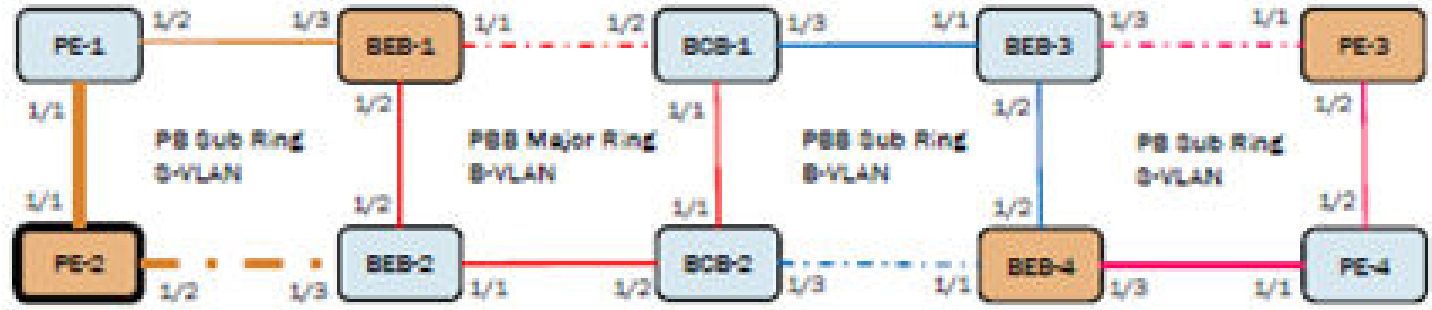
FIGURE 132 ERP configuration with same VLAN



Sample configurations

PB ring node

FIGURE 133 PB ring node sample configuration



PE-2 Configuration with S-VLAN 200:

```
!
esi svlan encapsulation svlan
vlan 200
tagged ethe 1/1 ethe 1/2
!
erp 200
left-interface esi svlan vlan 200 ethe 1/1
right-interface esi svlan vlan 200 ethe 1/2
rpl-owner
sub-ring
rpl esi svlan vlan 200 ethe 1/2
enable
!
interface ethernet 1/1
port-type provider-network
enable
!
interface ethernet 1/2
port-type provider-network
enable
```

PBB interconection node (BEB)

FIGURE 134 PBB interconection node (BEB) sample configuration

BEB-1 Configuration for Major ring B-VLAN 100

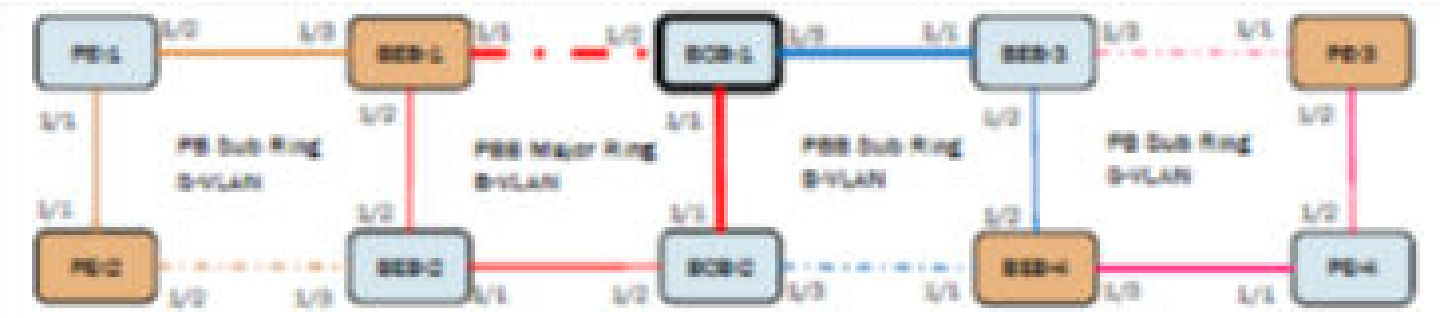
```
!
esi bvlan encapsulation bvlan
vlan 100
tagged ethe 1/1 ethe 1/2
!
erp 100
left-interface esi bvlan vlan 100 ethe 1/1
right-interface esi bvlan vlan 100 ethe 1/2
rpl-owner
raps-default-mac
rpl esi bvlan vlan 100 ethe 1/1
enable
!
interface ethernet 1/1
port-type backbone-network
enable
!
interface ethernet 1/2
port-type backbone-network
enable
```

BEB-1 Configuration for Sub-ring S-VLAN 200

```
!
esi svlan encapsulation svlan
vlan 200
tagged ethe 1/3
!
erp 200
right-interface esi svlan vlan 200 ethe 1/3
raps-default-mac
sub-ring parent-ring-id 100
enable
!
interface ethernet 1/3
port-type backbone-edge
enable
!
```

PBB interconection node (BCB)

FIGURE 135 PBB interconnection node (BCB)



BCB-1 Configuration for Major ring B-VLAN 100

```
!
esi bvlan encapsulation bvlan
vlan 100
```

```
tagged ethe 1/1 ethe 1/2
!
erp 100
left-interface esi bvlan vlan 100 ethe 1/1
right-interface esi bvlan vlan 100 ethe 1/2
raps-default-mac
rpl esi bvlan vlan 100 ethe 1/2
enable
!
interface ethernet 1/1
port-type backbone-network
enable
!
interface ethernet 1/2
port-type backbone-network
enable
```

ERP support for PBB (MLX Series and XMR Series devices)

To support ERP protocol over a PBB network on the MLX Series and XMR Series platforms will make use of topology groups. The ERP protocol will be run on a regular L2 VLAN which will be the master VLAN of the topology group and the VPLS VLANs which will carry the PB/PBB traffic will be member VLANs of the topology group.

Configuration requirements

- Configure regular L2 VLANs for ERP operation
- Configure PBB VPLS VLANs for carrying PB/PBB traffic
- Configure the topology group
  - ERP protocol over the master VLAN
  - PB/PBB traffic over the member VPLS VLANs

Blocking of L2 protocols for PBB

The configurations discussed will be blocked

- ERP is the only protocol that will be supported for PBB, any other protocol configuration with PBB is blocked.
- If a topology group is configured with PBB VPLS member VLANs, then only ERP can be enabled on the master VLAN.
- If the master VLAN of a topology group is enabled with a protocol other than ERP, then PBB VPLS member VLAN configuration will not be allowed on that topology group.

Sample configurations

PB Ring node

Figure to be added here for PB ring node

PE-2 ERP Configuration with regular VLAN 201

```
vlan 201
tagged ethe 1/1 ethe 1/2
!
erp 201
left-interface vlan 201 ethe 1/1
right-interface vlan 201 ethe 1/2
sub-ring
raps-default-mac
rpl vlan 201 ethe 1/2
enable
!
```

PE-2 Topology group configuration

```
!
topology-group 1
master-vlan 201
member-vlan vpls id 1 vlan 200
```

!

PE-2 PBB Configuration with S-VLAN 200

```
!
tag-type 88a8 eth 1/1
tag-type 88a8 eth 1/2
!
router mpls
    vpls pb-pbb 1
        pbb
        vlan 200
        tagged ethe 1/1 ethe 1/2
```

Figure to be added here for PE-2 PBB Configuration with S-VLAN

BEB-1 ERP PB sub-ring with regular VLAN 201

```
!
vlan 201
tagged ethe 1/3
!
erp 201
right-interface vlan 201 ethe 1/3
raps-default-mac
sub-ring parent-ring-id 100
enable
!
```

BEB-1 topology group for PB sub-ring

```
!
topology-group 1
master-vlan 201
member-vlan vpls id 1 vlan 200
!
```

BEB-1 PB configuration with S-VLAN 200 and PBB configuration with B-VLAN 100 and ISID 10000

```
!
vlan 100
```

```
tagged eth 1/1 eth 1/2
!
tag-type 88a8 eth 1/1
tag-type 88a8 eth 1/2
tag-type 88a8 eth 1/3
!
router mpls
vpls pb-pbb 1
pbb
vlan 200
tagged ethe 1/3
vlan 100 isid 10000
tagged ethe 1/1 to 1/2
!
```

PBB Interconnection node (BCB)

Figure to be added here for PBB Interconnection node (BCB)

BCB-1 Configuration for Major ring B-VLAN 100

```
!
vlan 100
tagged ethe 1/1 ethe 1/2
!
tag-type 88a8 eth 1/1
tag-type 88a8 eth 1/2
!
erp 100
left-interface vlan 100 ethe 1/1
right-interface vlan 100 ethe 1/2
raps-default-mac
rpl vlan 100 ethe 1/2
enable
!
```

BCB-1 Configuration for Sub ring B-VLAN 100

```
!
vlan 100
tagged ethe 1/3
!
tag-type 88a8 eth 1/3
!
erp 101
left-interface vlan 100 ethe 1/3
sub-ring parent-ring-id 100
raps-default-mac
enable
!
```

Parent ring ID configuration

Figure to be added here for Parent Ring ID configuration

- Each sub-ring will be connected to either a major ring or a sub-ring which is referred to as a parent ring.
- In the figure above, the ERP instance 1 is the parent ring for sub-ring ERP instance 2 and ERP instance 2 is the parent ring for sub-ring ERP instance 3.
- If both the major ring and sub-ring are running on the same VLAN then the parent ring can be identified based on the VLAN, However, if they are running on different VLANs (especially in PB/PBB networks) with multiple ERP instances then the administrator must specify the parent ring ID using the **sub-ring parent-ring-id** command.

- The sub-ring parent-ring-id must be configured on the sub-ring ERP instance of an interconnection node.

RAPS-propagate-tc

Figure to be added here for RAPS-propagate-tc

- The **RAPS-propagate-tc** command must be configured on the sub-ring ERP instance.
- When the **raps-propagate-tc** command is configured, any topology change on the sub-ring will be communicated to the major ring by sending a R-APS(Flush event). This results in a FDB flush on all the major ring nodes.
- The RAPS-propagate-tc configuration is required only when a node is present in between two interconnection nodes (for example: Node-4 in above topology).

Node-3 Configuration for Major ring VLAN 100

```
!
vlan 100
tagged ethe 1/1 ethe 1/3
!
erp 100
left-interface vlan 100 ethe 1/3
right-interface vlan 100 ethe 1/1
raps-default-mac
enable
!
```

Node-3 Configuration for Sub ring VLAN 100

```
!
vlan 100
tagged ethe 1/2
!
erp 200
right-interface vlan 100 ethe 1/2
raps-default-mac
sub-ring parent-ring-id 100
raps-propagate-tc
rpl vlan 100 ethe 1/2
enable
!
```

Figure to be added here for RAPS-propagate-tc configuration

A and B are end-points exchanging traffic, active path is as indicated by green line.

Figure to be added here for RAPS-propagate-tc topology change

- When the link connecting Node-1 and Node-2 goes down, it results in topology change on the sub-ring.
- To restore the traffic between endpoints A and B, an FDB flush is performed on all sub-ring nodes, interconnection nodes (Node-3 and Node-5) and Node-4.

The sequence of events for restoring the traffic

When the link connecting Node-1 and Node-2 goes down, it results in topology change on sub-ring resulting in following events:

- The topology change on the sub-ring results in the generation of R-APS(SF) PDUs by Node-1 and Node-2 in addition to flushing their own FDBs.
- Node-3 and Node-5 on the reception of R-APS(SF) perform an FDB flush on the sub-ring ERP interface 1/2.

- Node-3 and Node-5 will also perform an FDB flush on the major ring interfaces 1/1 and 1/3. The major ring on which the FDB needs to be flushed is identified by the configured parent-ring-id.
- If the **raps-propagate-tc** command is configured on the sub-ring instance of Node-3 and Node-5, it will result in the generation of R-APS(Flush event) PDUs on major ring interfaces, on reception of this PDU all major ring nodes will perform FDB flush.

After the above events, the active path changes as indicated by green line in the figure above.

In the figure above, only Node-4 needs to be flushed to restore traffic between endpoints A and B. However, other nodes (Node-6 and Node-7) will also flush due to reception of R-APS(Flush event) as per standard. Due to this, raps-propagate-tc must be configured only when a node (Node-4) exists between interconnection nodes (Node-3 and Node-5).

Viewing ERP operational status and clearing ERP statistics

You can view operational status and statistics and clear statistics for all links or particular links.

Viewing ERP operational status and statistics

To view ERP statistics, enter the following command on the RPL owner:

Syntax: show erp [*enter* | *erp_id*]

To view ERP information for all links, enter **show erp** command followed by pressing the Enter key (carriage return). To view statistics for a particular link, enter the ERP ID after the command.

Example output:

```
device #show erp 7
ERP 7(version 2)- VLAN 504
==========================================================================
Erp ID Status Oper Node Topo
state role group
1 enabled Idle rpl-owner -
Ring type WTR WTB Guard Holdoff Msg
time(min) time(ms) time(ms) time(ms) intv(ms)
Major-ring 5 7000 2000 0 1000
I/F Port ERP port state Interface status Interface type
L 1/12 blocking normal rpl
R 1/11 forwarding normal non-rpl
RAPS sent RAPS rcvd RAPS dropped RAPS ignored Oper state changes
3 3 0 0 0
```

Table 42 summarizes the table fields and their meanings.

TABLE 42 Summary of CLI output for show erp command

This field...	Displays...
ERP id	The ERP ID is the number that was configured at setup. The ERN appends this number to the permanent portion of the MAC address (01-19-A7-00-00) used for ERP.
Status	Enabled or disabled
Operational state	Init, Idle, Protection, Manual Switch, Forced Switch or Pending
Node role	rpl-owner, non-rpl-node or rpl-node
Topology group	<topology group id> or "- " (- means N/A)
Ring type	Major-ring or Sub-ring
Timers	Configuration value for each timer
Interfaces (I/F)	L (left) or R (right)
Port	<slot/port>
ERP port state:	disabled, blocking, forwarding
Interface status:	normal, signal-fail, manual-switch or forced-switch
Interface type:	rpl or non-rpl
RAPS sent:	RAPS sent by MP (self generated)
RAPS rcvd:	RAPS received by MP
RAPS dropped:	RAPS dropped by MP
RAPS ignored:	RAPS ignored (for example, the guard-timer, or non regular type)

Clearing ERP statistics

You can clear ERP statistics by entering the **clear erp statistics** command and the specific *erp_id* to clear the statistics of one erp instance. You can clear all ERP statistics by entering the **clear erp statistics** command to clear the statistics of all erp instances.

```
device#
clear erp
7 statistics
```

Syntax: clear erp *erp_id* **statistics**

Multi-Chassis Trunking (MCT)

About Multi-Chassis Trunk (MCT)

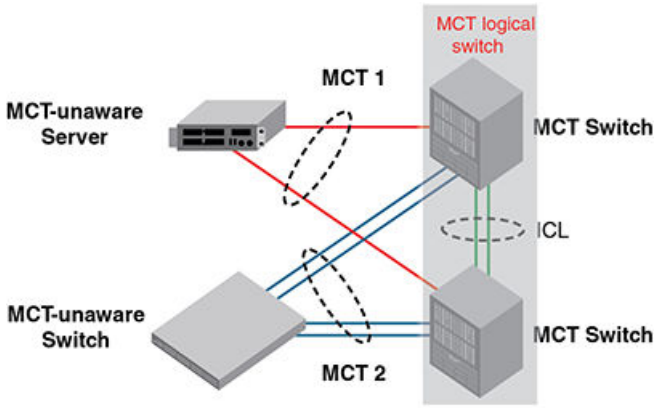
A Multi-Chassis Trunk (MCT) is a trunk that initiates at a single MCT-unaware server or switch and terminates at two MCT-aware switches.

Link Aggregation (LAG) trunks provide link level redundancy and increased capacity. However, LAG trunks do not provide switch-level redundancy. If the switch to which the LAG trunk is attached fails, the entire LAG trunk loses network connectivity. With MCT, member links of the LAG are connected to two chassis. The MCT switches may be directly connected using an Inter-Chassis Link (ICL) to enable data flow and control messages between them. In this model, if one MCT switch fails, a data path will remain through the other switch.

In an MCT scenario, all links are active and can be load shared to increase bandwidth. In addition, traffic restoration can be achieved in milliseconds after an MCT link failure or MCT switch failure.

MCT is designed to increase network resilience and performance.

FIGURE 136 Chassis trunk example



MCT Benefits

MCT provides the following benefits:

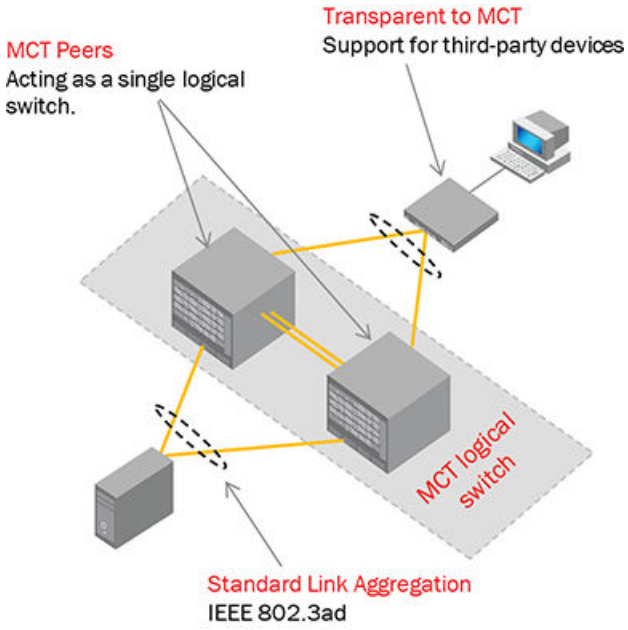
- Provides link-level and switch-level redundancy.
- Provides increased capacity because it utilizes all links (including redundant ones) for traffic transport. This contrasts with the use of the Spanning Tree Protocol, which does not use redundant links for transporting traffic.
- Provides traffic restoration in tens of milliseconds in case of link or switch failures.
- Allows servers and switches to have redundant connections to two switches and to fully utilize all links (including redundant ones) for traffic transport.
- Allows servers and switches to use standard link aggregation (802.3ad) to connect to redundant switches.
- MCT is easily deployed while enhancing existing multilayer switching without fundamentally changing the architecture.

How MCT works

The MCT is made up of the following:

- Sub-second failover in the event of a link, module, switch fabric, control plane, or node failure
- Layer 2 and Layer 3 forwarding (when using fast path forwarding) at the first hop regardless of VRRP-E state.
- Flow based load balancing rather than VLANs sharing across network links
- Ability to provide the resiliency regardless of the traffic type layer 3, layer 2 or non-IP (legacy protocols).
- Interaction with MRP to build larger resilient Layer 2 domains
- Enhancement to Link Aggregation Groups
- Provides nodal redundancy in addition to link and modular redundancy
- Operates at the physical level to provide sub-second failover

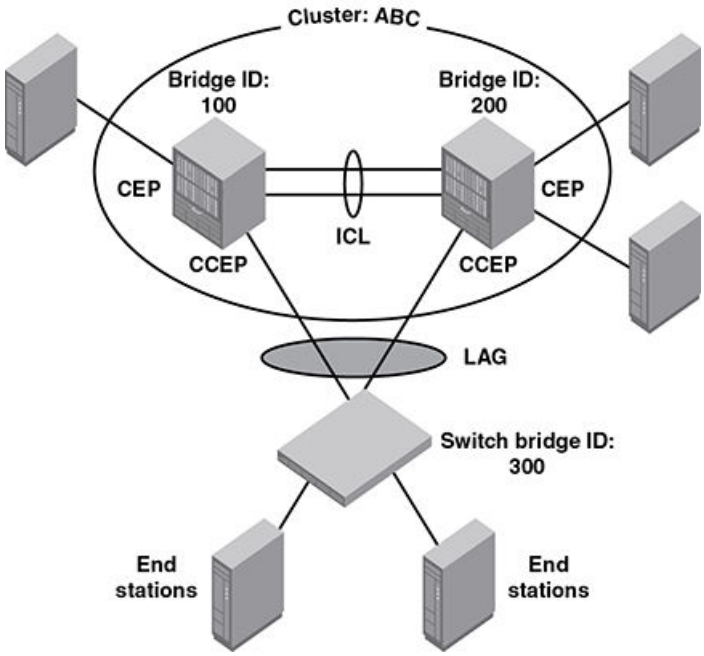
FIGURE 137 How MCT works



MCT components

To properly understand MCT, consider Figure 140, which shows an example of MCT deployment, functions and features.

FIGURE 138 MCT Components



MCT terminology

- MCT peer switches: A pair of switches connected as peers through the ICL. The LAG interface is spread across two MCT peer switches and acts as the single logical endpoint to the MCT client.
- MCT client: The MCT client is the device that connects with MCT peer switches through an IEEE 802.3ad link. 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 Inter-Chassis Link (ICL): A single-port or 1 GbE or 10 GbE interface between the two MCT peer switches. This link is typically a standard IEEE 802.3ad Link Aggregation interface. ICL ports should not be untagged members of any VLAN. The ICL is a tagged Layer 2 link, which carries packets for multiple VLANs. MCT VLANs are the VLANs on which MCT clients are operating. On the XMR Series or MLX Series devices, non-MCT VLANs can coexist with MCT VLANs on the ICL. However, on the CER 2000 Series and CES 2000 Series devices, only MCT VLANs are carried over the ICL. NetIron OS devices support up to 4096 ICL VLANs for a cluster.
- MCT Cluster Communication Protocol (CCP): The Extreme proprietary protocol that provides reliable, point-to-point transport to synchronize information between peers. CCP comprises two main components: CCP peer management and CCP client management. CCP peer management deals with establishing and maintaining a TCP transport session between peers, while CCP client management provides event-based, reliable packet transport to CCP peers.
- MCT Cluster Client Edge Port (CCEP): A physical port on one of the MCT peer switches that is a member of the LAG interface to the MCT client. To have a running MCT instance, at least one Link Aggregation Interface is needed with a member port on each peer switch.
- MCT Cluster Edge Port (CEP): A port on MCT peer switches that is neither a Cluster Client Edge Port nor an ICL port.
- MCT VLANs: VLANs on which MCT clients are operating. These VLANs are explicitly configured in the MCT configuration by the user.

  NOTE
  For MCT VLANs, MAC learning is disabled on the ICL ports, while MAC learning is enabled on ICL port for non-MCT VLANs.

- MCT session VLANs: The VLAN used by the cluster for control operations. CCP runs over this VLAN. The interface can be a single link or a LAG port. If it is a LAG port, it should be the primary port of the LAG.

  NOTE
  The MCT session VLAN's subnet will not be distributed in routing protocols using redistribute commands.

- RBridge ID: RBridge ID is a value assigned to MCT nodes and clients to uniquely identify them, and helps in associating the source MAC address with an MCT node.
- MAC Database Update Protocol (MDUP)
- CL: Cluster Local MACs
- CCL: Cluster Client Local MACs
- CR: Cluster Remote MACs
- CCR: Cluster Client Remote MACs
- CCRR: Cluster Client RBridge Reachability
- MDB: MAC Database. The MDB can have multiple MAC entries for the same address.
- FDB: Forwarding MAC Database. The FDB will have the best MAC address only installed.

NOTE
BFD sessions configured with lower timer values may exhibit flaps when configured alongside MACSec on same line card. This issue is a known limitation. However, the MCT sessions are stable with 1 second for 3 tries in such scenarios.

Dynamic LAGs

MCT Client creates a single dynamic LAG towards the MCT nodes. For MCT nodes the dynamic LAG consists of two LAGs, each is configured on one of the MCT devices. A dynamic LAG runs Link Aggregation Control Protocol (LACP).

For the two dynamic LAGs of the MCT to behave as a single LAG from the MCT client's perspective, both of the dynamic LAGs should have the same LACP system ID and key, referred to as the MCT system ID and MCT key. In a system configuration with multiple MCT peers, the LACP system priority on both the MCT nodes should be same.

The MCT system ID and MCT key is uniquely defined for one MCT. They have the following attributes:

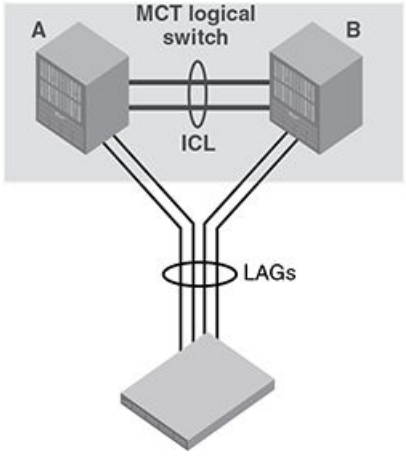
- MCT base system id = 0180.c200.0000
- MCT system id = MCT base system ID + cluster ID
- The cluster ID is user configurable on each MCT peer and unique across the MCT system
- MCT base key = 30000
- MCT LAG Group ID = MCT base key + client bridge ID

NOTE
Each MCT node has a unique cluster ID and an MCT client ID.

MCT peers

Each MCT physical node, A and B, will act as an MCT peer and they are connected using an ICL. The pair of MCT peers will act as one logical switch for the access switch or server so that the MCT pair can connect using standard LAG to them. This is illustrated in MCT peers.

FIGURE 139 MCT peers



ICL traffic handling

An ICL link on the device can be a single port or a static or LACP LAG. Non-MCT VLANs can co-exist with MCT VLANs on the ICL only on the MLX Series and XMR Series devices .

> NOTE
> In cases where routed traffic needs to flow between non-MCT VLANs and MCT member VLANs, add these non-MCT VLANs as a member-VLANs under the MCT cluster configuration. Failing to perform this step might cause one arm routing (OAR) on the ICL link and a high CPU condition.

For MCT VLANs, MAC learning is disabled on ICL ports.

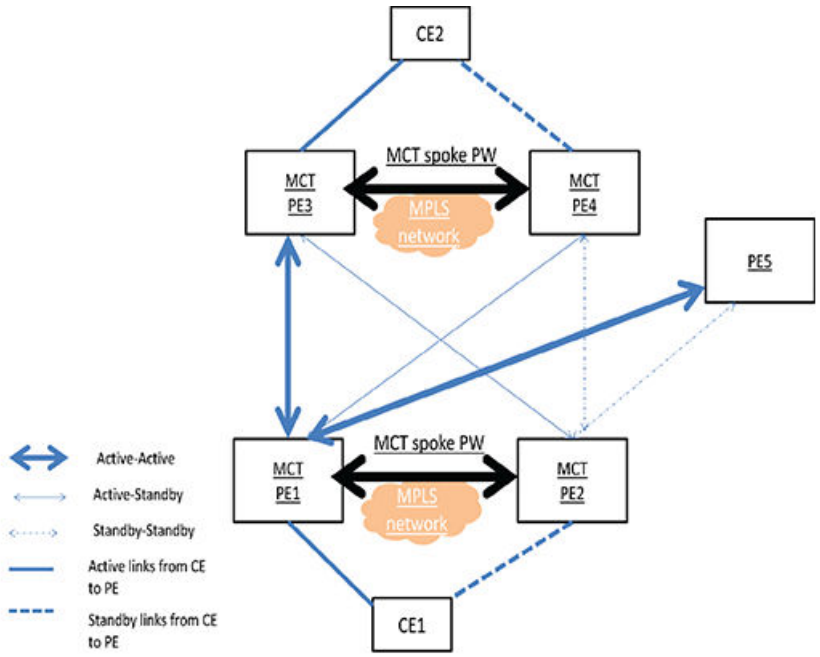
MCT Active-Passive mode

Using the MCT Active-Active mode, both MCT nodes will be Active forwarding the traffic coming from the client nodes on the CCEP links.

Using the MCT Active-Passive mode, the CCEP ports on one of the MCT nodes will be made Passive by blocking, and the other MCT node will be Active.

To enable MCT Active-Passive mode to work, both MCT nodes must be configured with the ACTIVE-PASSIVE feature. If only one node is configured, then cluster CCP session will not come up.

FIGURE 140 MCT Active-Passive topology



MCT Active-Passive mode is an example of an MCT Active-Passive topology. PE1 and PE2 are nodes of cluster, MCT1. PE3 and PE4 are nodes of another MCT cluster, MCT2.

- CE1 and CE2 are clients of MCT1 and MCT2 respectively.
- Data traffic will be forwarded on the Active-Active links between the PEs.
- Data traffic will be forwarded on the Active links from the CE to PE.

Multicast snooping over MCT

To support multicast snooping over MCT for NetIron OS devices, the ICL port is used to synchronize the following information between the cluster switches using MDUP:

- MAC - forward entries (mcache entries on MCT VLAN).
- IGMP or MLD Join or Leave (control packets on MCT VLAN).
- PIM-SM Join or Prune (control packets on MCT VLAN).

IGMP or MLD snooping

Each cluster switch in the MCT VLAN can be configured either as active or passive. There is no restriction for cluster switches to run Active-Active or Passive-Passive configuration.

Implementation of MCT is exactly the same for both IGMP or MLD Snooping, so all further details are explained only with reference to IGMP.

IGMP and MLD Control Packet Processing on MCT cluster switches

For IGMP reports and leaves,

- Native Packets coming to CPU from CCEP endpoints will be encapsulated in MDUP header and sent across ICL link through the TCP connection to the remote cluster peer, along with required control information in the header indicating the packet was received from a CCEP link, VLAN and client RBridgeID.
- Native Packets coming to CPU from CEP endpoint will be encapsulated in MDUP header and sent across ICL link through the TCP connection to the remote cluster peer, along with required control information in the header indicating the packet was received from a CEP link, VLAN and peer RBridgeID.
- Native Packets coming to CPU from ICL will not be processed to learn the group member. Only the MDUP messages that are sent through the TCP connection established between ICL links will be used to build the group OIF list.
- Packets coming from CEP and CCEP will be forwarded to router ports that are CEP, CCEP and ICL.
- Packets coming from ICL will be forwarded to router ports that are CEP. Packets will not be forwarded to router ports that are CCEP unless the peer CCEP ports are down.

For IGMP queries,

- Queries coming from CEP and CCEP will be forwarded to non-querier port that is CEP, CCEP and ICL.
- Queries coming from ICL will be forwarded to non-querier port that is CEP. Packet will not be forwarded to non-querier port that is CCEP unless the peer CCEP port is down.

Forwarding entries for multicast snooping

TABLE 43 Forwarding entries (,G)

Event	MCT-1	MCT-2
No-Join	(*,G)->blackhole	(*,G)->blackhole
(S,G)Join on (MCT-1)CEP	(*,G)->CEP - sources maintained on a port hash-list.	(*,G)->ICL
(S,G)Join on (MCT-2)CEP	(*,G)->ICL	(*,G)->CEP
(S,G)Join on (MCT-1)CCEP	(*,G)->CCEP , ICL]	(*,G)->CCEP, ICL
(S,G)Join on (MCT-2)CCEP	(*,G)->CCEP, ICL	(*,G)->CCEP, ICL

TABLE 44 Forwarding entries (S,G)

Event	MCT-1	MCT-2
No-Join	(S,G)->blackhole	(S,G)->blackhole
Join (MCT-1)CEP	(S,G)->CEP	(S,G)->ICL
Join on (MCT-2)CEP	(S,G)->ICL (S,G)->CEP	(S,G)->CEP
Join (MCT-1)CCEP	(S,G)->CCEP, ICL	(S,G)->CCEP, ICL
Join (MCT-2)CCEP	(S,G)->CCEP, ICL	(S,G)->CCEP, ICL

L2 protocol packet handling

The default behavior is to forward or flood STP and RSTP BPDU packets.

If the **no cluster-l2protocol-forward** command is configured on global basis or **cluster-l2protocol-forward** disable is configured on a port, the STP protocol packets coming on the ICL ports of MCT VLANs are dropped.

All other L2 protocol packets will be flooded on the MCT VLANs or dropped. The **cluster-l2protocol-forward** command is not applicable to these protocol packets. It only applies to STP or RSTP BPDU packets on the ICL ports only.

Forwarding broadcast, multicast and unknown unicast traffic

Traffic received from non-ICL ports is forwarded the same way as non-MCT devices.Traffic received from an ICL port is not forwarded to the CCEP port if the peer MCT switch has the reachability to the same cluster client.

Both the MCT nodes must be multicast enabled for multicast routing on MCT.

> NOTE
> When there is a double failure, the forwarding behavior will be unpredictable and there may be a complete traffic loss. For example, when both the ICL cluster link and any one leg of the client CCEP link fail. From the physical topology perspective, it may appear like a path is available, while traffic may not be forwarded.

CES 2000 Series and CER 2000 Series forwarding

The ICL port must belong to VLANs that are cluster member VLANs.

Syncing interface MACs to peer MCT devices

The MCT device uses an interface MAC to identify the packets that are addressed to the switch. Such packets may be received by a peer MCT device. The peer MCT device switches packets over the ICL to the local MCT switch to be routed properly.

MCT L2 protocols

When configuring L2 protocols, you should consider the following.

MRP

- An ICL interface can not be configured as MRP secondary interface and vice-versa as the ICL cannot be BLOCKING.
- MRP can not be enabled on MCT CCEP port and vice-versa.

G.8032

- If the port is an ERP interface, it can not be enabled as a CCEP port.
- If the interface is ICL interface it can not be configured with MS, FS or RPL.
- G.8032 and MCT are not supported together.

STP

- The STP algorithm has been modified such that ICL never goes to blocking. ICL guard mechanism ensures that if ICL is going in blocking state then the port on which the superior BPDUs are being received is moved to BLOCKING state and ICL guard timer starts running on it. This timer runs as long as Superior BPDUs are received on this interface. As long as this timer runs on an interface the Superior BPDUs are dropped.
- The modified STP algorithm also ensures that the CCEP interface state on both the MCT peers is same.
- The CCEP STP state information between MCT peers is synchronized using messages that are sent over CCP.
- Only one of the MCT peers send BPDUs towards the MCT Client. It is decided by whosoever is the designated bridge on the ICL.
- New STP States:
  - BLK_BY_ICL state indicates that the superior BPDUs were being received on this interface which could have led to BLOCKING of ICL interface, due to which ICL port guard mechanism has been triggered on this port.
  - FWD_BY_MCT state indicates that the MCT peer has set the CCEP state to forwarding.
  - BLK_BY_MCT state indicates that the MCT peer has set the CCEP state to blocking.

MCT L3 protocols

When configuring dynamic L3 protocol support over MCT, you should consider the following.

- L3 Protocols IS-ISv4, IS-ISv6, BGP4, BGPv6, OSPFv2, OSPFv3, RIP, and RIPng are supported.
- CCEP and CEP are on different L3 networks.
- Any of the L3 protocol can be configured on the CCEP ports and CEP ports.
- An ICL interface can not be configured with L3 routing protocols.
- A CEP Network is reachable via MCT nodes even when the CCEP is down on one of the peers , the traffic will take redundant path via MCT.
- Active and Passive L3 routing protocols interfaces are supported.

MCT feature interaction

Use the following feature matrix when configuring MCT:

TABLE 45 MCT feature interaction matrix

Supported	Not Supported
LACP on Inter-Chassis Link (ICL) and Customer Client Edge Ports (CCEP).	MSTP, VSRP, and PIM.
VRRP on CCEP.	**ESI VLANs** on CCEP or ICL ports.
MRP and **MRP II** supported with the restriction that ICL port cannot be the secondary port of the MRP ring.	**GRE** is not supported on the ICL VE interfaces.
BGP, **IS-IS**, and **OSPF** on CCEP.	**BFD** on CCEP.
802.1ad on CCEP or ICL ports.	**DAI** on CCEP.

TABLE 45 MCT feature interaction matrix (continued)

Supported	Not Supported
Flooding features (VLAN CPU protection, multicast flooding, 802.1ad and others) on cluster VLANs.	**802.1ah** on CCEP or ICL ports.
Multi-VRF	**MSTP**
UDLD as independent boxes.	**VPLS** on ICL ports.
VPLS and **VLL** on CCEP.	**VLL** on ICL ports.
Link OAM as independent boxes.	**MPLS** on CCEP or ICL ports.
802.1ag as independent boxes.	**Hitless Upgrade** is not supported, on the MLX Series and XMR Series devices, however it is compatible. If the operation is performed with cluster configuration the TCP session is reestablished. The MAC addresses from the cluster peers will be re-validated and programmed accordingly. Extreme recommends shutting down all the CCEP on the cluster node so that there is graceful failover and then hitless operation can be performed.
ARP as independent boxes.	**Hitless Failover** is not supported on the MLX Series and XMR Series devices, however it is compatible. If the operation is performed with cluster configuration the TCP session is reestablished. The MACs from the cluster peers will be re-validated and programmed accordingly. Extreme recommends shutting down all the CCEP on the cluster node so that there is graceful failover and then hitless operation can be performed.
STP and RSTP	**Multi-port ARP** will not be allowed on CCEP or ICL ports on the MLX Series and XMR Series devices.
Port MAC Security on the node where it is programmed.	**Multiport MAC** is not supported on CCEP or ICL ports. Configuration will be rejected when trying to configure multiport MAC addresses with a port mask which contains either a CCEP or ICL port and vice-versa on the MLX Series and XMR Series devices.
802.1x on the node where it is programmed.	
Static MAC configuration - Static MACs are programmed on both local and remote peers as static entries.	

Active-Active MCT configuration considerations

- On Customer Client Edge Ports (CCEP), MCT does not support 802.1ah (listed in MCT feature interaction on page 417 as unsupported).
- ICL ports should be not be an untagged member of any VLAN. An ICL is preferably a LAG that provides port level redundancy and higher bandwidth for cluster communication.
- On the MLX Series and XMR Series devices, ICL ports can be part of MCT VLANs as well as regular VLANs.
- The MLX Series and XMR Series devices will disable MAC learning on ICL ports for the VLANs configured in the cluster. However, MAC learning is enabled on ICL port for non-cluster VLANs.
- In cases where routed traffic needs to flow between non-MCT VLANs and MCT member VLANs, add these non-MCT VLANs as a member-VLANs under the MCT cluster configuration. Failing to perform this step might cause one arm routing (OAR) on the ICL link and a high CPU condition.
- MAC Database Update Protocol (MUDP) will synchronize all MAC entries for VLANs served by ICL link.
- Cluster ID should be same on both cluster switches.
- Cluster RBridge ID should not conflict with any client RBridge ID or the peer RBridge ID.
- Client RBridge ID is unique and it should be same on the cluster switches.

- You can add any ports to the session VLAN (For the purpose of adding a port to a LAG), but Extreme recommends keeping only ICL ports as tagged members for the session VLAN during operation.
- MCT clients may support 16 members per LAG.
- An ICL interface cannot be configured as the CCEP port in any client.
- CCEP ports on MCT node can be single port or LAG.
- If ICL or client interfaces need to be configured as LAG interface then only the primary port of the LAG needs to be specified in the ICL or client configuration.
- Once the cluster is deployed, only the cluster member VLANs and client isolation mode can be modified. Other configurations are not allowed to change.
- Once the client is deployed, any configuration under client is not allowed to change.
- Clients can be added or deleted even if the cluster is deployed.
- When the cluster is not deployed, then all the clients in the cluster become inactive.
- As soon as a port is configured as an ICL port it is removed from default VLAN.
- If an ICL or CCEP is a LAG interface, the LAG has to be configured separately on each node.

Configuring Active-Active MCT

The following basic steps are required to build an MCT scenario. Refer to Single level MCT example on page 422 for detailed instructions.

1. Create and deploy a LAG that will be used as ICL port.
2. Create and deploy the LAGs facing the clients.
3. Enable Layer 2 switching (**no route-only** command) either globally or on specific interfaces.
4. Create and add ports to a client or member VLAN that they will be using for communication.
   a) At the CCEP or CCP, these ports may be tagged or untagged into the VLAN.
   b) At the ICL, these ports may only be tagged into the VLAN. The ICL ports cannot be untagged in any VLAN.
5. Create a dedicated session VLAN for the ICL interfaces for CCP communication. ICL ports will be tagged into the session VLAN. It is recommended to use a high VLAN number that will not be touched by data VLANs.
6. Create a virtual routing interface and associate this with the session VLAN. This will be used to address the link between the MCT peers.
7. Configure the MCT cluster.
   a) Create a cluster with any name but with a cluster ID matching the MCT peer.
   b) Configure a unique RBridge ID for this peer. The RBridge ID must be unique across all MCT peers and CCEPs.
   c) Configure the session VLAN for the cluster.
   d) Configure one or more client or member VLANs for the cluster.
   e) Configure the ICL port or LAG being used for the cluster.
   f) Configure the MCT peer for this cluster.
   g) Configure the time delay for the LACP blocked state to enable the MCT nodes to process the remote CCEP events. This configuration step is supported on MLX Series and XMR Series devices only.
8. Deploy the cluster. All attributes except for client or member VLANs cannot be changed after the cluster is deployed.
9. Configure the MCT cluster client instances.

Active-Passive MCT

Active-Passive MCT configuration considerations

- To enable MCT Active-Passive mode, both the MCT nodes must be configured as Active-Passive. If only one node is configured, then cluster CCP session will not come up.
- Active-Passive mode is supported for the following combinations.
  - L2VPN MCT configured.
  - L2 + L2VPN MCT configured.
- Active-Passive mode does not support L2 MCT.
- VLL and VPLS are supported over L2VPN MCT.
- Node where the Pseudo Wire's are Active must be configured as the MCT Active node .
- The CCEP ports must be part of dynamic LAG. Active-Passive mode is not supported on static LAG CCEP ports and single port CCEP.
- CCEP ports elected under passive node will be made LACP-BLOCKED.
- VLL PW's role is based on the client role, the node under which the client ports are active the PW's will be active from that particular node towards the upstream.
- When using VPLS, the Pseudo Wire's role and the client role are independent, and is based on the client role election based on the configuration.

Configuring Active-Passive MCT

After configuring the basic steps required to build an MCT scenario, complete the following steps to configure the MCT cluster.

1. Create a cluster with any name but with a cluster ID matching the MCT peer.
2. Configure a unique RBridge ID for this peer. The RBridge ID must be unique across all MCT peers and CCEPs.
3. Configure the session VLAN for the cluster.
4. Configure one or more client or member VLANs for the cluster.
5. Configure the LAG being used for the cluster.
6. Enable the Active or Passive mode.
7. Configure the client role.
8. Configure the client role revertible mode.
9. Configure the client role revertible timer.
10. Configure the MCT peer for this cluster.
11. Configure the ICL port or LAG being used for the cluster.
12. Configure the MCT peer for this cluster.
13. Configure the time delay for the LACP blocked state to enable the MCT nodes to process the remote CCEP events. This configuration step is supported on MLX Series and XMR Series devices devices only.
14. Deploy the cluster. All attributes except for client or member VLANs cannot be changed after the cluster is deployed.

15. Configure the MCT cluster client instances.

    After deploying the cluster, both cluster peers will begin exchanging their cluster mode and client role information. The selection will take place based on the criteria listed in Table 46.

TABLE 46 Client role election criteria

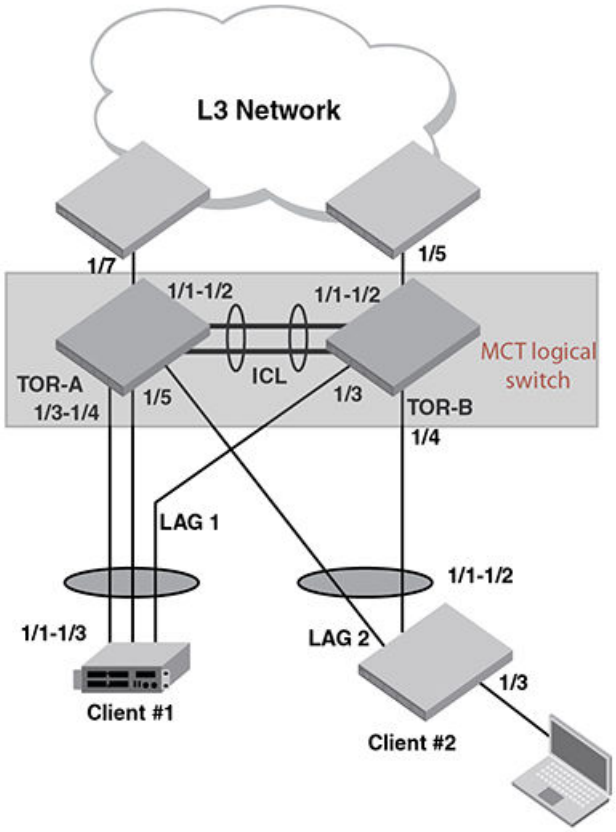
PE1 client role configuration	PE2 client role configuration	Active links of MCT
Active	Active	Based on the RBridge ID
Passive	Active	PE2
N/A	Active	PE2
Active	Passive	PE1
Passive	Passive	Based on the RBridge ID
N/A	Passive	PE1
Active	N/A	PE1
Passive	N/A	PE2
N/A	N/A	Based on the RBridge ID

Sample Active-Passive MCT cluster configurations

```
Cluster MCT1 1
  rbridge-id 100
  mode-active-passive
  client-role active
  client-role-revertible-delay timer 5
  l2vpn-peer 12.12.12.12 rbridge-id 101
  deploy
 client c1
  rbridge-id 100
  client-interface ethernet 1/6
  deploy
Cluster MCT2 1
  rbridge-id 200
  mode-active-passive
  client-role-revertible-delay timer 5
  l2vpn-peer 11.11.11.11 rbridge-id 101
  client-interfaces sync_ccep_early lacp-delay 5
        deploy
 client c1
  client-role passive
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
```

Single level MCT example

FIGURE 141 Single level MCT



The following steps are task for configuring a MCT scenario as shown in **Figure 176** .

> **NOTE**
> Save the current configuration files for both chassis operating in standalone mode before you begin creating a MCT.

TOR-A (Top of rack MCT capable switch)

See **Figure 176 .**

Creating LAG-1

1. You can either assign a LAG **ID** explicitly or it will be automatically generated by the system. The LAG **ID** stays the same across system reload and hitless upgrade.

   The command to configure LAGs allows explicit configuration of the LAG ID for static and dynamic LAGs.

   To create a LAG with the LAG ID option, enter a command such as the following.

   ```
   device(config)# lag 1 dynamic
   device(config-lag-1)#
   ```

   Syntax: [no] lag name [**static** | **dynamic**] [**id number**]

   The **ID** parameter is optional. The value of the **ID** parameter that you can enter is from 1 to 256. If you do not enter a LAG **ID** , the system will generate one automatically. Once the LAG **ID** is generated the system will save it in the configuration file along with the LAG name, therefore the value will stay the same across system reload.

   > NOTE
   > The LAG ID parameter is for static and dynamic LAGs only. No explicit configuration of a LAG id is allowed on keepalive LAGs.

   The **static** parameter specifies that the LAG with the name specified by the *lag-name* variable will be configured as a static LAG.

   The **dynamic** option specifies that the LAG with the name specified by the *lag-name* variable will be configured as a dynamic LAG.

2. Define the ports the LAG will be using as shown in the following.

   ```
   device(config-lag-1)# ports ethernet 1/1 to 1/2
   ```

   Syntax: [no] ports ethernet [**slot/port**] | **to** | [**slot/port**]

   Use the appropriate *slot/port* variable to specify a Ethernet port within the LAG that you want to enable.

3. The primary port must be explicitly assigned using the **primary-port** command. To designate the primary port for the static LAG "1", use the following command.

   ```
   device(config-lag-1)# primary-port 1/1
   ```

   Syntax: [no] primary-port slot/port

   Once a primary port has been configured for a LAG, all configurations that apply to the primary port are applied to the other ports in the LAG.

   > NOTE
   > This configuration is only applicable for configuration of a static or dynamic LAGs

4. After configuring a LAG, you must explicitly enable it before it begins aggregating traffic. This task is accomplished by executing the deploy command within the LAG configuration. After the deploy command runs, the LAG is in the aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG. The running configuration will no longer display deployed LAG ports other than the primary port.

   To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keep-alive LAGs. After a non keep-alive LAG is deployed, a trunk is formed. If there is only one port in the LAG, a single port is formed. For a dynamic LAG, LACP is stared for each LAG port.

   Use a command such as the following to deploy LAG 1

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

   Syntax: [no] deploy [**forced** | **passive**]

   When the **deploy** command is executed:

   - For a static and dynamic LAGs, the current veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a **no** is formed with all the ports in the LAG.
   - For dynamic LAGs, LACP is activated on all LAG ports. When activating LACP, use active mode if passive is not specified; otherwise, use passive mode.
   - For a keep-alive LAGs, a LAG is formed, and LACP is started on the LAG port.

   Once the deploy command is issued, all LAG ports will behave like a single port.

   If the **no deploy** command is executed, then the LAG is removed. For dynamic LAGs, LACP is de-activated on all of the LAG ports.

   If the **no deploy** command is issued and more than 1 LAG port is not disabled the command is aborted and the following error message is displayed: "Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted." Using the **forced** keyword with the **no deploy** command in the previous situation, the un-deployment of the LAG is executed.

5. Assign a name to an individual port within a LAG using the **port-name** command within the LAG configuration as shown in the following. Using the **port-name** command is optional.

   ```
   device(config-lag-1)# port-name ICL-to-TOR-B:1/1 ethernet 1/1
   device(config-lag-1)# port-name ICL-to-TOR-B:1/2 ethernet 1/2
   ```

   Syntax: [no] port-name text ethernet [**slot/port**] | **pos** [**slot/port**]

   The *text* variable specifies the port name. The name can be up to 50 characters long.

   Use the **ethernet** option with the appropriate *slot/port* variable to apply the specified name to an Ethernet port within the LAG.

   Use the **pos** option with the appropriate *slot/port* variable to apply the specified name to a Packet-over-SONET port within the LAG.

Creating LAG 2

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create LAG 2 as shown below.

   ```
   device(config)# lag 2 dynamic id 2
   device(config-lag-2)#
   ```

2. Define the ports the LAG will be using.

   ```
   device(config-lag-2)# ports ethernet 1/3  to 1/4
   ```

3. Deploy the LAG 2 as shown below.

```
device(config-lag-2)# deploy
```

4. Assign a name to an individual port within a LAG.

```
device(config-lag-2)# port-name lag-client-1:1/1 ethernet 1/3
device(config-lag-2)# port-name lag-client-1:1/2 ethernet 1/4
```

Creating LAG 3

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create LAG 3 as shown below.

```
device(config)# lag 3 dynamic id 3
device(config-lag-3)#
```

2. Define the ports the LAG will be using.

```
device(config-lag-3)# ports ethernet 1/5
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-3)# primary-port 1/5
```

4. Deploy the LAG 3 as shown below.

```
device(config-lag-3)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-2)# port-name lag-client-2:1/1 ethernet 1/5
device(config-lag-2)# port-name lag-client-1:1/2 ethernet 1/4
```

Enable layer 2 switching

By default, Extreme devices support routing over layer 2 switching. You can enable layer 2 switching globally or on individual port using the **no route-only** command. The **no route-only** and **route-only** commands prompts you for whether or not you want to change the "route-only" behavior. You must enter **y** if you want to proceed or **n** if you do not. To enable Layer 2 switching only on a specific interface, go to the interface configuration level for that interface, and add the **no route-only** command.

NOTE
On the XMR Series and MLX Series routers, **route-only** is the default condition. Because **route-only** is the default condition, it will not be displayed in the configuration. If you use **no route-only** to enable switching, the **no route-only** command will be displayed in the configuration.

NOTE
On the CES 2000 Series and CER 2000 Series devices, **route-only** is disabled by default. Therefore, if **route-only** is enabled on a CES 2000 Series or CER 2000 Series device, it will be displayed in the configuration.

Use commands such as the following to enable Layer 2 switching.

```
device(config)# no route-only
```

Syntax: [no] route-only

Creating VLANs for client traffic

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating VLANs.

Creating VLANs for client traffic

1. At the global CONFIG level assign an ID to the VLAN.

```
device(config)# vlan 2
```

2. Add the client ports to the VLAN as either tagged or untagged. In this example the client ports are untagged.

```
device(config-vlan-2)# untag eth 1/3 to 1/5
```

3. Add the ICL port or LAG to the VLAN as tagged. ICL ports cannot be untagged in any VLAN and will automatically be removed from the default VLAN upon MCT cluster configuration.

```
device(config-vlan-2)#tag eth 1/1 to 1/2
```

Create the session VLAN

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating VLANs.

1. At the global CONFIG level assign an ID to the VLAN .

```
device(config)# vlan 4090 name Session-VLAN
```

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-4090)# tagged ether 1/1 to 1/2
```

3. Configure a virtual routing interface on each IP protocol VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

```
device(config-vlan-4090)# tagged ether 1/1 to 1/2
device(
config-vlan-4090)# router-interface ve 100
```

Assign the hostname (optional)

To configure a system name, enter commands such as the following.

```
device(config)# hostname TOR-A
```

Enabling interfaces

The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the ports on a device for ethernet interfaces 1/1, 1/3, and 1/5, enter the following commands.

```
device(config)# interface ether 1/1
device(config-if-e10000-1/1)# enable
device(config)# interface ether 1/3
device(config-if-e10000-1/3)# enable
device(config)# interface ether 1/5
device(config-if-e10000-1/5)# enable
```

Syntax: [no] enable

Assigning a port name (optional)

A port name can be assigned to help identify interfaces on the network. You can assign a port name to physical ports, virtual routing interfaces, and loopback interfaces. To assign a name to a ports 1/6 and 1/7, enter the following commands.

```
device(config)# interface ethe 1/6
device(config-if-e10000-1/6)# port-name CEP-PC
device(config-if-e10000-1/6)# enable
device(config)# interface ethe 1/7
device(config-if-e10000-1/7)# port-name to-L3-ECMP
device(config-if-e10000-1/7)# enable
```

Syntax: [no] port-name *text*

Syntax: [no] enable

The *text* parameter is an alphanumeric string. The name can have up to 255 characters on a device and can include blanks. You do not need to use quotation marks around the string, even when it contains blanks.

Adding a virtual interface

Add a virtual interface and configure an IP address on the interface by entering commands such as the following.

```
device(config)# interface ve 100
device(config-vif-100)# ip address 1.1.1.1/24
```

Syntax: [no] interface [**ve** *ve-id*]

Syntax: [no] ip address *ip-addrip-mask*

The *ve-id* variable allows you to specify a VE interface ID.

Configuring the cluster operation mode

See **Figure 176** and Creating LAG-1 on page 423.

1. To configure a device with cluster ID 1, enter a command such as the following.

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

   Syntax: [no] cluster *cluster-namecluster-id*

   The *cluster-name* parameters specify the cluster name with a limit of 64 characters.

   The *cluster-id* parameters specify the cluster ID (1-65535).

2. Configure the local RBridge ID for the cluster. This RBridge ID is used by the peer to communicate with this cluster node and to define CCEPs. The RBridge ID needs to be unique across the cluster and unique between MCT peer Bridge IDs as well as cluster client instances. To configure the local rbridge, enter a command such as following

   ```
   device(config-cluster-TOR)#rbridge-id 1
   ```

   Syntax: [no] rbridge-id *id*

   The *id* parameter specifies the remote bridge ID. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN can be in the range 1-4090, but it cannot be the default VLAN. A check is made during the cluster deploy in addition to a dynamic check. The default VLAN cannot be changed to a VLAN which is already defined as cluster session. Note: ICL ports must be tagged within the session VLAN. Enter a command such as the following to create the session VLAN.

```
device(config-cluster-TOR)# session-vlan 4090
```

Syntax: [no] session-vlan *vlan-id*

The *vlan-id* parameter specifies the VLAN range. Possible values are 1 - 4090.

4. Specify the VLAN range on which cluster is operating. This would be the range for which there would be MAC synchronization. Multiple VLAN ranges would be supported for the configuration. Enter a command such as the following to create the member VLAN.

```
device(config-cluster-TOR)# member-vlan
 2
```

Syntax: [no] member-vlan *x* [**to** *y*]

NOTE
The VLAN range is allowed to change even if cluster is deployed.

5. Specify the ICL for the cluster. The ICL interface can be a single link or trunk port. If it is a trunk port, it should be the primary port of the trunk . Only one ICL is supported. Enter a command such as the following to create the ICL for the cluster.

```
device(config-cluster-TOR)#icl
 TOR ethernet 1/1
```

Syntax: [no] icl *icl-name* **ethernet** *x/y*

The *icl-name* parameter can be up to 64 characters in length.

The **ethernet** *x/y* parameter is the ICL interface.

6. Specify the rbridge and ICL for the peers by entering a command such as the following.

```
device(config-cluster-TOR)# peer 1.1.1.2 rbridge-id 2 icl TOR
```

Syntax: [no] peer *peer-ip* **rbridge-id** *peer-rbridge* **icl** *map-icl*

The *peer-ip* parameter should be in same subnet as that of cluster management interface.

The *peer-rbridge* parameter should be different from cluster rbridge and any other client in the cluster

The *map-icl* parameter is the ICL name to reach this cluster peer.

7. The cluster can be deployed separately without any clients configured. The **deploy** command brings the cluster into effect. The following can be changed when the cluster is deployed:
   - – Client isolation mode
   - – Member VLANs
   - – Clients added and removed.

8. The **deploy** command also preforms a consistency check of the entire cluster configuration. If anything is amiss, an error message is sent. The following specific information is checked during deployment:
   - If the cluster management VLAN is configured
   - If the cluster peer is configured
   - If the cluster ICL is configured

   Enter a command such as the following to deploy the cluster configuration.

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

   Syntax: [no] deploy

Creating cluster client 1

See **Figure 176** .

1. Create a cluster client instance and change the mode to the client instance. If an instance is already present, then directly change the mode to the client instance mode.

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

   Syntax: [no] client *client-name*

   The *client-name* parameter can be 64 characters (maximum).

   If it is a two port MCT, the maximum clients supported on the XMR Series or MLX Series device is 1536/2.

   If it is a two port MCT, the maximum clients supported on the CES 2000 Series or CER 2000 Series device is 1535/2.
2. Configure the local RBridge ID for the cluster. This RBridge ID is used by the peer to communicate with this cluster node. To configure the local rbridge, enter a command such as following

   ```
   device(config-cluster-TOR-client-1)#rbridge-id 100
   ```

   Syntax: [no] rbridge-id *id*

   The *id* parameter specifies the local bridge id. Possible values are 1 - 35535 (16 bit value).
3. The cluster session VLAN is the VLAN used by the cluster for control operations. Add the (CCEP or CEP) interfaces to the cluster client instance. The interface can be a single link or LAG port. If it is LAG port, it should be the primary port of the LAG.

   ```
   device(config-cluster-TOR-client-1)#client-interface ether 1/3
   ```

   Syntax: [no] client-interface *interface* **interface : ethernet** *x/y*

   The **ethernet***x/y* parameter is the ethernet interface.
4. Deploy the cluster client. If cluster is not deployed, the configuration will be taken but the client state machine will not be started. The consistency checks for client will be done at the time of client deploy. The following configuration checks will be preformed:
   - – Client interface is configured
     – Client interface is not same as any other client interface or ICL interface
     – Client RBridge ID is not same as cluster rbridge or any peer rbridge

   Once the client is deployed, the configuration inside the client will not be allowed to change, To deploy the client configuration, enter a command such as the following.

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

   Syntax: [no] deploy

Create cluster client 2

See **Figure 176** and Create cluster client 1 on page 436 for additional information for creating cluster clients.

1. Create a cluster client instance.

```
device(config-cluster-TOR)# client client-2
```

2. Configure the client RBridge ID.

```
device(config-cluster-TOR-client-2)#rbridge-id 200
```

3. Create a cluster client interface.

```
device(config-cluster-TOR-client-2)#client-interface ether 1/5
```

4. Deploy the cluster client.

```
device(config-cluster-TOR-client-2)deploy
```

TOR-B

Creating LAG-1

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create a LAG with the LAG ID option.

```
device(config)# lag 1 dynamic id 1
device(config-lag-1)#
```

2. Define the port the LAG will be using:

```
device(config-lag-1)# ports ethernet 1/6
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-1)# primary-port 1/1
```

4. Deploy a LAG as shown below.

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

Assign a name to an individual port within a LAG.

```
device(config-lag-1)# port-name lag-client-2:1/2 ethernet 1/6
```

Creating LAG 2

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
device(config)# lag 2 dynamic id 2
device(config-lag-2)#
```

2. Define the port the LAG will be using.

```
device(config-lag-2)# ports ethernet 1/7
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-2)# primary-port 1/7
```

4. Deploy a LAG as shown below.

```
device(config-lag-2)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-2)# port-name lag-client-3:1/2 ethernet 1/7
```

Creating LAG 3

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
device(config)# lag 3 dynamic id 3
device(config-lag-3)#
```

2. Define the port the LAG will be using.

```
device(config-lag-3)# ports ethernet 1/3 to 1/4
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-3)# primary-port 1/3
```

4. Deploy a LAG as shown below.

```
device(config-lag-3)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-3)# ICL-to-TOR-A:1/3 ethernet 1/3
device(config-lag-3)# ICL-to-TOR-A:1/4 ethernet 1/4
```

Creating LAG 4

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
device(config)# lag 4 dynamic id 4
device(config-lag-4)#
```

2. Define the port the LAG will be using.

```
device(config-lag-4)# ports ethernet 1/5
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-4)# primary-port 1/5
```

4. Deploy a LAG as shown below.

```
device(config-lag-4)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-3)# lag-client-1:1/2 ethernet 1/5
```

Enable layer 2 switching

By default, Extreme devices support routing over layer 2 switching. You can enable layer 2 switching globally or on individual port using the **no route-only** command. The **no route-only** and **route-only** commands prompts you for whether or not you want to change the "route-only" behavior. You must enter **y** if you want to proceed or **n** if you do not. To enable Layer 2 switching only on a specific interface, go to the interface configuration level for that interface, and add the **no route-only** command.

NOTE
On the CER 2000 Series and CES 2000 Series devices, the **no route-only** interface configuration is only valid when the interface is strictly untagged in a regular VLAN. Configure **no route-only** at the global configuration if tagged/dual-mode ports are being used in the system.

NOTE
On the XMR Series and MLX Series devices, **route-only** is the default condition. Because **route-only** is the default condition, it will not be displayed in the configuration. If you use **no route-only** to enable switching, the **no route-only** command will be displayed in the configuration.

NOTE
On the CES 2000 Series and CER 2000 Series devices, **route-only** is disabled by default. Therefore, if **route-only** is enabled on a CES 2000 Series or CER 2000 Series device, it will be displayed in the configuration.

Use the following command to enable Layer 2 switching.

```
device(config)# no route-only
```

Syntax: [no] route-only

The following warning messages are displayed by the system when there is a conflict between global and interface level route only configurations on a VLAN tagged interface.

Warning message	Configuration
`"no route-only" interface configuration conflicts with the global "route-only" configuration, This configuration will be applied on 1/5 interface when the interface becomes strictly untagged interface`	Global config : **route-only** Interface config on vlan tagged port: **no route-only**
`"route-only" interface configuration conflicts with the global "no route-only" configuration, This configuration will be applied on 1/6 interface when the interface becomes strictly untagged interface`	Global config: **no route-only** Interface config on vlan tagged port: **route-only**

Creating VLANs

See Single level MCT example on page 422

VLAN 1

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN.

```
device(config)# vlan 2
```

2. Add ports to that VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-2)# tagged e 1/1 to 1/8
device(config-vlan-2)# no untag ether 1/3 to 1/4
```

VLAN 2

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN 2.

```
device(config)# vlan 2 client-vlan
```

2. Add ports to that VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-2)# untag ether 1/5 to 1/7
device(config-vlan-2)# tagged ether 1/3 to 1/4
```

Create the session VLAN

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN 4090.

```
device(config)# vlan 4090 name Session-VLAN
```

Syntax: [no] vlan vlan-id name *vlan-name*

VLAN IDs can be in the range of 1 - 4090. Use the **no** form of the command to delete the VLAN from the configuration.

The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

In addition to a VLAN number, you can assign a name to a VLAN by entering name *vlan-name*. Enter up to 35 characters for name.

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-4090)# tagged ether 1/3 to 1/4
```

3. Configure the appropriate IP routing parameters on each of the virtual routing interfaces.

```
device(config-vlan-4090)# tagged ether 1/3 to 1/4
device(
config-vlan-4090)# router-interface ve 100
```

Assign the hostname - optional

Configure a system name for the device and save the information locally in the configuration file for future reference. The information is not required for system operation but recommended. When you configure a system name, it replaces the default system name in the CLI command prompt.

To configure a system name, enter a command such as the following.

```
device(config)# hostname TOR-B
```

Syntax: [no] hostname name

The *name* can be up to 255 alphanumeric characters. The text strings can contain blanks.

Enabling interfaces

The ports can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the ports on a device for ethernet interfaces 1/3, 1/5, 1/6,and 1/7, enter the following commands.

```
device(config)# interface ether 1/3
device(config-if-e10000-1/3)# enable
device(config)# interface ether 1/5
device(config-if-e10000-1/5)# enable
device(config)# interface ether 1/6
device(config-if-e10000-1/6)# enable
device(config)# interface ether 1/7
device(config-if-e10000-1/7)# enable
```

Syntax: [no] enable

Adding a virtual interface

Add a virtual interface and configure an IP address on the interface by entering commands such as the following.

```
device(config)# interface ve 100
device(config-vif-100)# ip address 1.1.1.2/24
```

Syntax: [no] interface [**ve** *ve-id*]

Syntax: [no] ip address *ip-addrip-mask*

The *ve-id* variable allows you to specify a VE interface ID.

Configuring the cluster operation mode

The cluster can be deployed separately without any client configured. When the cluster is deployed, it will check all the deployed clients and start the state machine for the clients. See Single level MCT example on page 422.

1. Configure one cluster ID or name on the device so that all route-reflector clients for the device become members of the cluster. To configure a device with cluster ID 1, enter the following command.

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

   Syntax: [no] cluster *cluster-name cluster-id*

   The *cluster-name* parameters specify the cluster name with a limit of 64 characters.

   The *cluster-id* parameters specify the cluster ID (1-65535). The default is the device ID.

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

   ```
   device(config-cluster-TOR)#rbridge-id 2
   ```

   Syntax: [no] rbridge-id *id*

   The *id* parameters specify the remote bridge ID. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN is in range 1-4090 but cannot be default VLAN. A check is made during the cluster deploy and in addition to a dynamic check. The default VLAN cannot be changed to a VLAN which is already defined as cluster session

```
device(config-cluster-TOR)# session-vlan
 4090
```

Syntax: [no] session-vlan *vlan-id*

The *vlan-id* parameters specify the VLAN range. Possible values are 1 - 4090.

4. Specify the VLAN range on which cluster is operating. This would be the range for which there would be MAC synchronization. Multiple VLAN ranges would be supported for the configuration. Enter a command such as the following to create the member VLAN.

```
device(config-cluster-TOR)# member-vlan
 2
```

Syntax: [no] member-vlan x to *y*

NOTE
The VLAN range is allowed to change even if cluster is deployed.

The new VLAN range will over-ride the previous configured range.

5. Specify the ICL for the cluster. The ICL interface can be a single link or trunk port. If it is a trunk port, it should be the primary port of the trunk. Only one ICL is supported.

```
device(config-cluster-TOR)#icl
 TOR ethernet 1/3
```

Syntax: [no] icl *icl-name* **ethernet** *x/y*

The *icl-name* parameter can be 64 characters (maximum).

The **ethernet** *x/y* parameter is the ICL interface.

6. Specify the rbridge and ICL for the peers by entering a command such as the following.

```
device(config-cluster-TOR)# peer 1.1.1.1 rbridge-id 1 icl TOR
```

Syntax: [no] peer *peer-ip* **rbridge-id** *peer-rbridge* **icl** *map-icl*

The *peer-ip* parameter should be in same subnet as that of cluster management interface.

The *peer-rbridge* parameter should be different from cluster rbridge and any other client in the cluster

The *map-icl* parameter is the ICL name to reach this cluster peer.

7. Clusters can be deployed separately without any client configured. The **deploy** command brings the cluster into effect. Once the cluster is deployed, the configuration inside the cluster can not be changed. The **deploy** command also preforms a consistency check of the entire cluster configuration. If anything is amiss, an error message is sent. The specific information checked during deploy:
   - If the cluster management VLAN is configured
   - If the cluster peer is configured
   - If the cluster ICL is configured

   Enter a command such as the following to deploy the cluster configuration.

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

Syntax: [no] deploy

Create cluster client 1

1. Create a cluster client instance and change the mode to the client instance. If an instance is already present, then directly change the mode to client instance mode.

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

Syntax: [no] client *client-name*

The *client-name* parameter can be 64 characters (maximum).

If it is a two port MCT, the maximum clients supported on the XMR Series or MLX Series is 1536/2.

If it is a two port MCT, the maximum clients supported on the CES 2000 Series or CER 2000 Series system is 50/2.

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

```
device(config-cluster-TOR-client-1)#rbridge-id 100
```

Syntax: [no] rbridge-id *id*

The *id* parameters specify the remote bridge id. Possible values are 1 - 35535 (16 bit value).

3. Create a cluster client interface. The interface can be a single link or trunk port. If it is trunk port, it should be the primary port of the trunk.

```
device(config-cluster-TOR-client-1)#client-interface ether 1/5
```

Syntax: [no] client-interface *interface* **interface : ethernet** *x/y*

The **ethernet***x/y* parameter is the ethernet interface.

4. Deploy the cluster client. If cluster is not deployed, the configuration will be taken but the client FSM will not be started. The consistency checks for client will be done at the time of client deploy. The following configuration checks will be preformed:

- – Client interface is configured
  – Client interface is not same as any other client interface or ICL interface
  – Client RBridge ID is not same as cluster rbridge or any peer rbridge

Once the client is deployed, the configuration inside the client will not be allowed to change, To deploy the client configuration, enter a command such as the following.

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

Syntax: [no] deploy

Create cluster client 2

See Single level MCT example on page 422 and Create cluster client 1 on page 436 for additional information on creating cluster clients.

1. Create a cluster client instance and change the mode to the client instance mode.

```
device(config-cluster-TOR)# client client-2
```

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

```
device(config-cluster-TOR-client-2)#rbridge-id 200
```

3. Create a cluster client interface.

```
device(config-cluster-TOR-client-2)#client-interface ether 1/6
```

4. Deploy the cluster client.

```
device(config-cluster-TOR-client-2)deploy
```

Create cluster client 3

See Single level MCT example on page 422 and Create cluster client 1 on page 436 for additional information on creating cluster clients.

1. Create a cluster client instance and change the mode to the client instance mode.

```
device(config-cluster-TOR)# client client-3
```

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

```
device(config-cluster-TOR-client-3)#rbridge-id 300
```

3. Create a cluster client interface. The interface can be a single link or LAG port. If it is LAG port, it should be the primary port of the LAG.

```
device(config-cluster-TOR-client-2)#client-interface ether 1/7
```

4. Deploy the cluster client.

```
device(config-cluster-TOR-client-2)deploy
```

Configuring Client-1

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create LAG 1 as shown below.

```
device(config)# lag 1 dynamic id 1
device(config-lag-1)#
```

2. Define the ports the LAG will be using.

```
device(config-lag-1)# enable ethernet 1/1 to 1/3
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-1)# primary-port 1/1
```

4. Deploy the LAG as shown below.

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

5. Assign a name to an individual port within a LAG.

```
device(config-lag-1)# port-name lag-to-TOR-A ethernet 1/1
device(config-lag-1)# port-name lag-to-TOR-B ethernet 1/3
```

6. The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the port on a device for ethernet interface 1/1, enter the following command.

```
device(config-if-e10000-1/1)# enable
```

Configuring Client 2

See **Figure 176** and Creating LAG-1 on page 423 for additional information on creating a LAG.

1. Create a LAG with the LAG ID option, enter a command such as the following.

```
device(config)# lag 1 dynamic id 1
device(config-lag-1)#
```

2. Define the ports the LAG will be using.

```
device(config-lag-1)# ports ethernet 1/1 to 1/2
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-1)# primary-port 1/1
```

4. Deploy the LAG as shown below.

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

5. Assign a name to an individual port within a LAG.

```
device(config-lag-1)# port-name lag-to-TOR-A ethernet 1/1
device(config-lag-1)# port-name lag-to-TOR-B ethernet 1/2
```

6. The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the port on a device for ethernet interface 1/1 and 1/3, enter the following commands.

```
device(config)# interface ether 1/1
device(config-if-e10000-1/1)# enable
device(config)# interface ether 1/3
device(config-if-e10000-1/3)# enable
```

7. Assign a name to an individual port within a LAG.

```
device(config-if-e10000-1/3)# port-name host-to-PC
device(config-if-e10000-1/3)# enable
```

Optional cluster operation features

A cluster can operate in two modes:

Cluster Failover Mode

Fast-failover (default) - As soon as the ICL interface goes down the CCP goes down. All the remote MACs are flushed.

Slow-failover - Even if the ICL interface goes down the CCP waits for the hold-time before making the CCP down. Remote MACs are flushed only when the CCP is down.

To disable the fast-failover mode, enter a command such as the following.

```
device(config-cluster-TOR)#peer 1.1.1.1 disable-fast-failover
```

Syntax: [no] peer *peer-ip* **disable-fast-failover**

Client isolation mode

NOTE
The CLI will allow modification of the **client-islolation** mode on MCT cluster nodes even when the cluster is deployed. You must create the same isolation mode on both cluster nodes.

The client operates in the following mode:

Strict mode: When the CCP goes down, the client interfaces on both the cluster nodes are administratively shutdown. In this mode, the client is completely isolated from the network if CCP is not operational.

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

Syntax: [no] client-isolation strict

Shutdown all client interfaces

Use the **client-interfaces shutdown** command when preforming a hitless-upgrade operation. This command can be used to shutdown all the local client interfaces in the cluster. This would result in failover of traffic to the cluster peer.

```
device(config-cluster-TOR)#client-interfaces shutdown
```

Syntax: [no] client-interfaces shutdown

Client interfaces delay

Use the **client-interfaces delay** command to set the delay before brining up the CCEP port. This command is used to set the delay, so that after a node is reloaded, with just L2vpn peer alone, the delay to bring up the CCEP port will be the designated value.

```
device(config-cluster-TOR)#client-interfaces delay 60
```

Syntax: [no] client-interfaces delay *time insec*

The default value for delay is 90 seconds. The acceptable values range between 20 to 1800 seconds.

Active/Passive mode

To configure MCT to operate in the Active/Passive mode, set the mode for the cluster. The cluster mode should be configured on both the node. If both nodes are not configured Active-Passive, then the CCP will not be brought up. This configuration is not allowed after the cluster is deployed.

```
device(config-cluster-c)# mode-active-passive
```

Syntax: [no] mode-active-passive

Client-role

The **client-role** command is used for configuring the per client role as Active or Passive. When the global and per client role configurations are present, then per client role configuration takes the highest precedence.

If Global and per client role configurations is not present, then the rbridge-id used for client role election, the node with the lowest rbridge-id will be Active.

This global command will be applicable for all the MCT clients.

```
device(config-cluster-c)#client-role active
```

This command will be applicable for a single MCT client.

```
device(config-cluster-c-client-c1)# client-role passive
```

Syntax: [**no**] **client-role** *role*

The *role* parameter specifies if the client role is active or passive.

Client-role-revertible-delay timer

The **client-role-revertible-delay timer** command is used for configuring client role revertible delay mode in case of port flaps, and to revert the client role after client role switch.

```
device(config-cluster-c-client-c1)#client-role-revertible-delay timer 5
```

Syntax: [no] client-role-revertible-delay timer *value*

Enter a value of 1-240 minutes. Default value is 1 minute.

Displaying cluster information

Use the **show cluster** command to display the entire cluster configuration and **show tech cluster** command to display cluster configuration and operation information. See the *Extreme NetIron Diagnostic Reference* for additional information on these commands.

Keep-alive VLAN

CCRR message are used to exchange information between peers. When the CCP is up, CCRR messages are sent over CCP. When the CCP client reachability is down, you can use the **keep-alive-vlan** command under cluster context so CCRR messages are periodically sent over the keep-alive-vlan. Only one VLAN can be configured as a **keep-alive-vlan** . The keep-alive VLAN cannot be a member VLAN of the MCT and this VLAN can be tagged or untagged. A port in keep-alive-vlan cannot be assigned to another VLAN.

```
device(config-cluster-TOR)#keep-alive-vlan 10
```

Syntax: [no] keep-alive-vlan *vlan-id*

The *vlan_id* parameters specify the VLAN range. Possible values are 1 - 4090.

When the CCP is down:

- If **keep-alive-vlan** is configured, then CCRR messages are sent periodically for every 1 second over that VLAN.
- When CCP is down and keep-alive vlan is configured, Master/Slave selection is based on following criteria:

1. If one node's CCEPs are up and other node's CCEPs are down then the node with local CCEPs down becomes Slave
2. Otherwise, the node with higher RBridge ID becomes Slave.
    - If no packetsare received from the peer for a period of 3 seconds, then the peer box is considered down.
    - If **keep-alive-vlan** is not configured and both the peers are up, then both peers keep forwarding the traffic independently.

Keep-alive timers and hold-time

To specify the **keep-alive timers** and **hold-time** for the peers, enter a command such as the following.

```
device(config-cluster-TOR)# peer 1.1.1.1 timers keep-alive 40 hold-time 120
```

Syntax: [no] peer *peer-ip* **timers keep-alive** *keep-alivetime* **hold-time** *hold-time*

The *peer-ip* parameter should be in same subnet as that of cluster management interface.

The *keep-alive time* parameter can be 0 to 21845 (default 30 seconds.)

The *hold-time* parameter can be 3 to 65535 (default 90 seconds) must be at least 3 times the keep alive time.

NOTE
Keep-alive-vlan and keep-alive timers are not related. The keep-alive timer is used by CCP.

L2 protocol forwarding

MCT will forward or drop L2 protocol packets when corresponding features are disabled. The packets will either be forwarded as regular multicast that floods to the VLAN or dropped. When forwarded, the packet received from ICL will not be forwarded to CCEP port if the peer MCT switch has the reachability to the same cluster client.

When designing a network, the ICL port or LAG must have enough bandwidth to support all the traffic from clients (in case of client links connected to one node failure case).

For L2 forwarding, appropriate CAM profiles need to be used to be able to program all the MAC entries into the CAM (especially when using LAG interfaces on MCT nodes for ICL and client interfaces).

By default , MCT acts as a hub for STP, or RSTP. Switches connected to MCT can run STP normally. When STP, RSTP, or MSTP is enabled, the L2 protocol forwarding configuration is ignored and has no effect.

To configure L2 protocol forwarding globally, enter a command such as the following.

```
device(config)#cluster-l2protocol-forward
```

To disable L2 protocol forwarding on an interface, enter a command such as the following.

```
device(config-if-e1000-1/2)#cluster-l2protocol-forward disable
```

To remove L2 protocol forwarding configuration on an interface, enter a command such as the following

```
device(config-if-e1000-1/2)#no cluster-l2protocol-forward <enable | disable>
```

Syntax: [no]cluster-l2protocol-forward [**enable** | **disable**]

Interface level configuration overwrites the global level configuration.

TABLE 47 L2 protocol forwarding action on an MCT and non-MCT switch

Protocol	Destination MAC	Non-MCT switch forwarding action	MCT switch forwarding action
Untagged 802.1Q BPDU	01-80-c2-00-00-00	Flood to the VLAN	Forward to ports that are enabled by the **cluster-l2protocol-forward** command.
Tagged 802.1Q BPDU	01-80-c2-00-00-00	Flood to the VLAN	Forward to ports that are enabled by the **cluster-l2protocol-forward** command.
802.1Q Provider BPDU	01-80-c2-00-00-08	Dropped on CES 2000 Series or CER 2000 Series devices. Flood to	Same as non-MCT switch

TABLE 47 L2 protocol forwarding action on an MCT and non-MCT switch (continued)

Protocol	Destination MAC	Non-MCT switch forwarding action	MCT switch forwarding action
		the VLAN on XMR Series and MLX Series devices.	
802.3 Slow Protocols (e.g. LACP)	01-80-c2-00-00-02	Dropped	Same as non-MCT switch
802.1X PAE address	01-80-c2-00-00-03	Dropped	Same as non-MCT switch
802.1Q Provider Bridge GVRP	01-80-c2-00-00-0D	Flood to the VLAN	Same as non-MCT switch
802.1AB LLDP	01-80-c2-00-00-0E	Flood to the VLAN	Same as non-MCT switch
802.1D GMRP	01-80-c2-00-00-20	Flood to the VLAN	Same as non-MCT switch
802.1Q GVRP	01-80-c2-00-00-21	Flood to the VLAN	Same as non-MCT switch
Foundry MRP (Metro Ring Protocol)	03-04-80-00-00-00	Flood to the VLAN	Same as non-MCT switch
Foundry FDP (Foundry Discovery Protocol)	01-e0-52-cc-cc-cc	Flood to the VLAN	Same as non-MCT switch
CDP	01-00-00-cc-cc-cc	Flood to the VLAN	Same as non-MCT switch
PVST	01-00-0c-cc-cc-cd	Flood to the VLAN	Same as non-MCT switch
SuperSpan	03-80-c2-xx-xx-00	Flood to the VLAN	Same as non-MCT switch
VSRP Control	03-04-80-00-01-00	Flood to the VLAN	Same as non-MCT switch
VSRP Source	03-04-80-00-01-01	Flood to the VLAN	Same as non-MCT switch
Loop detection MAC	Base MAC address | 0x03000000	Flood to the VLAN	Same as non-MCT switch

Port loop detection

Port loop detection is used to detect L2 loops in MCT (due to misconfiguration). When using MCT, it requires the ICL ports to be strictly tagged. The port loop detection feature supports strictly tagged ports.

Loop detection for specific VLAN on a port

Strict mode loop detection can be configured on a specific VLAN for a given port. To configure loop detection on VLAN 10 for interface 1/1, enter a command such as the following.

```
device(config-if-e1000-1/1)#loop-detection vlan 10
```

Syntax: [no] loop-detection [**vlan** *vlan_id*]

Where **vlan-id** enables Loose Mode configuration for a VLAN group.

A port can be tagged or untagged member of this VLAN.

Multiple VLANs can have loop detection configured for a given port. Loop detection BPDUs will be sent out of each configured VLAN on that port.

Loop detection shutdown-disable

Use the **loop-detection shutdown-disable** command to disable the port shutdown feature in case of loop detection. This feature will ensure that the ICL stays up when a loop detection PDU is received on the ICL. This command will be applied to both strict mode or loose mode loop detection. To configure **loop-detection shutdown-disable** to shutdown port 1/1 used for the ICL link, enter a command such as the following.

```
device(config-if-e1000-1/1)#loop-detection shutdown-disable
```

Syntax: loop-detection shutdown-disable

Loop-detection shutdown-sending-port

By default, the receive-port is shutdown by loop detection. The **loop-detection shutdown-sending-port** command will shutdown the port that sent the loop detection PDUs instead of shutting down the receiving port. This will ensure that the ICL stays up when a loop detection PDU is received on the ICL.

This feature is only applicable to strict mode loop detection.

```
device(config-if-e1000-1/1)#loop-detection shutdown-sending-port
```

Syntax: [no] loop-detection shutdown-sending-port

Loop-detection-syslog-duration

If any of the ports has shutdown disabled, any loop detection will be logged into the syslog. Since the port is not shutdown, loop detect PDUs will come at a very fast rate and entries into the syslog are throttled.

By default, syslog-duration is 10 minutes. The configurable range is from 10 minutes to 1440 minutes.This is a global command and any changes will be applied to all interfaces.To configure **loop-detection-syslog-duration** for every 30 minutes, enter a command such as the following.

```
device(config)# loop-detection-syslog-duration 30
```

Syntax: [no] loop-detection-syslog-duration *mins*

The *mins* parameter specifies the configurable range which is from 10 minutes to 1440 minutes.

MCT failover scenarios

1. ICL interface or CCP goes down (Keep alive configured)

   When the keepalive VLAN is used and finds the cluster nodes reachability when the ICL or CCP goes down. If the peer node is reachable over keepalive VLAN, the MCT nodes perform performs the Master/Slave negotiation per client. After negotiation, the Slave shuts down its client ports whereas the Master client ports continue to forward the traffic.

   The Master/Slave negotiation is done per MCT client on the basis of RBridge Id and client Local or Remote reachability. If the client is reachable from both MCT nodes, the higher RBridge Id becomes the Master. If client is reachable from one of the MCT nodes, only then the node on which it is reachable becomes the Master.

   If the peer is not reachable over the keepalive VLAN, then both cluster nodeswill keep forwarding.

   NOTE
   Extreme recommends to use keepalive VLANs with the MCT configurations. This will provide a backdoor reachability ifthe ICL interface goes down.

2. ICL interface or CCP goes down (Keep alive not configured)

   When the keepalive VLAN is not configured, both cluster nodes will keep forwarding. Use the **client-isolation strict** command to remove the client interface as soon as ICL goes down and isolate the client completely

3. MCT node goes down.

   When the MCT nodes goes down, the traffic will failover to the other MCT node.

4. Hitless failover performed on one of the MCT nodes

   Traffic is switched over to the other node. However, the CCP will go down and come back up again once the hitless failover is completed.

   Use the **client-interfaces shutdown** command to shutdown all the client interfaces so that the traffic failovers to the other MCT node first. Then perform the hitless failover.

5. Client interface on one of the MCT node goes down

   When hitless failover happens on an MLX Series and XMR Series, that node flushes all the MACs and will reestablish cluster CCP session. In this case, the user may notice some traffic impact.

6. Double failures - The ICL goes down and client interface goes down on one MCT node.

   Multiple failures could drop traffic in this scenario even if there is actual physical path is available.

Show commands

Use the **show cluster** command to display the peer and client states.

```
device#show cluster
Cluster CLUSTER-1 2000
==================
Rbridge Id: 35535, Session Vlan: 2001, Keep-Alive Vlan: 301
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range: 2 to 2000 2002 to 4090
Active Member Vlan Range: 2 to 3 21 to 148 201 404 to 445 501 to 508 2010 3511 to 3574 4021 to 4025 4051
4070 4080 4087 4090
 ICL Info:
 ---------
 Name            Port  Trunk
 ICL-1           2/1   6
```

```
 Peer Info:
 ----------
 Peer IP: 1.1.1.1, Peer Rbridge Id: 1, ICL: ICL-1
 KeepAlive Interval: 50 , Hold Time: 300, Fast Failover
 Active Vlan Range:  2 to 3 21 to 148 201 404 to 445 501 to 508 2010 3511 to 3574 4021 to 4025 4051 4070
4080 4087 4090
 Peer State: CCP Up (Up Time:   0 days:19 hr:24 min: 8 sec)
 Client Info:
 ------------
 Name            Rbridge-id Config     Port  Trunk FSM-State
 Client1         2222       Deployed   1/2     3   Up
 Client2         222        Deployed   1/40    -   Up
```

Syntax: show cluster

Use the **show cluster client** command to display additional State Machine information including the reason for Local CCEP down.

```
device#show cluster mct client c2
...
State: Remote Up
Reason for Local CCEP down: "client-interfaces shutdown
" command
Number of times Local CCEP down: 2
Number of times Remote CCEP down: 1
Number of times Remote Client undeployed: 1
Total CCRR packets sent: 12
Total CCRR packets received: 13
```

Syntax: show cluster client

The following reasons are displayed for Local CCEP down.

TABLE 48 Reason for Local CCEP down

Reason for Local CCEP down	means....
client-interfaces shutdown	command is configured
client-isolation strict	command is configured
Deploy mismatch	Client is not deployed remotely
Slave state	Client is in Slave State when CCP is down
cluster and client undeployed	Neither the Cluster or Client is deployed.
cluster undeployed	Cluster is not deployed
client undeployed	Client is not deployed

Use the **show cluster client disabled** command to display the MCT spokePW state for disabled CCEP ports. When the MP switchover occurs in the remote peer of the L2VPN MCT, the CCEP ports are disabled and then enabled after the switchover. During the switchover, the local CCEP is in disabled state because of the client-interface delay, even though the CCEP ports are UP. The **disabled** option displays the actual port spokePW state, UP or DOWN, instead of the configured state for both L2 and L2VPN client ports.

```
device#show cluster 1 client disabled
Name   Rbridge-id  Config    Port   Trunk  FSM-State   SpokePW-state

 R3     11       Deployed    3/3    3      Remote Up   DOWN
```

Syslogs and debugging

The following system log messages are displayed when the remote Cluster Client Edge Port (CCEP) state is changed or the remote client is deployed or undeployed.

```
SYSLOG: Jun 1 15:43:36:<14>Jun 1 15:43:36 CES, CLUSTER FSM: Cluster mct (Id: 1), client c2 (RBridge Id: 4) -
Remote client deployed
SYSLOG: Jun 1 16:04:24:<14>Jun 1 16:04:24 CES, CLUSTER FSM: Cluster mct (Id: 1), client c2 (RBridge Id: 4) -
Remote client CCEP up
```

CCEP syslog messages generated during the LACP delay state

The following system log messages are displayed during the LACP delay for different states of the CCEP.

```
Aug 23 23:19:48:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP up
Aug 23 23:19:48:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Local client CCEP up
Aug 23 23:19:48:I:System: Interface ethernet 4/5, state up
Aug 23 23:19:48:W:LACP: ethernet 4/5 state changes from LACP-BLOCKED to FORWARD
Aug 23 23:19:43:W:LACP: ethernet 4/5 state changes from DOWN to LACP-BLOCKED
Aug 23 23:19:48:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP up
Aug 23 23:19:45:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP up
Aug 23 23:19:12:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP down
```

Sample configuration

The output below is a sample configuration using port loop detection.

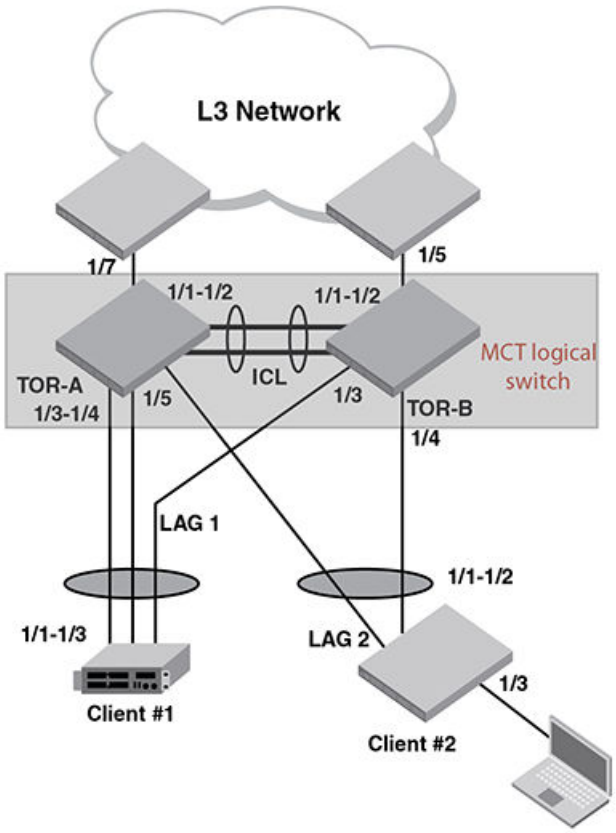
```
device#show run
lag "icl1" dynamic id 1
ports ethernet 3/20 ethernet 4/9
primary-port 3/20
deploy
!
vlan 1 name DEFAULT-VLAN
no untagged ethe 3/20 ethe 4/9
!
vlan 10
tagged ethe 3/20 ethe 4/9
router-interface ve 10
!
vlan 11
untagged ethe 4/17
tagged ethe 3/11 ethe 3/20 ethe 4/9
loop-detection
!
vlan 15
tagged ethe 3/20 ethe 4/9
!
vlan 20
tagged ethe 3/11 ethe 3/20 ethe 4/9 ethe 4/17
!
no route-only
logging console
telnet server
loop-detection-interval 1
loop-detection-disable-duration 1
loop-detection-syslog-duration 11
!
interface ethernet 3/20
loop-detection shutdown-disable
loop-detection vlan 20
!
interface ethernet 4/17
enable
!
```

```
loop-detection shutdown-sending-port
loop-detection vlan 20
loop-detection vlan 11
!
interface ve 10
ip address 10.10.10.1/24
!
!
!
cluster abc 1
rbridge-id 100
session-vlan 10
keep-alive-vlan 30
member-vlan 11 to 20
member-vlan 40 to 50
icl icl1 ethernet 3/20
peer 10.10.10.2 rbridge-id 200 icl icl1
client c1
  rbridge-id 300
  client-interface ethernet 3/11
!
```

Failover scenarios for Layer 2 multicast over MCT

Figure 144 shows an example multicast snooping configuration.

FIGURE 142 Multicast snooping over MCT



The following failure modes can occur for Layer 2 multicast over MCT.

Local CCEP down event:

- Outgoing traffic on local CCEP will now go through ICL and go out of remote CCEP.
- Incoming traffic on local CCEP will now ingress through remote CCEP, and then ingress through ICL locally.

Local CCEP up event:

- Outgoing traffic on remote CCEP (after egressing through local ICL) will now start going out of local CCEP.
- Incoming traffic from client through ICL (after ingressing on remote CCEP) will now switch back to local CCEP (this is true only if the client trunk hashing sends the traffic towards local CCEP). CCP (Cluster communication protocol) Down event:
- All related information (i.e. IGMP/MLD group, mcache, router port, static port, pim-sm snooping entry) that was synced from the MCT peer will now be marked for aging locally.

Multicast show commands

Use the **show ip pim mcache** command to display if a CCEP port is being blocked or being forwarded to.

```
device#show ip pim mcache
IP Multicast Mcache Table
Entry Flags    : SM  - Sparse Mode, SSM - Source Specific Mutlicast, DM - Dense Mode
                 RPT    - RPT Bit, SPT - SPT Bit, LSRC - Local Source, LRCV - Local Receiver, JOIN - Join
Upstream
                 HW - HW Forwarding Enabled, FAST - Resource Allocated, TAG - Need For Replication Entry
                 REGPROB - Register In Progress, REGSUPP - Register Suppression Timer
                 MSDPADV - Advertise MSDP, NEEDRTE - Route Required for Src/RP,  PRUN - DM Prune Upstream
Interface Flags: IM - Immediate, IH - Inherited, WA - Won Assert
                 MJ - Membership Join, MI - Membership Include, ME - Membership Exclude
                 BR - Blocked RPT, BA - Blocked Assert, BF - Blocked Filter, BI - Blocked IIF,
             BM - Blocked MCT
Total entries in mcache: 1
1     (2.2.2.101, 239.0.1.3) in v200 (e2/15), Uptime 00:01:08, Rate 42229 (SM)
      Source is directly connected. RP 2.2.2.1
      Flags (0x3046cec1) SM SPT L2REG LSRC LRCV JOIN HW FAST MSDPADV
      fast ports: ethe 2/1
      AgeSltMsk: 00000002,  FID: 0x8006, MVID: NotReq, RegPkt: 0, AvgRate: 41688, profile: none
      Forwarding_oif: 1, Immediate_oif: 1, Blocked_oif: 1
      L2 (HW) 1:
          TR(e2/1,e2/1),  00:01:08/181, Flags: IM IH
      Blocked OIF 1:
          TR(e1/5,e1/5)(VL200), 00:01:08/0, Flags: MJ BM
Number of matching entries: 1
device#
```

The **show ip igmp interface** command has been enhanced to show the IGMP Query suppression state on CCEP ports.

```
device#show ip igmp interface
---------+------+---------+---------------+---------+-----+-----+---------
Intf/Port|Groups| Version |Querier        |  Timer  |V1Rtr|V2Rtr|Tracking
         |      |Oper  Cfg|               |OQrr GenQ|     |     |
---------+------+----+----+---------------+----+----+-----+-----+---------
v200           2    2    -                                      Disabled
    e2/15           2    - Self               0   59 No    No
     e2/1           2    - Self               0   59 No    Yes
     e1/5           2    - Self (MCT-Blk)
     0   40 No    No
device#
```

The IPv6 version of the command **show ipv6 mld interface** has been similarly enhanced.

```
device#show ipv6 mld interface
---------+------+---------+-----------------------------------------+---------+-----+---------
Intf/Port|Groups| Version |Querier                                  |  Timer  |V1Rtr|Tracking
         |      |Oper  Cfg|                                         |OQrr GenQ|     |
---------+------+----+----+-----------------------------------------+----+----+-----+---------
```

```
v62             0    2    2                                                    Disabled
     e2/1            2    - Self  (MCT-Blk)                    0   79 No
    e1/37            2    - Self                               0  108 No
    e1/33            2    - Self                               0  108 No
device#
```

MAC operations

This section describes MAC related configuration operations

MAC Database Update (MDUP)

The MACs that are learned locally are given the highest priority or the cost of 0 so they are always selected as best MAC.

Each MAC is advertised with a cost. Low cost MACs are given preference over high cost MACs.

If a MAC moves from a CCEP port to a CEP port, a MAC move message is sent to the peer and the peer moves the MAC from its CCEP ports to the ICL links.

When peer MACs in a cluster become out of sync for any client, you can recover by performing the following tasks, which are described in the following sections:

- Perform periodic health checks on both peers to determine if the MAC table for any client is out of sync.
- Manually or dynamically synchronize the MAC table.
- Display synchronization details using show commands.

Enabling MAC health check

NOTE
Client health check is disabled by default.

To enable client health check for all configured clients in cluster mode, enter the **client-health-check** command.

```
device(config-cluster-TORA)# client-health-check
```

Syntax: [no] client-health-check

Disabling MAC health check

The **no** version of the client-health-check command disables the client health check for all configured clients in cluster mode.

```
device(config-cluster-TORA)# no client-health-check
```

Syntax: [no] client-health-check

Configuring the health check timer

To configure the periodic timer value of the cluster MAC's synchronization, enter the **client-health-check** command. Time values range from 30 through 120 seconds and 60 seconds is the default.

```
device(config-cluster-TORA)#client-health-check timer 60
```

Syntax: [no]client-health-check [**timer**] [*value inseconds*]

Disabling the health check timer

The **no** version of the **client-health-check timer** command sets the timer to the default of 60 seconds.

```
device(config-cluster-TORA)# no client-health-check timer 60
```

Syntax: [no] client-health-check [**timer**] [*value inseconds*]

Enabling dynamic MAC learning

When the MAC learning mode is set to dynamic and when the remote peer learns that remote client MAC entries are out of sync, a MDUP INFO message is dynamically sent for that particular client. If the dynamic learning mode is disabled, the learning mode is set to the default of manual and the MAC entries must be synchronized manually.

To set the MAC learning mode to dynamic, enter the **client-health-check learning-mode** command

```
device(config-cluster-TORA)#client-health-check learning mode dynamic
```

Syntax: [no] client-health-check learning-mode [**dynamic** | **manual**]

Disabling dynamic MAC learning

The no version of the **client-health-check learning-mode** command sets the client health check learning mode to the default of manual and disables the dynamic MAC entries' synchronization.

```
device(config-cluster-TORA)#no client-health-check learning mode dynamic
```

Syntax: [no] client-health-check learning-mode [**dynamic** | **manual**]

Manually synchronizing MAC entries and MCT peers

When the MAC database becomes asynchronous between the MCT peers, you can manually synchronize MAC entries specific to various combinations of the cluster, client, and VLANs, as follows:

- All interface MAC entries
- VLAN interface MAC entries
- ALL MAC entries
- VLAN MAC entries
- Client MAC entries
- VLAND and Client MAC entries
- All Client MAC entries
- Single MAC entry

Manually synchronizing all interface MAC entries

The **cluster sync-cluster-intf-mac** command synchronizes all interface MAC entries between the MCT peers of a cluster.

```
device#Cluster sync-cluster-intf-mac
Cluster - All Interface MAC's Requested from peer
device#
```

Manually synchronizing VLAN interface MAC entries

The **cluster sync-cluster-intf-mac vlan-id** command synchronizes the interface MAC entries for the specified VLAN ID between MCT peers. Basic validation occurs to ensure the VLAN ID is within the valid range.

```
device#Cluster sync-cluster-intf-mac vlan-id 3
Cluster - Interface MAC's Requested from peer for vlan-id: 3
device#
```

Manually synchronizing all MAC entries

The **cluster sync-cluster-mac** command synchronizes the entire MAC database between MCT peers.

```
device#Cluster sync-cluster-mac
Cluster -  All Macs Requested from peer
device
```

Manually synchronizing VLAN MAC entries

The cluster sync-cluster-mac vlan-id command synchronizes all MAC entries associated with the specified VLAN between MCT peers.

```
device#Cluster sync-cluster-mac vlan-id 5
Cluster - Macs Requested from peer for Vlan: 5
device#
```

Manually synchronizing client MAC entries

The **cluster sync-cluster-mac client-rbridge-id** command synchronizes all MAC entries associated with the specified client between MCT peers.

```
device#Cluster sync-cluster-mac client-rbridge-id 100
Cluster - Macs Requested from Peer for client rbridge-id: 100
```

Manually synchronizing VLAND and client MAC entries

The **cluster sync-cluster-mac vlan-id** and **client-rbridge-id** commands synchronize all MAC entries associated with the specified client and a VLAN between MCT peers.

```
device#cluster sync-cluster-mac vlan-id 5 client-rbridge-id 200
Cluster Macs Requested from peer for Vlan: 5 and client rbridge-id: 200
```

Manually synchronizing all client MAC entries

The **cluster sync-cluster-mac client-all** command synchronizes all configured client MAC entries between MCT peers.

```
device#Cluster sync-cluster-mac client-all
Cluster - Macs Requested from Peer for all Clients
```

Manually synchronizing a single MAC entries

The **cluster sync-cluster-mac mac** command synchronizes the specified MAC entry specific to a VLAN from the MCT peer.

```
device#Cluster sync-cluster-mac mac 001b.eda4.1d41 vlan-id 8
Cluster - Mac Update Requested from peer
```

Set the client-interfaces delay value

Use the **client-interfaces delay** command to set the delay before brining up the CCEP port. This command is used to set the delay, so that after a node is reloaded, with just L2vpn peer alone, the delay to bring up the CCEP port will be the designated value.

```
device(config-cluster-TOR)#client-interfaces delay 60
```

Syntax: [no] client-interfaces delay *time in sec*

The default value for delay is 30 seconds. The acceptable values range between 20 to 600 seconds.

NOTE
Client-interface delay is only applied with just L2 VPN. It does not support L2+L2VPN.

Enabling Cluster MAC synchronization

MAC table learning allows periodic synchronization between MCT peers, based on a configured value (in minutes). If cluster MAC synchronization is enabled but no value is specified, the default synchronization value is 15 minutes.

NOTE
Cluster MAC synchronization is disabled by default.

To enable cluster MAC synchronization for all configured clients in cluster mode, enter the **cluster-mac-sync** command.

```
device(config-cluster-TORA)# cluster-mac-sync
```

Syntax: cluster-mac-sync

Disabling Cluster MAC synchronization

The **no** version of the **cluster-mac-sync** command disables the cluster MAC synchronization for all configured clients in cluster mode.

```
device(config-cluster-TORA)# no cluster-mac-sync
```

Syntax: [no]cluster-mac-sync

Configuring the Cluster MAC synchronization timer

To configure the timer value of the cluster MAC's synchronization, enter the **cluster-mac-sync** command. Time values range from 5 through 60 minutes and 15 minutes is the default.

```
device(config-cluster-TORA)#cluster-mac-sync timer 45
```

Syntax: cluster-mac-sync [**timer**] [*value inseconds*]

Disabling the Cluster MAC synchronization timer

The **nocluster-mac-sync** command sets the timer to the default of 15 minutes.

Syntax: [no] cluster-mac-sync [**timer**] [*value inseconds*]

Cluster MAC types

Cluster Local MAC (CL): MACs that are learned on VLANs that belongs to cluster VLAN range and on CEP locally.

MACs are synchronized to the cluster peer and are subject to aging.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/1 0 20 CL Default_ESI
```

Cluster Remote MAC (CR): MACs that are learned via MDUP message from the peer (CL on the peer) The MACs are always programmed on the ICL port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for these MACs with a cost of 1, and associated with the peer rbridge id.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CR
Default_ESI
```

Cluster Client Local MAC (CCL): MACs that are learned on VLANs that belongs to cluster VLAN range and on CCEP ports.

The MACs are synchronized to the cluster peer and are subject to aging. A MDB entry is created for these MACs with a cost of 0 and are associated with the client and cluster rbridge IDs.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCL
Default_ESI
```

Cluster Client Remote MAC (CCR): MACs that are learned via MDUP message from the peer (CCL on the peer) The MACs are always programmed on the corresponding CCEP port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for the MACs with the cost of 1, and are associated with the client and peer rbridge ids.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCR
Default_ESI
```

MAC aging

Only the local MAC entries are aged on a node. The remote MAC entries will be aged based on explict MDUP messages only.

The remote MACs learned through MDUP messages are dynamic MACs with the exception that they never age from FDB.

MAC flush

If the CEP port is down, the MACs are flushed and individual MAC deletion messages are sent to the Peer.

If the CCEP local port is down, the MACs are flushed locally and individual MAC deletion messages are sent to peer.

If the **clear mac** command is given, all the MDB and FDB are rebuilt.

If the **clear mac vlan** command is given, all the local MDB and FDB are rebuilt for that VLAN.

MAC movement happens normally on the local node.

CEP to CCEP MAC movement - MAC movement normally happens on the local node, and deletes all the other MDBs from the peer to create a new local MDB.

CCEP to CEP MAC movement - MAC movement happens normally on the local node and delete all the other MDBs from the peer to create a new local MDB.

Flooding support on VLANs

The NetIron OS device supports the existing VLAN hardware flooding features such as unknown-unicast-flooding and vlan-cpu-protection on cluster VLANs. However, some changes were made to the way CAM entries are programmed. To support MCT cluster VLANs, the following changes to how the CAM is programmed:

- If the ICL port is part of a PPCR, then the device will program the specific (port or VLAN) based hardware flooding CAM entries on that PPCR. This is to avoid duplicate hardware flooding packets to be sent to CCEP ports.
- On an ICL port, the FID in pram will point to MCT_VLAN_CCEP_CONTROL_FID
- On non-ICL port, the FID in pram will point to VLAN_FID

Handling the MAC mismatch scenario in MCT

To handle a MAC address mismatch in the Layer 2 Ethernet header and the ARP sender MAC address in MCT, configure the static MAC in the CCEP port to avoid the traffic impact.

When the CCEP port goes down, the configuration moves the static MAC address from CCEP to the ICL port during a CCEP port shutdown. When the CCEP is up, the static MAC address moves the static MAC address from the ICL port to the CCEP.

The MAC mismatch configuration is supported on the XMR Series, MLX Series, CES 2000 Series, and CER 2000 Series devices.

Configuration steps

1. Set up the MCT topology and ensure that the MCT cluster is up and running.
2. Configure the static MAC address on the local CCEP similar to the following example.

```
device(config)# vlan 200
device(config)# static-mac-address 0001.2222.2323 ethernet 1/1
```

3. Enter the **cluster-client-static-mac-move** command for static MAC address movement on both the MCT peer switches in the cluster.

The syslog message helps you identify the root cause for the traffic outage scenario and you can proceed with the static MAC address workaround in MCT by configuring the static MAC address in the CCEP port. The following syslog message is displayed when there is a MAC address mismatch.

```
SYSLOG: <14>Dec 16 05:53:23 MLX_1 MAC_MISMATCH_DETECTION: ARP pkt received with diff eth source MAC
and diff ARP sender MAC. Eth src MAC: 0024.3892.4c02 ARP sender MAC: 0034.2867.2c01.
```

Show Commands

To display all MAC entries, use the **show mac** command as shown below:

```
device# show mac
Total active entries from all ports = 120000
Type Code - ST:Static SEC:Secure 1x:Dot1x NA: NotAvail A:Allow D:Deny
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
Port Type - CEP:Customer Edge PNP:Provider Network BEP:Backbone Edge
BNP:Backbone Network
Vlan Type - C:Customer S:Service B:Backbone I:ISID
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0101.84ff 1/13 0 20 CCL
Default_ESI
0000.0100.4780 1/13 0 20 CCR
Default_ESI
0000.0800.0663 1/14 0 20 CL
Default_ESI
```

```
0000.0800.0870 1/1 0 20 CR
Default_ESI
```

Syntax: show mac

To display all the Cluster Local MAC entries for a cluster, use the **show mac cluster** command as shown below:

```
device#show mac cluster abc
Total Cluster Enabled(CL+CR+CCL+CCR) MACs: 451
Total Cluster Local(CL) MACs: 100
Total Cluster Remote(CR) MACs: 151
Total Cluster Client Macs(CCL+CCR) for all clients: 200
Total Cluster Client Local(CCL) MACs for all clients: 200
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCL
Default_ESI
0000.0800.3500 1/13 0 20 CCR
Default_ESI
```

Syntax: show mac [**cluster** *id* | *name* **local** | **remote**]

Clear MAC commands

To clear all MACs in the system, enter a command such as the following.

```
device#clear mac
```

Syntax: clear mac

Clear cluster specific MACs

To clear cluster specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster TOR 1 local
```

Syntax: clear mac cluster *cluster-id* | *cluster-name* { **local** | **remote** }

Clear client specific MACs

To clear client specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster TOR 1 client 1 local
```

Syntax: clear mac cluster *cluster-id* | *cluster-name* **client** *client-name* { **local** | **remote** }

Clear VLAN specific MACs

To clear VLAN specific MACs in the system, enter a command such as the following.

```
device#clear mac vlan 2
```

Syntax: clear mac vlan *vlan_id*

Clear cluster VLAN specific MACs

To clear cluster VLAN specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster cluster TOR 1 vlan 1 local
```

Syntax: clear mac cluster *cluster_id* | *cluster-name* **vlan** *vlan_id* { **Local** | **Remote** }

Clear cluster client vlan specific MACs

To clear cluster client specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster TOR 1 vlan 2 client client 1 local
```

Syntax: clear mac cluster *cluster_id* | *cluster-name* **vlan** *vlan_id* **client** *client_name* { **Local** | **Remote** }

Displaying MDUP packet statistics

To display the statistics of MDUP packets, enter a command such as the following.

```
device#show mac mdup-stats
MDUP Information
================
MDUP Data buffers in queue : 0
MDUP Statistics
===============
MDUP Update Messages sent: 7
Add Mac sent: 20
Del Mac sent: 0
Move Mac sent: 0
MDUP Mac Info Messages sent: 1
MDUP Flush Messages sent: 1
MDUP Synch Messages sent: 0
MDUP Update Messages received: 3
Add Mac received: 40
Del Mac received: 0
Move Mac received: 0
MDUP Mac Info Messages received: 0
MDUP Flush Messages received: 0
MDUP Synch Messages received: 0
```

Syntax: show mac mdup-stats

Clearing the statistics of MDUP packets

To clear the statistics of MDUP packets, enter a command such as the following.

```
device# clear mac mdup-stats
```

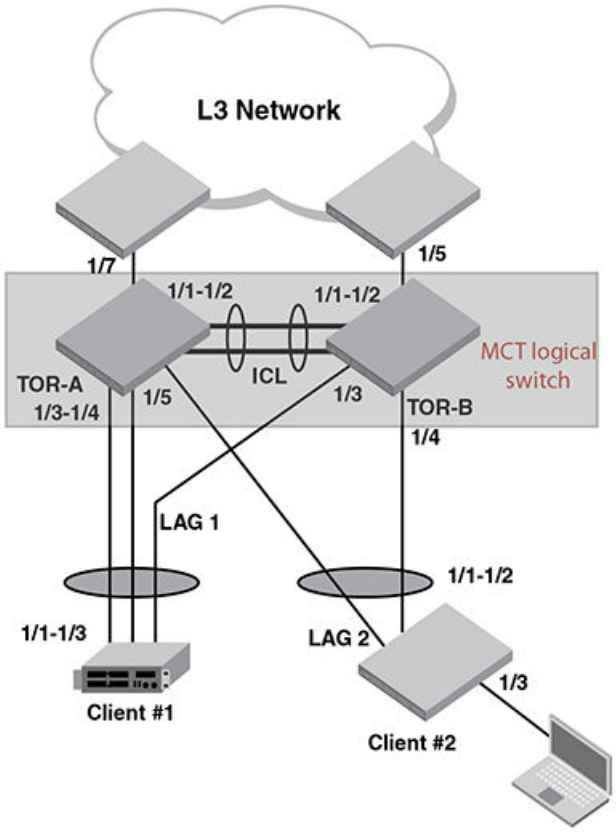
Syntax: clear mac mdup-stats

MCT configuration examples

The following examples displays the module provisioning information from a configuration file:

Single level MCT example

FIGURE 143 Single level MCT



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-B:1/1" ethernet 1/1
 port-name "ICL-to-TOR-B:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/1" ethernet 1/3
 port-name "lag-client-1:1/2" ethernet 1/4
!
lag "3" dynamic id 3
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-2:1/1" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
```

```
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/7
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 port-name CEP-PC
 enable
!
interface ethernet 1/7
 port-name to-L3-ECMP
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/5
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-A:1/1" ethernet 1/1
 port-name "ICL-to-TOR-A:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/3" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4
 primary-port 1/4
 deploy
```

```
 port-name "lag-client-2:1/2" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ethernet 1/5
 port-name to-L3-ECMP
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

Client-1:

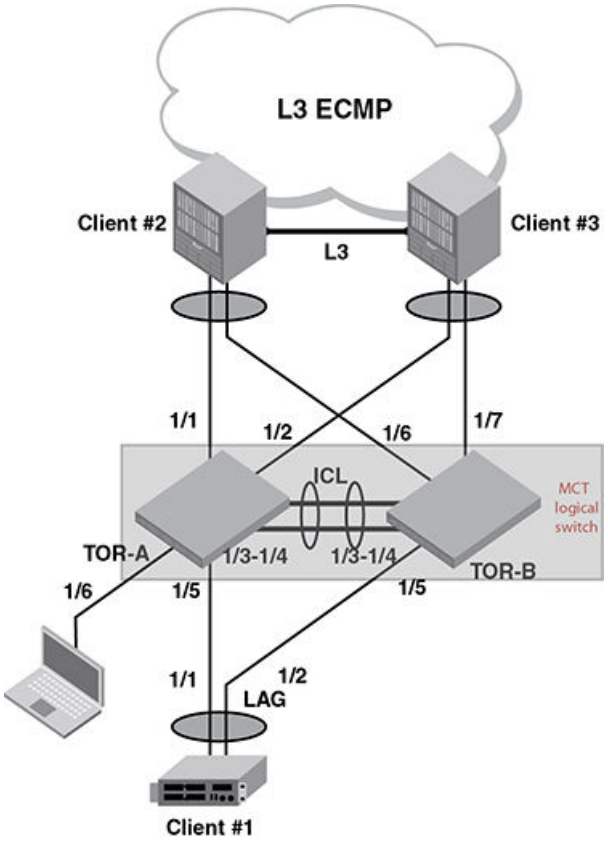
```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/3
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-A" ethernet 1/2
 port-name "lag-to TOR-B" ethernet 1/3
!
interface ethernet 1/1
 enable
!
end
----------------------------------------
```

Client-2:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name to-Host-PC
 enable
!
end
```

Single level MCT- extension example

FIGURE 144 Single level MCT- extension



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1
 primary-port 1/1
 deploy
```

```
 port-name "lag-client-2:1/1" ethernet 1/1
!
lag "2" dynamic id 2
 ports ethernet 1/2
 primary-port 1/2
 deploy
 port-name "lag-client-3:1/1" ethernet 1/2
!
lag "3" dynamic id 3
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "ICL-to-TOR-B:1/3" ethernet 1/3
 port-name "ICL-to-TOR-B:1/4" ethernet 1/4
!
lag "4" dynamic id 4
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-1:1/1" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/3 to 1/4
!
vlan 2 name client-VLAN
 untagged ethe 1/1 to 1/2 ethe 1/5 to 1/6
 tagged ethe 1/3 to 1/4
!
vlan 4090 name Session-VLAN
 tagged ethe 1/3 to 1/4
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/2
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 port-name CEP-PC
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/3
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/5
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/1
  deploy
 client Client-3
```

```
  rbridge-id 300
  client-interface ethernet 1/2
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/6
 primary-port 1/6
 deploy
 port-name "lag-client-2:1/2" ethernet 1/6
!
lag "2" dynamic id 2
 ports ethernet 1/7
 primary-port 1/7
 deploy
 port-name "lag-client-3:1/2" ethernet 1/7
!
lag "3" dynamic id 3
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "ICL-to-TOR-A:1/3" ethernet 1/3
 port-name "ICL-to-TOR-A:1/4" ethernet 1/4
!
lag "4" dynamic id 4
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-1:1/2" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/3 to 1/4
!
vlan 2 name client-VLAN
 untagged ethe 1/5 to 1/7
 tagged ethe 1/3 to 1/4
!
vlan 4090 name Session-VLAN
 tagged ethe 1/3 to 1/4
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 enable
!
interface ethernet 1/7
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/3
```

```
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/5
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/6
  deploy
 client Client-3
  rbridge-id 300
  client-interface ethernet 1/7
  deploy
!
end
----------------------------------------
```

Client-1:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
interface ethernet 1/1
 enable
!
end
----------------------------------------
```

Client-2:

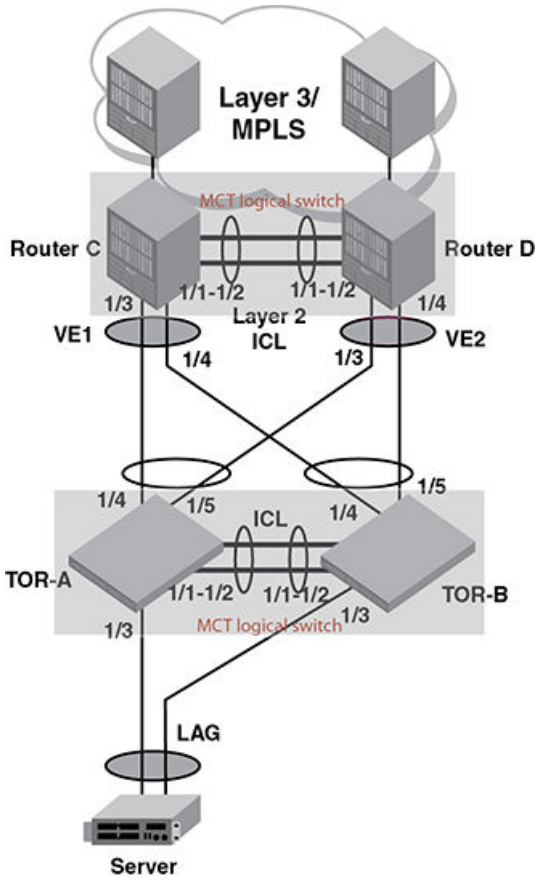
```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
vlan 2
 untagged ethe 1/1 to 1/3
 router-interface ve 2
!
router ospf
 area 0
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name L3-ECMP-Cloud
!
interface ve 2
 ip address 10.10.10.1/24
 ip ospf area 0
!
end
----------------------------------------
```

Client-3:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
vlan 2
 untagged ethe 1/1 to 1/3
 router-interface ve 2
!
router ospf
 area 0
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name L3-ECMP-Cloud
!
interface ve 2
 ip address 10.10.10.2/24
 ip ospf area 0
!
end
```

Two level MCT example

FIGURE 145 Two level MCT



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-B:1/1" ethernet 1/1
 port-name "ICL-to-TOR-B:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-Server:1" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4 to 1/5
 primary-port 1/4
 deploy
 port-name "lag-Router-C:1/3" ethernet 1/4
 port-name "lag-Router-D:1/3" ethernet 1/5
!
no route-only
!
```

```
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Server-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Routers
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-A:1/1" ethernet 1/1
 port-name "ICL-to-TOR-A:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-Server:2" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4 to 1/5
 primary-port 1/4
 deploy
 port-name "lag-Router-C:1/4" ethernet 1/4
 port-name "lag-Router-D:1/4" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
```

```
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Server-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Routers
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

Router C:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Router-D:1/1" ethernet 1/1
 port-name "ICL-to-Router-D:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-TOR-A:1/4" ethernet 1/3
 port-name "lag-TOR-B:1/4" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
```

```
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MPLS-Cloud
 enable
!
interface ve 100
 ip address 1.1.1.3/24
!
!
cluster Router 2
 rbridge-id 3
 session-vlan 4090
 member-vlan 2
 icl Router ethernet 1/1
 peer 1.1.1.4 rbridge-id 4 icl Router
 deploy
 client TOR
  rbridge-id 1
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

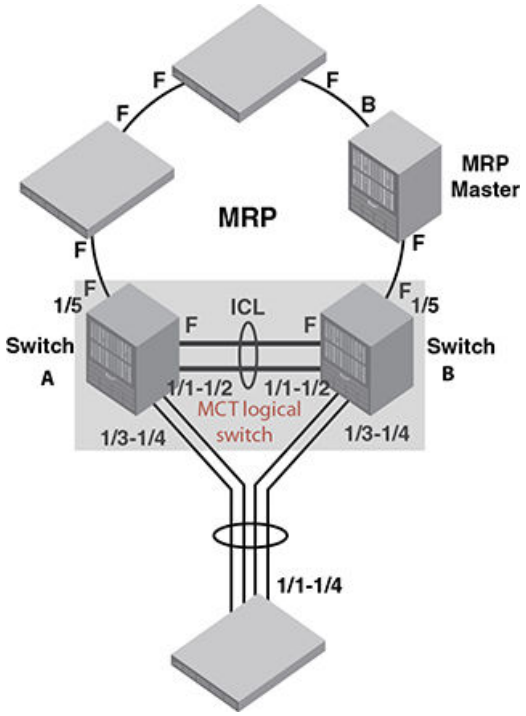
Router D:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Router-C:1/1" ethernet 1/1
 port-name "ICL-to-Router-C:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-TOR-A:1/5" ethernet 1/3
 port-name "lag-TOR-B:1/5" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
```

```
!
interface ethernet 1/5
 port-name MPLS-cloud
 enable
!
interface ve 100
 ip address 1.1.1.4/24
!
!
cluster Router 2
 rbridge-id 4
 session-vlan 4090
 member-vlan 2
 icl Router ethernet 1/1
 peer 1.1.1.3 rbridge-id 3 icl Router
 deploy
 client TOR
  rbridge-id 1
  client-interface ethernet 1/3
  deploy
!
end
```

MRP integration with MCT example

FIGURE 146 MRP integration with MCT



MCT-capable-switch-A

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-2:1/1" ethernet 1/1
 port-name "ICL-to-Switch-2:1/2" ethernet 1/2
```

```
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/1" ethernet 1/3
 port-name "lag-client-1:1/2" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
 metro-ring-1
  ring-interfaces ethe 1/1 ethe 1/5
  enable
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname Switch-1
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MRP-from-Master
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster MRPRing 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl MRPRing ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl MRPRing
 deploy
 client client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

MCT-capable-switch-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-1:1/1" ethernet 1/1
 port-name "ICL-to-Switch-1:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/3" ethernet 1/3
 port-name "lag-client-1:1/4" ethernet 1/4
```

```
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
 metro-ring-1
  ring-interfaces ethe 1/1 ethe 1/5
  enable
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname Switch-2
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MRP-to-Master
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster MRPRing 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl MRPRing ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl MRPRing
 deploy
 client client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

client-Switch:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/4
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-1:1/3" ethernet 1/1
 port-name "ICL-to-Switch-1:1/4" ethernet 1/2
 port-name "ICL-to-Switch-2:1/3" ethernet 1/1
 port-name "ICL-to-Switch-2:1/4" ethernet 1/2
!
interface ethernet 1/1
 enable
!
end
```

Configuring sync CCEP early LACP delay

In an Active-Passive MCT cluster configuration, when an CCEP port that was in the Down state on the MCT active peer comes back up, the **client-interfaces sync_ccep_early lacp-delay** command ensures that LACP on the MCT active peer's CCEP LAG stays in the LACP-BLOCKED state for the interval configured by the command. This time-delay interval provides additional time to the passive MCT peer to process the REMOTE-UP event. This ensures the correct handling of the BUM packets that are received in the interim between the port being enabled and the REMOTE-UP event being processed.

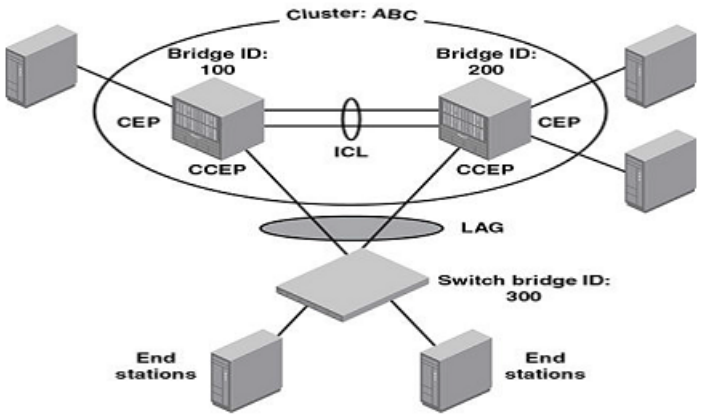
To configure the time delay, enter the **client-interfaces sync_ccep_early lacp-delay** command as follows:

```
device(config-cluster-MCT)# client-interfaces sync_ccep_early lacp-delay 5
```

In this example, the time delay is set to five seconds.

The following network diagram illustrates a cluster on an MCT node.

FIGURE 147 Sample network diagram of a cluster on an MCT node



Using this diagram, the following example is its configuration that includes the **client-interfaces sync_ccep_early lacp-delay** command.

```
cluster "ABC" 1
 rbridge-id 100
 session-vlan 99
 icl l2icl ethernet 2/6
 peer 172.16.10.2 rbridge-id 200 icl l2icl
 l2vpn-peer 1.1.1.2 rbridge-id 200
 client-interfaces sync_ccep_early lacp-delay 5
 deploy
 client "switch bridge"
  rbridge-id 300
  client-interface ethernet 3/13
  deploy
```

The following example shows the **show cluster config** output after the **client-interfaces sync_ccep_early lacp-delay** command has been configured.

```
device(config-cluster-MCT)# show cluster config
cluster "ABC" 1
 rbridge-id 100
 session-vlan 99
 icl l2icl ethernet 2/6
 peer 172.16.10.2 rbridge-id 200 icl l2icl
 l2vpn-peer 1.1.1.2 rbridge-id 200
 client-interfaces sync_ccep_early lacp-delay 5
```

```
deploy
client "switch bridge"
 rbridge-id 300
 client-interface ethernet 3/13
 deploy
```

The following example shows the **show cluster** output after the **client-interfaces sync_ccep_early lacp-delay** command has been configured.

```
device(config)#show cluster
Cluster mct 1
=============
Rbridge Id: 100, Session Vlan: 99
Cluster State: Deploy
Early sync of CCEP-UP info to MCT node enabled
lacp delay configured 5
Client Isolation Mode: Loose
Configured Member Vlan Range: 10
Active Member Vlan Range: 10
Total Clients Configured : 1 ( Deployed Clients: 1)

 ICL Info:
 ---------
 Name                                Port  Trunk
 icl                                  2/6   2

 Peer Info:
 ----------
 Peer IP: 1.1.1.2, Peer Rbridge Id: 200, ICL: icl
 KeepAlive Interval: 30 , Hold Time: 90, Fast Failover
 Active Vlan Range:  10
 Peer State: CCP Up (Up Time:   0 days: 0 hr:19 min:12 sec)

 Client Info:
 ------------
 Name               Rbridge-id    Config     Port  Trunk FSM-State
 switch bridge        10           Deployed   3/13    3    Up
```

MCT for VRRP or VRRP-E

One MCT switch is the VRRP or VRRP-E master routerand the other MCT switch is VRRP or VRRP-Ebackup router

The MCT switch that acts as backup router needs to ensure that packets sent to a VRRP-E virtual IP address can be L2 switched to the VRRP-E master router for forwarding. The MCT switch that acts as master router will sync the VRRP-E MAC to the other MCT switch that acts as backup router. Both data traffic and VRRP-E control traffic travel through the ICL unless the short-path forwarding feature is enabled.

L3 traffic forwarding from CEP ports to CCEP ports

Traffic destined to the CCEP ports from the client or CEP ports follow the normal IP routing on both master and backup routers. By default, the best route should not involve the ICL link. Only when the direct link from CEP ports to CCEP ports are down will the traffic be re-routed to pass through ICL link.

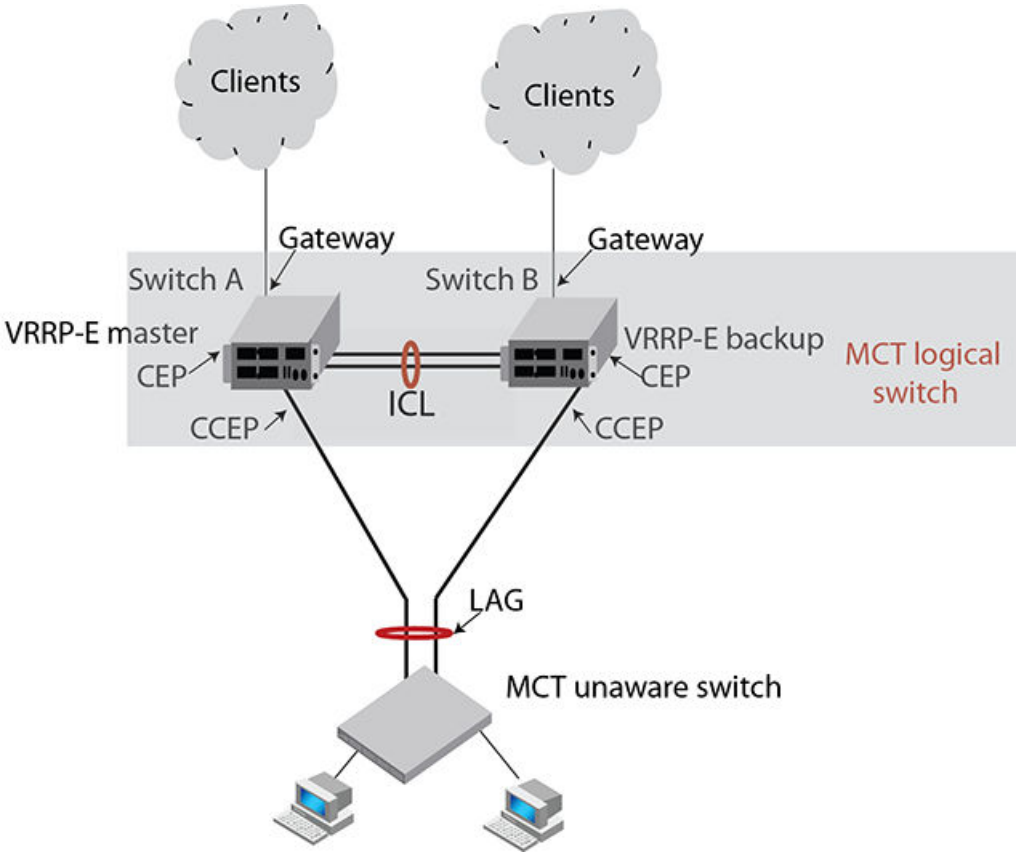
ARP broadcast resolution

Assuming that switch A is VRRP-E master router and switch B is the backup router. ARP request (a broadcast packet) from S1 that is sent through direct link to switch B will be sent to switch A for processing through ICL link. Since MAC learning is disabled on ICL link, the ARP will not be learned automatically through the ICL link. When the ARP request is received by switch A, the reply will be sent through direct link from switch A to S1. If by the time the ARP reply was received the MAC address for the MCT on S1 is not learned yet, the reply packet may be flooded to both the CCEP ports and ICL ports.

Both MCT switches are VRRP or VRRP-E backup routers

In the following figure, both MCT switches A and B need to ensure packets sent to VRRP-E virtual IP address can be L2 switched to the VRRP-E master router for forwarding. The MCT switch that has direct connection to the master router (who actually learned the VRRP-E MAC from the master) will sync the VRRP-E MAC to the other MCT switch that does not have direct connection to the master. Both data traffic and VRRP-E control traffic travel through ICL unless the short-path forwarding feature is enabled.

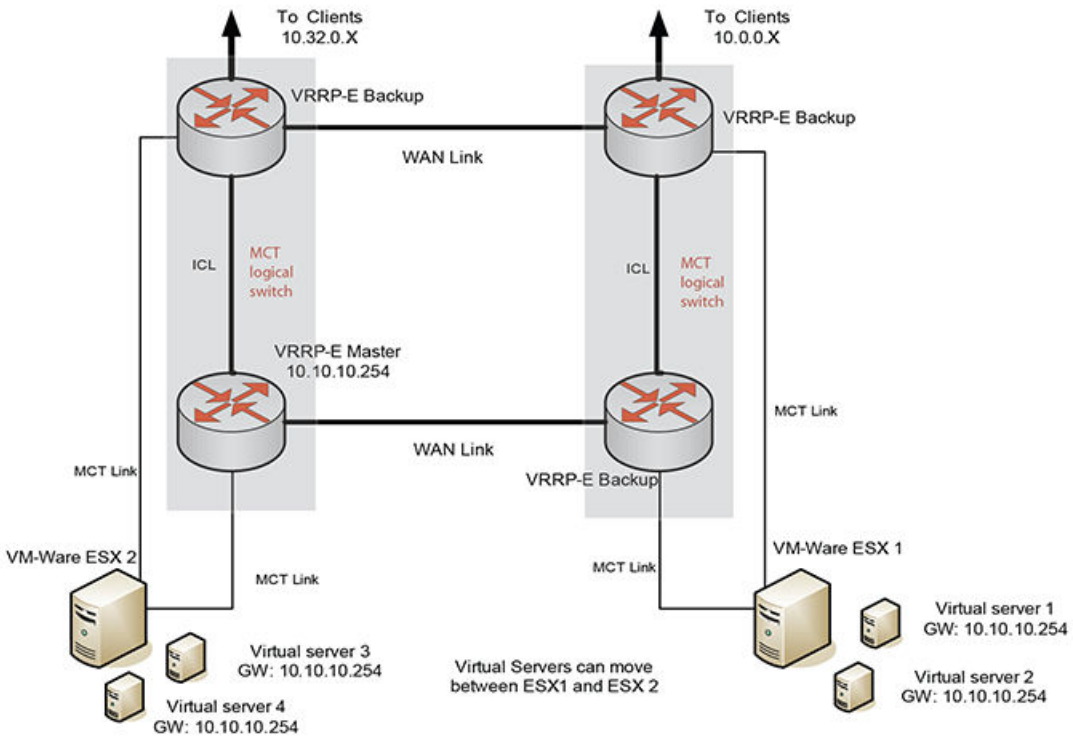
FIGURE 148 Example of MCTS that are Layer 2 switched



In the following figure, MCTs are deployed on two sites that are connected through two WAN links.

- Two WAN links are completely independent. Switch A and B form MCT 1 and switch C and D form MCT 2. There are L2 protocols running on the VRRP-E routers. L2 protocols will block one of the WAN links to ensure loop-free topology.

FIGURE 149 Example of MCTs that are deployed on two sites that are connected through two WAN links.



The following configurations are provided for the example shown in the previous figure.

NOTE
You must configure some Layer 2 protocols, such as STP and RSTP, on the VLAN to avoid a loop in this topology.

- Switch-A and Switch-B are MCT peers.
- Switch-C and Switch-D are MCT peers.

Switch-A configuration

```
lag "1" dynamic id 1
ports ethernet 1/1 to 1/2
primary-port 1/1
deploy
port-name "ICL-to-Switch-B:1/1" ethernet 1/1
port-name "ICL-to-Switch-B:1/2" ethernet 1/2
!
lag "2" dynamic id 2
ports ethernet 1/3
primary-port 1/3
deploy
port-name "lag-client-Esx1:1" ethernet 1/3
!
vlan 1 name DEFAULT-VLAN
no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
untagged ethe 1/3
tagged ethe 1/1 to 1/2 eth 1/4
router-interface ve 2
!
vlan 4090 name Session-VLAN
tagged ethe 1/1 to 1/2
router-interface ve 100
!
interface eth 1/4
port-name "Wan-1" ethernet 1/4

hostname Switch-A
!
interface ve 100
ip address 1.1.1.1/24
!
```

Cluster configuration:

```
cluster Switch-A 1
rbridge-id 1
session-vlan 4090
member-vlan 2
icl Switch-A ethernet 1/1
peer 1.1.1.2 rbridge-id 2 icl Switch-A
deploy
client Server-1
rbridge-id 100
client-interface ethernet 1/3
deploy
!
```

VRRPE configuration:

```
router vrrp-extended
interface ve 2
ip address 10.10.10.250/24
vrrp-extended vrid 10
ip-address 10.10.10.254
short-path-forwarding
```

Switch-B configuration

```
lag "1" dynamic id 1
ports ethernet 1/1 to 1/2
primary-port 1/1
deploy
port-name "ICL-to-Switch-A:1/1" ethernet 1/1
port-name "ICL-to-Switch-A:1/2" ethernet 1/2
!
lag "2" dynamic id 2
ports ethernet 1/3
primary-port 1/3
deploy
port-name "lag-client-Esx1:2" ethernet 1/3
!
!
vlan 1 name DEFAULT-VLAN
no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
untagged ethe 1/3
tagged ethe 1/1 to 1/2 eth 1/4
!
vlan 4090 name Session-VLAN
tagged ethe 1/1 to 1/2
router-interface ve 100
!
interface eth 1/4
port-name "Wan-2" ethernet 1/4

hostname Switch-B
!
interface ve 100
ip address 1.1.1.2/24
!
```

Cluster configuration:

```
cluster Switch-B 1
rbridge-id 2
session-vlan 4090
member-vlan 2
icl Switch-B ethernet 1/1
peer 1.1.1.1 rbridge-id 1 icl Switch-B
deploy
client Server-1
rbridge-id 100
client-interface ethernet 1/3
deploy
!
```

VRRPE configuration:

```
router vrrp-extended
interface ve 2
ip address 10.10.10.251/24
vrrp-extended vrid 10
ip-address 10.10.10.254
short-path-forwarding
end
```

Switch-C configuration

```
lag "1" dynamic id 1
ports ethernet 1/1 to 1/2
primary-port 1/1
deploy
port-name "ICL-to-Switch-D:1/1" ethernet 1/1
port-name "ICL-to-Switch-D:1/2" ethernet 1/2
!
lag "2" dynamic id 2
ports ethernet 1/3
primary-port 1/3
deploy
port-name "lag-client-Esx2:1" ethernet 1/3
!
no route-only
!
vlan 1 name DEFAULT-VLAN
no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
untagged ethe 1/3
tagged ethe 1/1 to 1/2 eth 1/4
!
vlan 4090 name Session-VLAN
tagged ethe 1/1 to 1/2
router-interface ve 100
!
interface eth 1/4
port-name "Wan-1" ethernet 1/4

hostname Switch-C
!
interface ve 100
ip address 1.1.1.3/24
!!
```

Cluster configuration:

```
cluster Switch-C 2
rbridge-id 3
session-vlan 4090
member-vlan 2
icl Switch ethernet 1/1
peer 1.1.1.4 rbridge-id 4 icl Switch
deploy
client Server-2
rbridge-id 1
client-interface ethernet 1/3
deploy
!
```

VRRPE configuration:

```
router vrrp-extended
interface ve 2
ip address 10.10.10.252/24
vrrp-extended vrid 10
ip-address 10.10.10.254
short-path-forwarding
end
```

Switch-D configuration

```
lag "1" dynamic id 1
ports ethernet 1/1 to 1/2
primary-port 1/1
deploy
port-name "ICL-to-Switch-C:1/1" ethernet 1/1
port-name "ICL-to-Switch-C:1/2" ethernet 1/2
!
lag "2" dynamic id 2
ports ethernet 1/3
primary-port 1/3
deploy
port-name "lag-client-Esx2:2" ethernet 1/3
!
no route-only
!
vlan 1 name DEFAULT-VLAN
no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
untagged ethe 1/3 to 1/5
tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
tagged ethe 1/1 to 1/2
router-interface ve 100
!
hostname Switch-D
!
interface ve 100
ip address 1.1.1.4/24
!!
```

Cluster configuration:

```
cluster Switch-D 2
rbridge-id 4
session-vlan 4090
member-vlan 2
icl Switch ethernet 1/1
peer 1.1.1.3 rbridge-id 3 icl Switch
deploy
client Server-2
rbridge-id 1
client-interface ethernet 1/3
deploy
!
```

VRRPE configuration:

```
router vrrp-extended
interface ve 2
ip address 10.10.10.253/24
vrrp-extended vrid 10
ip-address 10.10.10.254
short-path-forwarding
end
```

Configuration considerations

- VRRE-E virtual MAC will be synced and learned on ICL ports on backup routers through the ICL.
- VRRP or VRRP-E master router will be broadcast hello packets to all VLAN member ports including ICL ports. Normal VLAN FID will be used for broadcasting.
- VRRP or VRRP-E backup routers will not flood back hello packets received from ICL ports to ICL ports, but will be flooded to other non- ICL ports.

- In the current release, MCT switches must have complete routing information using static routes for L3 forwarding.
- For MCT switches configured with VRRP or VRRP-E, track-port features can be enabled to track the link status to the core switches so the VRRP or VRRP-E failover can be triggered.

  NOTE
  Extreme recommends disabling ICMP redirect globally to avoid unintended CPU forwarding of traffic when VRRP or VRRP-E is configured.

L3 traffic forwarding behaviors

When one MCT switch act as VRRP or VRRP-E master router and the other MCT switch is VRRP or VRRP-E backup, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be L2 switched to the VRRP-E master router for forwarding.
- The VRRP-E MAC will be learned by the other MCT switch that acts as backup router.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

When both MCT devices act as the VRRP or VRRP-E backup routers, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be L2 switched to the VRRP-E master router for forwarding.
- VRRP-E MAC will be learned by both MCT switches acting as backup routers.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

VRRP-E short-path forwarding and revertible option

The **track-port** command will monitor the status of the outgoing port on the backup. It will revert back to standard behavior (no short-path forwarding) temporarily even if short-path forwarding is configured.

Under the VRRP-E VRID configuration level, use the **short-path-forwarding** command. If the revertible option is not enabled, the default behavior will remain the same. Use the following command to enable short path forwarding.

```
device(config-if-e1000-vrid-2)#short-path-forwarding revert-priority
 60
```

Syntax: [no] short-path-forwarding [**revert-priority** *value*]

Use the supplied priority value as a threshold to determine if the **short-path-forwarding** behavior should be effective or not. If one or more ports tracked by the **track-port** command go down, the current priority of VRRP-E will be lowered by a specific amount configured in the **track-port** command for each port that goes down.

Once the current-priority is lower than the threshold, the **short-path-forwarding** will be temporally suspended and revert back to the regular VRRP-E forwarding behavior without **short-path-forwarding** enabled.

The reverting behavior is only temporary. If one or more of the already down ports tracked by the **track-port** command come back, it is possible that the current priority of VRRP-E will be higher than the threshold again and the **short-path-forwarding** behavior will be resumed.

IPv6 VRRP-E short-path forwarding and revertible option

Short-path forwarding enables the short path forwarding on an IPV6 VRRP-E device. It will revert back to standard behavior (no short-path forwarding) temporarily even if short-path forwarding is configured.

Configuration considerations

- VRRP-E virtual MAC will be synced and learned on ICL ports on backup routers through the ICL.
- ICL ports must be member ports of VLANs that CCEP ports are members of.
- VRRP or VRRP-E master router will be broadcast hello packets to all VLAN member ports including ICL ports. Normal VLAN FID will be used for broadcasting.
- VRRP or VRRP-E backup routers will not flood back hello packets received from ICL ports to ICL ports, but will be flooded to other non- ICL ports.
- MCT switches must have complete routing information using static routes for L3 forwarding.
- For MCT switches configured with VRRP or VRRP-E, track-port features can be enabled to track the link status to the core switches so the VRRP or VRRP-E failover can be triggered.

  NOTE
  Extreme recommends disabling ICMP redirect globally to avoid unintended CPU forwarding of traffic when VRRP or VRRP-E is configured.

L3 traffic forwarding behaviors

When one MCT switch act as VRRP or VRRP-E master router and the other MCT switch is VRRP or VRRP-E backup, the following behavior will be seen:

- Packets sent to VRRP-E virtual IPv6 address will be L2 switched to the VRRP-E master router for forwarding.
- The VRRP-E MAC will be learned by the other MCT switch that acts as backup router.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

When both MCT devices act as the VRRP or VRRP-E backup routers, the following behavior will be seen:

- Packets sent to VRRP-E virtual IPv6 address will be L2 switched to the VRRP-E master router for forwarding.
- VRRP-E MAC will be learned by both MCT switches acting as backup routers.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

Under the IPv6 VRRP-E VRID configuration level, use the **short-path-forwarding** command. If the revertible option is not enabled, short path forwarding will be disabled if the VRRP-E router priority is below the revert-priority configured value.Use the following command to enable short path forwarding.

device(config-if-e1000-vrid-2)# short-path-forwarding revert-priority 60

Syntax: [**no**] **short-path-forwarding** [**revert-priority***value*]

Use the supplied priority value as a threshold to determine if the short-path-forwarding behavior should be effective or not. If one or more ports tracked by the track-port command go down, the current priority of IPv6 VRRP-E will be lowered by a specific amount configured in the track-port command for each port that goes down.

Once the current-priority is lower than the threshold, the short-path-forwarding will be temporally suspended and revert back to the regular VRRP-E forwarding behavior without short-path-forwarding enabled.

The reverting behavior is only temporary. If one or more of the already down ports tracked by the track-port command come back, it is possible that the current priority of VRRP-E will be higher than the threshold again and the short-path-forwarding behavior will be resumed.

IPv6 VRRP-E short-path forwarding delay

Use IPv6 VRRP-e short-path forwarding delay to configure the time delay required to enable short path forwarding after reloading the backup router. When configured, short path forwarding will be enabled only after the configured delay time after the MP initialization is completed (from the time all modules in the system are UP). Default value is set to 0 seconds.

This is global IPv6 VRRP-E configuration will effect all IPv6 VRRP-E instances.

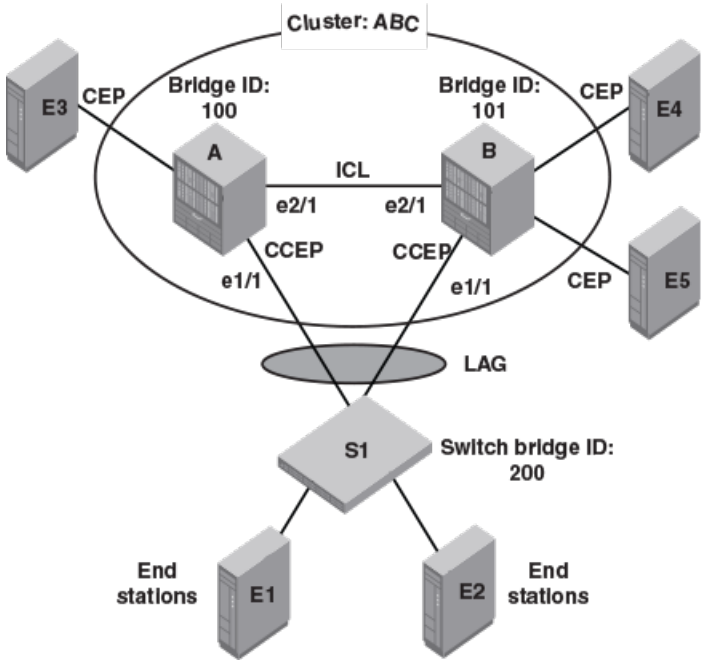
```
device(config)# [no] short-path-forwarding-delay 100
```

Syntax: **short-path-forwarding-delay** *seconds*

Sample configurations

```
device(config)#short-path-forwarding-delay 100
device(config)#ipv6 router vrrp-extended
device(config-ipv6-vrrpe-router)#interface ve 10
device(config-vif-10)# ipv6 address 2003::10:11/64
device(config-vif-10)#ipv6 vrrp-extended vrid 10
device(config-vif-10-ipv6-vrid-10)#backup priority 50
device(config-vif-10-ipv6-vrid-10)#ipv6-address 2003::11:50
device(config-vif-10-ipv6-vrid-10)#short-path-forwarding revert-priority 120
```

Sample MCT Configuration



Switch A:

```
vlan 4090
 tagged ethe 2/1
 router-interface ve 1
!
```

interface ve 1

```
 ip address 192.168.1.1/24
!
```

cluster ABC

```
 rbridge-id 100
 session-vlan 4090
 member-vlan 100 to 300
 icl icl_a_b ethernet 2/1
 peer 10.10.20.2 rbridge-id 101 icl icl_a_b
 deploy
 client switch_s1
  rbridge-id 200
  client-interface ethernet 1/1
  deploy
  exit
!
```

IPv6 VRRP Configuration

```
vlan 200
 tagged ethe 1/1 ethe 2/1
 router-interface ve 10
!
```

Ipv6 router vrrp

```
interface ve 10
 ipv6 address 10::1/64
 ipv6 vrrp vrid 10
  backup priority 50
  ipv6-address 10::100
  activate
!
```

Switch B:

```
vlan 4090
 tagged ethe 2/1
 router-interface ve 1
!
```

interface ve 1

```
ip address 192.168.1.2/24
!
```

cluster ABC

```
rbridge-id 101
 session-vlan 4090
 member-vlan 100 to 300
 icl icl_a_b ethernet 2/1
peer 10.10.20.1 rbridge-id 100 icl icl_a_b
 deploy
 client switch_s1
  rbridge-id 200
  client-interface ethernet 1/1
  deploy
  exit
!
```

IPv6 VRRP Configuration

```
vlan 200
 tagged ethe 1/1 ethe 2/1
```

```
 router-interface ve 10
!
```

Ipv6 router vrrp

```
interface ve 10
 ipv6 address 10::2/64
 ipv6 vrrp vrid 10
  backup priority 50
  ipv6-address 10::100
  activate
!
```

NOTE

Cluster client-rbridge-id on both switch A and B have to be same value for a given MCT.

Switch S1:

```
lag "mct_s1" static id 1
 ports ethernet 7/1 to 7/2
 primary-port 7/1
 deploy
!
vlan 200
 tagged ethe 7/1
 router-interface ve 10
!
interface ve 10
 ipv6 address 10::99/64
```

L2VPN support for L2 MCT clusters

For a L2VPN MCT, L2 MCT peer configuration is not required as it operates independently.

- L2VPN MCT does not require direct ICL as L2VPN may use no-direct MPLS network to the peer MCT node.
- L2VPN MCT does not require L2's session VLAN (and have peer in the same subnet as peer) as peer need not be directly connected.
- L2VPN and L2 MCT can be supported simultaneously
- When L2VPN and L2 needs to be run concurrently, you must configure the L2 peer parameters (similar to MCT/L2) and also configure L2VPN peer as well. The L2 peer will be used for the MCT communication.
- When using the L2VPN service, you must configure a L2VPN peer as well.
- MCT L2VPN is HLOS is not supported but compatible.

Support for non-direct ICL

L2VPN MCT functionality supports non-direct ICL functionality apart from the existing direct ICL requirement that is needed for L2.

A L2VPN MCT session needs to be established or verified at run time so that L2VPN traffic can traverse the ICL path.

The **l2vpn-peer** command needs to be configured to support L2VPN over MCT

A l2vpn peer address is required to be on a remote MCT node loopback address on which the L2VPN sessions is formed.

- This configuration is not mandated by configuration but is required for MCT operation to support L2VPN services.
- A cluster L2 peer address configuration is not required to support L2VPN over MCT.

NOTE
For MCT L2VPN services to function, you must configure an operational MPLS tunnel (RSVP or LDP) between the MCT peers.

L2VPN timers

Following optional timers are needed when using MCT L2VPN.

Keep-alive timer : This is used to detect a CCP that is down. The default is 300 milliseconds with hold time of 900 milliseconds. This timer cannot be changed dynamically once the cluster is deployed.

Node-Keep-Alive timer: This is used to detect whether the peer MCT node is down. The default is 2 seconds with hold time of 6 seconds. This timer cannot be changed dynamically once the cluster is deployed. This timer is to quickly detect CCP communication between the MCT peers and to failover quickly. This failure can happen due to route flaps or congestion in the network or in the MCT peer nodes. (Compared to L2 or MCT peer, this is similar to ICL link down).

NOTE
In CER 2000 Series and CES 2000 Series devices, the hold time should be configured to a minimum value of 1800 milliseconds to avoid CCP flaps when the **no client-interface shutdown** command is run. For example: `l2vpn-peer 10.6.240.3 timers keep-alive 600 hold-time 1800`

Cluster CCP session rules

NOTE
CCP (Cluster Communication Protocol) can run either on L2VPN peer (in cases where only l2vpn peer configured) or on L2 configured peer (in cases where L2 and L2VPN services need to coexist).

NOTE
If using a L2VPN peer in conjunction with L2 Peer, the L2VPN keepalive timers are not used and we will rely on MCT L2 peer for MCT communication. Additionally, for L2VPN peer case, these timers need to match on both nodes. If not, the CCP using L2VPN will not come up.

L2VPN peer only configurations

The CCP comes up only if:

- CCP session to L2VPN peer has to come up using regular TCP session.

The CCP goes down if either of the below conditions is TRUE:

- L2VPN CCP is down (TCP shuts down the CCP session for some reason).2VPN-CCP-timer expires.

The remote node down event is generated if both scenarios below are TRUE:

- L2VPN-CCP keepalive timer expires. This will indicate route failure (300 milliseconds * 3 ~ 900 milliseconds).
- L2VPN-node keepalive timer expires. This timer is to detect the reachability of the remote node. (Compared to MCT or L2 peer, this is similar to remote MCT node down or not reachable).

  NOTE
  It does not need to be route failure for the timer to expire.

- This timer is to quickly detect CCP communication between the MCT peers and to failover quickly. This failure can happen due to route flaps or congestion in the network or in the MCT peer nodes. (Compared to L2 or MCT peer, this is similar to ICL link down).

L2VPN peer with a L2 peer

The CCP comes up if the following is TRUE:

- Peer normal CCP session comes up

The CCP goes down if the following is TRUE:

- L2 peer CCP session goes down

The remote node down event is generated if the following condition is TRUE:

- L2 CCP session is down and keepalive VLAN does not respond

  NOTE
  CCP is dependent only on L2 in such a configuration

  NOTE
  The behavior for L2 peer only case will be similar to the original L2/MCT implementation.

Handling L2VPN spoke down

For MCT services in L2VPN to function, you must establish a spoke pseudowire (PW) between two MCT peers for each L2VPN service instance. This spoke PW will be used for sending traffic from the standby MCT node to the remote PE devices and also in failover scenarios. The spoke PW is like any other PW session between two peers and will be established using LDP, and can use any MPLS tunnel transport mechanism (RSVP or LDP tunnel).

If the CCP is up and the L2VPN MCT spoke PW is down, the following actions occur for all L2VPN instances or some of the instances:

- For the L2VPN instances that are down, a list of MCT Cluster Client Edge Port (CCEP) clients is collected and triggers the MCT infrastructure to perform Master/Slave selection for those MCT clients.
  - For each MCT client, only the links connected to one of the MCT nodes will be up (Master) and the other MCT node will be down (Slave).
  - If MCT client ports are *not* shared by Layer 2 and L2VPN, the MCT spoke PW going down will not affect Layer 2.
  - If any MCT client ports are shared by Layer 2 and L2VPN, the MCT spoke PW going down will affect both Layer 2 and L2VPN.
- When the MCT spoke PW is down and comes up later, it will revert back to active-active MCT clients.

CCP down handling when both L2 and L2VPN exist

When no keep-alive VLAN is configured,

- Client links status will be controlled based on loose and strict configurations. Both links can stay up *or* both links can go down *or* one of them can stay up and one of them can stay down.
- For L2VPN, for both VLL and VPLS, both MCT nodes will take the Active role for signaling towards the remote PEs.

When L2 keep-alive VLAN is configured, the initial state will be same as if there is no keep-alive. Then once keep-alive probes are exchanged, only one of the client links will stay up.

Graceful restart support

Graceful switchover is handled for MCT/L2VPN by the following:

- Using the **client-interfaces shutdown** (cluster configuration) command.

- R1 sends graceful-upgrade-restart (MCT) message to R2 (where the MCT peer nodes is R1 and R2 and R1 needs to be upgraded) and R1 disables all client interfaces locally.
- L2VPN task brings down all the PWs to standby state
- R2 (MCT peer node) performs the following actions:
  - Process the graceful-upgrade-restart (MCT) message that is received via CCP. It will force local L2VPN instances to be forcefully active (though it does not match local configuration).
  - Send back graceful-upgrade-done (MCT) message to R1 to indicate the completion.
  - R1 receives the graceful-upgrade-done message from R2 and it generates a local syslog to indicate the user to continue with the reload/restart operation. User now can proceed with the upgrade.
  - This can minimize the traffic loss in cases where user wants to perform graceful restart operation on one of the MCT nodes.

Show commands

Use the **show cluster** command will display MCT cluster information.

```
device#show cluster
Cluster clu 1
=============
Rbridge Id: 4, Session Vlan: 0
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range:
Active Member Vlan Range:
show cluster clu client c1
Cluster clu 1
=============
Rbridge Id: 4, Session Vlan: 0
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range:
Active Member Vlan Range:
Client Info:
------------
L2VPN Peer Info:
---------------
Peer IP: 5.5.5.5, Peer Rbridge Id: 5
KeepAlive Interval: 300 , Hold Time: 900
Node KeepAlive Interval: 2000 , Hold Time: 6000
l2vpn-revertible-timer 300
Peer State: CCP Up (Up Time: 0 days: 0 hr:15 min:18 sec)
Client Info:
------------
Name Rbridge-id Config Port Trunk FSMState
c1 101 Deployed 1/4 2 Admin Up
c2 102 Deployed 1/2 1 Up
```

Syntax: show cluster

See Show commands on page 444 for additional information regarding the **show cluster** command output.

Sample Configurations

To support L2VPN services, the cluster with the newly added L2VPN parameters for this feature must be configured.

Below, the configuration highlighted is the new L2VPN configuration.

Assume: Local MCT node loopback 1.1.1.1; Peer MCT 2.2.2.2; Remote L2VPN Peers: 3.3.3.3, 4.4.4.4;

For L2 operation, the remote peer in direct ICL is assumed to be 10.10.10.2.

Cluster Configuration:

```
cluster mct_l2vpn 1
     rbridge-id 1
     [session-vlan 101]         // The below configuration is needed only for MCT/L2
     [icl interface eth 1/1 icl]
     [peer-address 10.10.10.2 rbridge-id <id>]
     l2vpn-peer 2.2.2.2 rbridge-id <id>
    [l2vpn-peer 2.2.2.2 timers keep-alive <msec> hold-time <msec>]
    [l2vpn-peer 2.2.2.2 timers node-keep-alive <sec> hold-time <sec>]
    deploy

    client MCT_CLIENT1
      rbridge-id 101

 client-interface e 1/1
     [vll-pw-redundancy-active]
      deploy
```

Sample MPLS VLL/VPLS Configuration:

```
router mpls
  vpls-policy
    [vpls-pw-redundancy-active]

  lsp MCT_PEER_LSP
    to 2.2.2.2
    enable
// Below L2VPN configuration is given just for completion
vll MCT_VLL1 101
   vll-peer 3.3.3.3 [4.4.4.4]
   [vll-pw-redundancy-active]

   vlan 101
     tagged eth 3/1
  vpls MCT_VPLS1 101
   vpls-peer 3.3.3.3  4.4.4.4 5.5.5.5  6.6.6.6
   [vpls-pw-redundancy-active]

   vlan 101
            tagged eth 3/1
```

MCT for VPLS

MCT helps organizations build scalable and resilient network infrastructures. MCT is an enhancement over the link aggregation standard, which allows multiple switches to appear as single logical switch connecting to another switch using a standard LAG. MCT is designed to achieve the desired active-active topology and efficient Layer 2 multipathing, while ensuring that the network scales effectively.

You can connect a customer edge device to an MCT, such as two MLX Series devices, and then run it over a VPLS network. This is ideal for inter-data center connectivity, or within a campus environment for extending across a backbone. One of the benefits of this is to simplify VM mobility and eliminate single point of failures. For any customers looking to inter-connect multiple data centers and offer new cloud based services, such as disaster recovery, MCT can help you achieve that. In a metro network, MCT helps providers offer their customers enhanced end to end resiliency for business services.

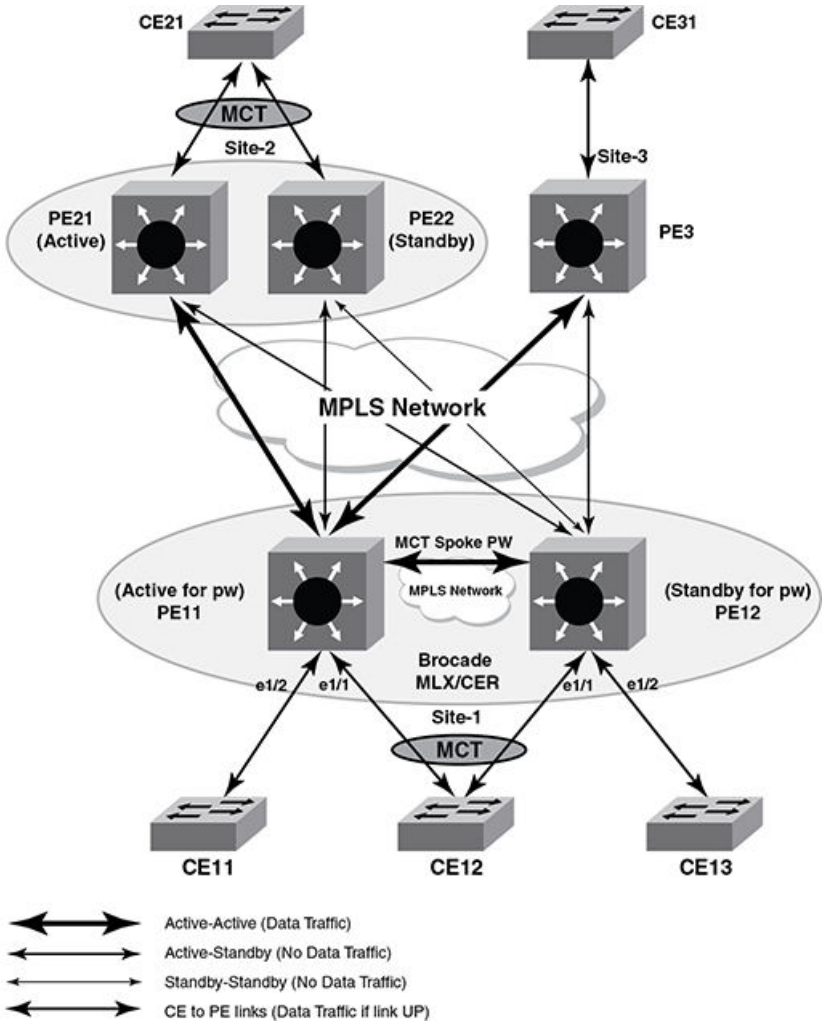
This feature supports dual-homing connectivity of CE devices to PE devices. Dual-homing enables link level and node level redundancy for CE's connected to PE's.

Figure 152 is a typical topology with MCT end-points for VPLS.

- PE11 and PE12 are the two nodes of the MCT Cluster.
- CE12 is connected to the two MCT nodes of the cluster using a LAG. From MCT nodes point of view, the links connected to CE12 are called CCEP end-points.

- CE11 and CE13 are single homed to PE11 and PE12 respectively. These are called the CEP end-points.

FIGURE 150 Sample topology with MCT end-points for VPLS



Configuration Considerations

Certain configurations must be correctly configured on both MCT nodes for MCT to work.

If the following are not configured correctly, then MCT functionality will not come up for that VPLS instance and both MCT nodes will independently come up as Active nodes.

- VPLS instance configuration should be with the same vc-id on both MCT nodes.
- Both MCT nodes should have "MCT" configured for the VPLS Instance.
- Both MCT nodes should have the same vc-mode (tagged or raw mode). If this configuration doesn't match, then the MCT Spoke PW between the two MCT nodes will not come up and MCT functionality will not work.
- Both **vpls-mtu** and **vpls-mtu-enforcement** configuration should be same on both MCT nodes. If this configuration does not match, then the MCT Spoke PW between the two MCT nodes may not come up and MCT functionality will not work.

The following configurations must be same on both MCT nodes for VPLS instance with MCT end-points.

- Same set of remote peers
- Local-switching enable or disable
- VPLS MAC table size

For each VPLS instance which has an MCT end-point, the MCT peer node is configured as a special VPLS peer. This is done on both MCT nodes. This is referred to as the cluster-peer and the PW between them is called the MCT Spoke PW.

One of the nodes from MCT pair is picked as active. This can be done either globally or controlled per VPLS instance.

Active MCT node will establish the PW session with the remote PEs and signal this active status in the status TLV.

MP switchover and Hitless OS upgrade is not supported but compatible.

CES Series and CER Series device limitations

Consider the following limitations for the CER 2000 Series and CES 2000 Series devices

- LSP of VPLS VC cannot run over ICL port.
- Spoke VC & other normal VC cannot share same port when local switching is disabled
- As the MCT VPLS uses twice the number of MACs in hardware, the **system max mac** shall be reduced to half when all the VPLS instances in the node are MCT enabled.

Scalability

The following table provides information about the scalability numbers for the MCT VPLS feature.

TABLE 49 MCT VPLS scalability

Capability	MLX Series	XMR Series
Maximum number of MCT VPLS instances in the core	4096	16,384
Maximum number of MCT L2VPN instances for MAC addresses	131,072	131,072
VLANs per VPLS or MCT	4096	4096
BUM rate limiting (VPLS CPU protection)	Enabled	Enabled
VPLS VLAN ports and VPLS peers	49,152	49,152

The following limitations impact the MCT VPLS scalability:

- VPLS CPU protection is limited due to the number of Mapped VLAN ID (MVID) resources. Currently, 2000 MVIDs are supported in the system and shared between multiple applications.
- VPLS supports Forwarding ID sharing (except on the BR-MLX-40Gx4-M 4-port, BR-MLX-4-port-10g-M-IPSEC-4-port, BR-MLX-10Gx20 20-port 1/10GbE, and BR-MLX-100Gx2-CFP2 modules).
- Dynamic Forwarding IDs are limited to 8000 in the system and shared between multiple applications.

Forwarding known unicast traffic

On Active MCT PE node, traffic will be forwarded and received from all the remote PE's.

For a host which is single homed to standby MCT PE node, the Active MCT PE node will send the packet on the MCT Spoke PW to standby MCT PE node.

On standby MCT PE node, traffic received from local CCEP and CEP ports will be forwarded to Active MCT PE node through the MCT Spoke PW which will then forward to the remote PE's.

On standby, the MCT PE node traffic received from all other PE nodes except the MCT Spoke PW is dropped.

Forwarding broadcast, unknown unicast, multicast traffic

Known unicast packets will be forwarded based on where the destination MAC is learned (Exception to Packets received on MCT Spoke PW and destined to CCEP) but for BUM traffic it will forwarded to all the destinations listed below.

On the active MCT PE Node

- Traffic received from CE's is forwarded to all Remote PE's, MCT Spoke PW and locally connected CE's.
- Traffic received from all Remote PE nodes (which doesn't include the MCT Spoke PW) is sent to all CE's and a copy is sent to MCT Spoke PW.
- Traffic received from the MCT Spoke PW, is sent to all the Remote PE's and locally connected CE's which are single-homed.

On Standby MCT PE Node

- Traffic received from CE's is forwarded to all locally connected CE's and to MCT Spoke PW which is the only PE session which is active.
- Traffic received from the MCT Spoke PW is forwarded to all locally connected CE's which are single homed to this standby MCT PE node.

MAC Learning and Synching

CCEP endpoints can send traffic to either of the MCT switches. This causes MAC addresses to move from CCEP port and the MCT Spoke PW continuously. To avoid MAC movement between the CCEP port and MCT spoke, MAC learning is disabled on MCT spoke PW and MAC addresses are synced between the two MCT peer switches using MAC database update protocol (MDUP).

MAC addresses learned are added to the MDUP database (MDB). MAC addresses learned locally on the cluster node are added to the local MDB with a cost of zero. MAC addresses learned from the peer cluster node are added to the remote MDB with a cost of one.

If a MAC address exists in both local and remote MDB, the MAC with the lowest cost is selected and added to the forwarding database (FDB). Cluster local MACs are always given preference over cluster remote MACs.

- On Active MCT PE Node MAC's are learned from the packets received from both CE and PE's (except the MCT Spoke PW).
- On standby MCT PE node MAC's are learned from packets received from CE's. (No packets are received from the remote PE's).
- There is no learning of MAC's for packets received on MCT Spoke PW. This is true on both Active and standby PE Node.
- For all VPLS instance which have MCT based endpoints, the complete MAC tables are synched between Master and Slave MCT peers.

MAC Aging

Cluster remote MDB entries are not aged locally. They are deleted only when an MDUP messages is received from the cluster peer switch.

MAC entries are deleted from FDB only if the MAC entry is not present in both local and remote MDBs.

Active-standby role change (revertible timer)

When a MCT node cannot establish a CCP connection with other MCT node, it will declare itself as Active and start the Remote Peering.

When the CCP session is established the node which is already Active will have higher preference over other MCT node if it is in Transit state.

MCT node which is configured to be Active will start the **l2vpn reversion-timer** to take over the Active role (if it is in standby role).

Local switching with MCT

With MCT-VPLS support, local-switching refers to traffic switched between end-points of both the MCT nodes of the cluster.

Identical **vpls-local-switching** configurations are required on the two MCT nodes for either supporting local-switching or disabling local-switching.

When local-switching is disabled, packets from the Active MCT node end-points are not sent to MCT-Spoke PW.

Packets from standby MCT node are always sent to MCT-Spoke PW.

CPU protection with MCT

CPU protection is supported with MCT VPLS.

CPU protection can be turned on independently on each MCT node of the cluster. When the MCT VPLS is enabled for a VPLS instance, it will use **cpu-protection** support with MCT.

Auto-discovery with MCT

VPLS auto-discovery is where the VPLS peer's are discovered by BGP and added into VPLS instead of manually configuring the peers.

If the MCT peer node is discovered as a VPLS peer by BGP, then it will not be added as VPLS remote peer

> NOTE
> This special handling is only for "MCT enabled VPLS Instances" and will not affect non-MCT VPLS instances.

Cluster-peer verses vpls-peer

Cluster-peer PW is the peering session between the two MCT nodes of the cluster. It is called MCT-Spoke PW.

- MCT-Spoke PW is different from the regular PW.
- MCT-Spoke PW bring up will be triggered only when the configuration sync between the two MCT nodes succeeds.
- MCT-Spoke PW signaling is triggered even though there are no local end points that are up or configured.
- MAC learning is disabled in software and hardware for the data traffic received on MCT-Spoke PW.

Graceful Restart and Upgrade

You can gracefully restart a node. If the node which is being brought down for maintenance has an "Active" role for any VPLS instance, that node will become the standby. This can be done using the **client-interface shutdown** command.

NOTE

- When the cluster is configured with **client-interface shutdown** command with only peer [L2] config, all clients on this MCT node are disabled except CCP. However, all data traffic will continue to be forwarded via the other MCT node. This is the known behavior before or after an upgrade.
- When the cluster is configured with **client-interface shutdown** command with l2vpn-peer config, the MCT-SPOKE-PW instance is disabled along with all clients on the specific MCT node. This is the known behavior before or after an upgrade. In this case, the other MCT node moves to MCT VPLS Active state, and all data traffic is forwarded from clients via this MCT node to the remote L2VPN peer.
- When the **client-interface shutdown** command is run to shut down the client interfaces, the MLX Series devices bring down the MCT Cluster Communication Protocol (CCP). However, the CER 2000 Series and CES 2000 Series devices do not bring down the CCP during the client interface shutdown.

PE to PE Forwarding

With the support of MCT end-point for VPLS, packets received from a remote PE are sent to cluster-peer (received from a PW and sent to another PW).

Similarly packets received from standby MCT node (using the cluster-cluster peer PW) are sent to Remote PE.

The received MPLS packet are de-capsulated and the L2 Payload is again encapsulated into MPLS packet with the right labels.

Unsupported features for MCT enabled VPLS instances

Following features are not supported for MCT enabled VPLS instances:

- 802.1ag
- IGMP-Snooping.
- VPLS-PBB

Configuring the MCT end-point for a VPLS instance

To enable MCT end-points for a VPLS instance, you must configure the **cluster-peer** for the VPLS Instance. This address should match the **l2vpn-peer** address that is done as part of the cluster configuration.

To configure a VPLS instance with a **cluster-peer** , there should not be any end-points or remote peers configured (auto-discovery should be disabled).

When the **cluster-peer** is configured, VPLS will enable the active or standby status TLV exchange with the remote peer's irrespective of other MCT configuration (like l2vpn-peer, cluster deployed or not deployed, CCEP end-points or no CCEP end-points for this VPLS instance).

If a l2vpn-peer configuration is already done and the cluster-peer configuration doesn't match with l2vpn-peer IP address, then configuration will be rejected.

To enable MCT functionality and to allow adding MCT and CCEP Ports as VPLS End-Points, enter a command such as the following.

```
device(config-mpls)#vpls test 10
deviceXMR4(config-mpls-vpls-test)# cluster-peer 12.12.12.12
```

Syntax: [no] cluster-peer *cluster-peer IPaddress*

The *cluster-peer IP address* parameter specifies the IP address of cluster peer.

The **no cluster-peer** command removes the cluster peer.

> NOTE
> Before removing the **cluster-peer** configuration for a VPLS instance using the command **no cluster-peer** , all the end points and remote peer configurations must be deleted.

Disabling cluster-peer mode for a VPLS instance error messages

If any end-point is configured while resetting the **cluster-peer** mode for the VPLS instance, the following error message will be displayed.

```
Error: End-point should not be configured while removing Cluster-Peer configuration.
```

If any remote peer is configured while resetting the cluster-peer mode for the VPLS instance, the following error message will be displayed.

```
Error: Remote-peer should not be configured while removing Cluster-Peer configuration.
```

If auto-discover is configured while resetting the cluster-peer mode for the VPLS instance, the following error message will be displayed.

```
Error: auto-discovery should not be configured while removing Cluster-Peer configuration.
```

VPLS global pw-redundancy (optional)

Once MCT is enabled for a VPLS instance, the two MCT cluster nodes synchronize the configuration with each other over the CCP and decide which node will take up the Active role and Standby role. PW redundancy provides is backup PWs ready so that traffic can be quickly failed over to the backup PWs. This command can be configured either globally for all VPLS Instances with MCT or for each VPLS instance individually.

> NOTE
> If it is not configured, the MCT node with lower rbridge-id will be elected as Active to signal to the remote PE's.

Use the **vpls-pw-redundancy-active** command at the global mode to set the pw-redundancy option for all VPLS Instances with MCT.

```
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)# vpls-pw-redundancy-active
```

Syntax: [no] vpls-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

Per VPLS instance pw-redundancy (optional)

If **vpls-pw-redundancy-active** is not configured per VPLS instance, the selection will be based on the global configuration. The per VPLS Instance configuration always has the higher priority over global configuration.

```
device(config-mpls)#vpls test 10
device(config-mpls-vpls-test)# vpls-pw-redundancy-active
```

Syntax: [no] vpls-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

Sample MCT configuration with VPLS endpoints

The following sample configuration shows the two MCT cluster nodes for the topology shown in MCT for VPLS on page 488.

```
Switch PE11:
MLX-PE11# show run
router mpls
 vpls test 10
  cluster-peer 12.12.12.12
  vpls-peer 21.21.21.21 22.22.22.22 3.3.3.3
  vlan 10
   tag eth 1/1 eth 1/2
……..
cluster abc 1
  rbridge-id 100
  l2vpn-peer 12.12.12.12 rbridge-id 101
  deploy
 client c1
  rbridge-id 300
  client-interface ethernet 1/1
  deploy

Switch PE12:
MLX-PE12# show run
router mpls
 vpls test 10
  cluster-peer 11.11.11.11
  vpls-peer 21.21.21.21 22.22.22.22 3.3.3.3
  vlan 10
   tag eth 1/1 eth 1/2
……..
cluster abc 1
  rbridge-id 101
  l2vpn-peer 11.11.11.11 rbridge-id 100
  deploy
 client c1
  rbridge-id 300
  client-interface ethernet 1/1
  deploy
```

VPLS show commands

```
device# show mpls vpls detail
VPLS test, Id 10, Max mac entries: 2048
Total vlans: 0, Tagged ports: 0 (0 Up), Untagged ports 0 (0 Up)
IFL-ID: n/a
VC-Mode: Raw
Total VPLS peers: 2 (2 Operational)
Cluster-Peer address: 12.12.12.12, State: Operational, Uptime: 2 hr 55 min
Tnnl in use: tnl2(3)[RSVP] Peer Index:0
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
Tnnl in use: tnl1(1024)[RSVP] Peer Index:1
Local VC lbl: 983041, Remote VC lbl: 983043
Local VC MTU: 1500, Remote VC MTU: 1500
Local PW preferential Status:Active, Remote PW preferential Status:Active
Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
CPU-Protection: ON, MVID: 0x001, VPLS FIDs: 0x0000a004, 0x0000ffff
Local Switching: Enabled
Extended Counter: ON
Multicast Snooping: Disabled
Cluster-peer: enabled, Role:Active State: VPLS_MCT_STATE_OPER
```

Syntax: show mpls vpls detail

Use the **show mpls vpls brief redundancy** command to display PW redundancy.

Syntax: show mpls vpls brief redundancy

TABLE 50 Output from the show mpls vpls brief redundancy **command**

Field	Description
Name	The configured name of the VPLS instance.
Id	The ID of this VPLS instance.
Ports Up	The number of ports in this VPLS instance that are up.
Num Peers	The number of VPLS peers this device has for this VPLS instance.
Peers Up	The number of VPLS peers with which a VC connection is completely operational.
MCT PW- Role	Active: Node will start peering with remote peers, signaling Status TLV as Active. Standby: Node will start peering with remote peers , signaling Status TLV as Standby Transit: MCT VPLS FSM is not in Operation state. Remote Peering is not yet enabled.

MCT for VLL

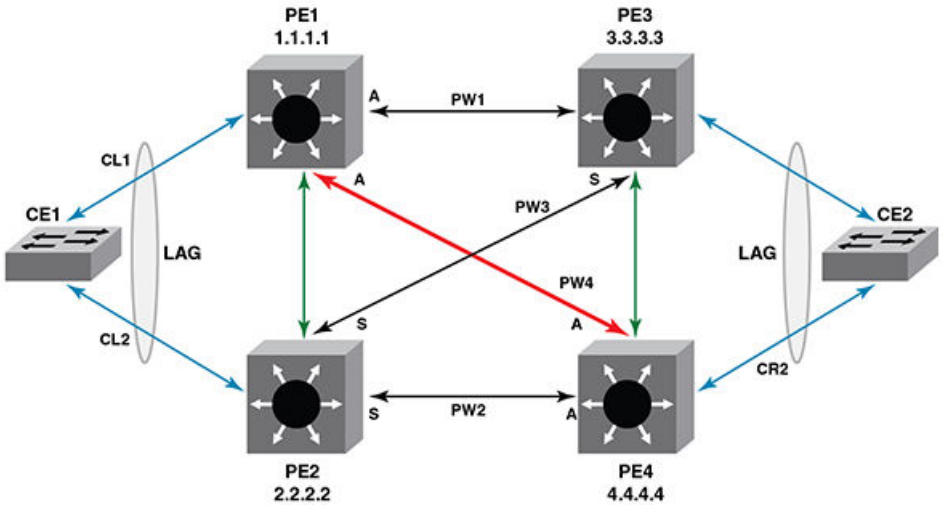
This feature supports dual-homing connectivity of CE devices to PE devices. Dual-homing enables link level and node level redundancy for CE's connected to PE's.

Figure 153 is a typical topology with MCT end-points for VLL. The VLL end-point connections from a single CE are dual homed to two PE nodes acting as an MCT cluster. From the CE, it has a LAG connection between CE and PE and is unaware of the MCT.

- PE1 and PE2 are two nodes of MCT cluster and work like a single PE. Similarly PE3 & PE4 are MCT cluster.
- CE1 is connected to PE1 and PE2 through a LAG.
- CE2 is connected to PE3 and PE2 through a LAG.
- PE1 has PW1 and PW3 vll PWs for same instance connected to remote MCT pairs.
- PE2 has PW2 and PW4 vll PWs for same instance connected to remote MCT pairs.
- For MCT pair PE1,PE2 PE1 is selected as active for a VLL instance, which it specifies as Active(A) in pw-redundancy status TLV during PW connection setup to remote PE pair.
- For MCT pair PE3 and PE4 nodes, PE4 is selected as active, which it specifies as Active(A) in pw-redundancy status TLV during PW connection setup to remote PE pair.
- There is a VLL instance mct-spoke-pw established between MCT pair for sending data traffic from standby to active node from the customer edge nodes(CE1, CE2).
- MCT cluster pair nodes communicate to each other using CCP communication provided by MCT infrastructure to identify as cluster pairs based on per instance and negotiate active/standby roles.
- For cluster pair PE1 and PE2, traffic from CE1 to CE2 flows as follows.
  - The traffic received directly from CE1 to PE1 is sent over the active PW PW3 to node PE4 which forwards it to client CE2.
  - The traffic received directly from CE1 to PE2 is sent over the MCT-SPOKE-PW to PE1. PE1 forwards this traffic over the active PW PW3 to node PE4 which forwards it to client CE2.

- For cluster pair PE1 and PE2, traffic from CE2 to CE1 flows as follows:
  - Traffic is received on the active PW for the pair PW3 on node PE1. Node PE1 forwards the traffic to CE1 directly over the local endpoint.
  - Traffic forwarded to PE2 in special case (where PE1 is Active but local CCEP is down)

FIGURE 151 Sample topology with MCT end-points for VLL



Configuration synchronization between MCT peers

MCT peers will exchange VLL information and updates.

VLL information sync

VLL addition and deletion information will be sent to MCT peer in the below scenarios.

- Whenever End Point (CCEP for MCT Client) and at least one VLL peer is configured, the VLL related information will sync to MCT peer. The MCT peer will add received VLL information to separate data structure (different from normal VLL structure).
- Whenever End Point (CCEP for MCT Client) or last VLL Peer are deleted then we will send message to MCT Peer to delete that VLL information.

Transparent forwarding of L2 and L3 protocols for CES and CER 2000 Series devices

Use the **forward-all-protocol** command to

- Add per port Layer 2 and Layer 3 (L2/L3) protocols ACL filters on the VLL end-point port (not on the MPLS normal interface).
- Add per port Layer 2 and Layer 3 protocols ACL filters on the normal L2 switching interface.

The command **no forward-all-protocol** removes the L2/L3 protocols ACL filters.

NOTE
The **forward-all-protocol** command is only applicable to the CER 2000 Series and CES 2000 Series devices.

To implement per port Layer 2 and Layer 3 (L2/L3) protocols ACL filters, enter a command similar to the following:

```
device(config)# int eth 1/1
device(config-if-e1000-1/1)# forward-all-protocol
```

Syntax: [no] forward-all-protocol

The command **no forward-all-protocol** deletes VLL endpoint port L2/L3 protocols ACL filters. For LAG, only the primary port needs to be configured.

NOTE
The **forward-all-protocol** command lets L2/L3 protocols on the port go with hardware forwarding without going to the CPU. If the **no forward-all-protocol** command is executed, the L2/L3 functions may be impacted.

The **show interfaces ethernet** *slot/port* command displays the configuration status of the **forward-all-protocol** command.

The following output example shows the **show interfaces ethernet** *slot/port* command with the **forward-all-protocol** command disabled.

```
device# show interfaces ethernet 1/1
GigabitEthernet1/1 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 001b.eda3.f841 (bia 001b.eda3.f841)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Member of 1 L2 VLAN(S) (tagged), port is in tagged mode, port state is Forwarding
  STP configured to ON, Priority is level0, flow control enabled
  Priority force disabled, Drop precedence level 0, Drop precedence force disabled
  dhcp-snooping-trust configured to OFF
  mirror disabled, monitor disabled
  LACP BPDU Forwarding:Disabled
  LLDP BPDU Forwarding:Disabled
  L2L3 protocols Forwarding:Disabled
  Not member of any active trunks
...
```

The following output example shows the **show interfaces ethernet** *slot/port* command with the **forward-all-protocol** command enabled.

```
device(config-if-e1000-1/1)# forward-all-protocol
device(config-if-e1000-1/1)# show interfaces ethernet 1/1
GigabitEthernet1/1 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 001b.eda3.f841 (bia 001b.eda3.f841)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Member of 1 L2 VLAN(S) (tagged), port is in tagged mode, port state is Forwarding
  STP configured to ON, Priority is level0, flow control enabled
  Priority force disabled, Drop precedence level 0, Drop precedence force disabled
  dhcp-snooping-trust configured to OFF
  mirror disabled, monitor disabled
  LACP BPDU Forwarding:Disabled
  LLDP BPDU Forwarding:Disabled
  L2L3 protocols Forwarding:Enabled
  Not member of any active trunks
...
```

The **forward-all-protocol** command forwards the following protocols by hardware instead of the CPU.

- For L2: UDLD (drop), FDP, CDP and MRP.
- For L3: IP broadcast (255.255.255.255), IP multicast ((224.0.0.x, 224.0.1.x) including RIP, OSPF, PIM, VRRP), ARP, DHCP, BOOTP, IS-IS, OSPF, ND6, RIPng, OSPFv3, PIMv6, anycast solicited node, DHCPv6.

NOTE
The **forward-all-protocol** command cannot be used on an interface running the above listed protocols because those protocol frames will not be processed by the CES/CER CPU.

Peer information sync

If VLL information is already sent to the MCT peer and VLL peer is updated, then the VLL peer information will sync to the MCT peer. VLL Peer information includes PW status and redundancy status.

End point status handling

Whenever there are any end point status changes, they are synchronized by MCT infrastructure and the events will be handled whenever logical status of endpoint changes from UP to DOWN or DOWN to UP.

End point mismatch

When end points configured in MCT nodes belongs to different MCT Clients, then the end point mismatch in both MCT nodes for that VLL and PWs for that VLL will be down.

Hitless upgrade

MCT VLL does not support Hitless upgrade, but is Hitless compatible.

Configuration Considerations

Extreme recommends making sure the VLL instance configuration is same on both nodes in the following aspects:

- Endpoint type(untagged or tagged or dual-tagged) and VLAN
- VC type(tagged-mode or raw-mode)
- Mtu and other operational parameters
- Peers configured for a given VLL instance

Configuring MCT VLL

MCT VLL requires following cluster level configurations. See Configuring the cluster operation mode on page 434 for additioan information on creating clusters.

L2VPN peer configuration

For each MCT VLL instance, a spoke-PW session is formed internally using the **l2vpn-peer** command to form a PW session to the remote MCT node. In this example, spoke-PW will be formed to peer 2.2.2.2.

```
device(config-mpls)#l2vpn-peer 2.2.2.2 rbridge-id 2
```

Syntax: [no] l2vpn-peer *ip-address* **rbridge-id** *id*

The *ip-address* variable specifies the IP address of the targeted peer.

The *id* parameters specify the remote bridge ID. Possible values are 1 - 35535 (16 bit value).

To disable the configuration, enter the **no** form of the command.

VLL global pw-redundancy (optional)

Once MCT is enabled for a VLL instance, the two MCT cluster nodes synchronize the configuration with each other over the CP and decide which node will take up the Active role and Standby role. PW redundancy support provides is backup PWs ready so that traffic can be quickly failed over to the backup PWs. This command can be configured either globally for all VLL instances with MCT or for each VLL instance individually.

> NOTE
> If it is not configured, the MCT node with lower rbridge-id will be elected as Active to signal to the remote PE's.

```
Use the vll-pw-redundancy-active
 command at the global mode to set the pw-redundancy option for all VLL Instances with MCT.
device(config-cluster-PE1-client-2)# rbridge-id 101
device(config-cluster-PE1-client-2)# client-interface ethernet 1/1
device(config-cluster-PE1-client-2)# vll-pw-redundancy-active
device(config-cluster-PE1-client-2)# deploy
```

Syntax: [no] vll-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

Per VLL instance pw-redundancy (optional)

If **vll-pw-reduncancy-active** is not configured per VLL instance, the selection will be based on the global configuration. The per VLL Instance configuration always has the higher priority over global configuration.

Vll-pw-redundancy-active option is used to load-balance VLL instances using client (for one client VLLs, one node is active and for other client, another MCT node can be active).

All VLL instances will not be active only on one node. This allows flexibility to provision VLL instances for each client differently.

When the client is deployed, all VLL instances that share the same MCT end-point(port e 1/1 in the example) will become MCT VLLs.

When the client configuration is un-deployed the VLL instance will be brought down and up without MCT.

The **vll-pw-redundancy-active** option if configured cannot be changed without un-deploying the client.

For VLL, two remote peers (remote MCT cluster pair) must be configured to support PW redundancy. In the configuration below, 3.3.3.3 and 4.4.4.4 are peers to reach remote node.

```
device(config-mpls)#vll test 10
device(config-mpls-vll-test)#vll-peer 3.3.3.3 [4.4.4.4]
device(config-mpls-vll-test)#vll-pw-redundancy-active
```

Syntax: [no] vll-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

If only one peer is specified, the PW redundancy status TLV with local status as active is supported.

Once the vll-instance is operational and if the vll level pwredundancy is changed, the pw-redundancy election is triggered, which can cause the vll-active state to change.

Setting the L2VPN global revertible timer

Whenever a node needs to become active for a given L2VPN instance, it sends a message to peer MCT node and starts this timer. Once this timer expires on configured active MCT node, it will move the L2VPN sessions to be active and the remote peer MCT node will move the sessions to be standby state for PW operation.

Use the **l2vpn-revertible-timer** command to start a revertible timer whenever a given L2VPN instance moves to different node as active node (against the configured active).

```
device(config-mpls)# l2vpn-revertible-timer 200
```

Syntax: [no] l2vpn-revertible-timer *sec*

Use the *sec* parameter to specify the amount to time before the L2VPN session becomes active. The default time is 300 seconds and can be a value from 0 throgh 65535 seconds. If the value is configured to be 0, then reversion will happen immediately.

> NOTE
> Immediate reversion can cause instability.

PW redundancy auto reversion timer option

If there is a transient condition where the standby node's pw is active and active node pw comes up, an auto-reversion timer of 60 seconds value(eg: 60) +/- 25% jitter seconds is started on the active node. Upon the auto reversion timer expiration, the remote pw on the active node changes the role to active and the remote pw on the standby node changes role to standby. This role change is coordinated between the MCT active and standby nodes.

Display commands

Use the **show mpls vll brief** and **show mpls detail** commands to display PW redundancy information for all VLLs and MCT related information.

```
device# show mpls vll brief
* - Active VLL Peer; U - UP; D - DOWN
                                                          Vll-Peer          Vll-Peer          MCT
Name      VC-ID       End-Point                           (State)           (State)           state
----      -----       ---------                           --------          --------          -----
vl            1        untag e 1/4(U)                      5.5.5.5(U)*       4.4.4.4(U)     Active
v2            2        tag vlan 201 e 1/4(U)               4.4.4.4(U)*       5.5.5.5(U)     Active
```

Syntax: show mpls vll brief

```
device# show mpls vll detail
VLL VLL1, VC-ID 1, VLL-INDEX 0
  End-point        : untagged   e 1/4
  End-Point state  : Up
  MCT state        : Active
  Local VC type    : tag
  Local VC MTU     : 4974
  COS              : --
  Extended Counters: Disabled
  Counter          : Disabled
  Vll-Peer         : 5.5.5.5 (Standby-Standby)
    State          : Down - no tummel LSP to vll-peer
    Remote VC type :                Remote VC MTU  :
    Local label    :                   Remote label   :
    Local group-id : 0                 Remote group-id:
    Tunnel LSP     : LSP_to_1 (tnl0)
    MCT Status TLV : Standby
  Vll-Peer         : 4.4.4.4 (Active-Active)
    State          : UP
    Remote VC type : tag               Remote VC MTU  : 1500
    Local label    : 798720            Remote label   : 798722
    Local group-id : 0                 Remote group-id: 0
    Tunnel LSP     : LSP4 (tnl2)
    MCT Status TLV : Active
  MCT Infomation   :
    Local CCEP state      : UP
    Remote CCEP state     : Down
```

```
    Pending reversion time: --
    Spoke PW          : 1.1.1.1
      State           : DOwn - endpoint is not UP (Reason: Election not done)
      Remote VC type  :                   Remote VC MTU  :
      Local label     :                   Remote label   :
      Local group-id  :                   Remote group-id: 0
      Tunnel LSP      : LSP_to_5 (tnl2)
    MCT Peer PWs      :
      VLL Peer            UP/DOWN  Active/Standby
      --------            -------  --------------
      1.1.1.1             UP       Active-Standby
      10.10.10.10         UP       Active-Active
```

Syntax: show mpls vll detail

MCT VLL sample configuration

```
device# show running configuration
router mpls
  l2vpn-revertible-timer 300
  lsp MCT_PEER_LSP            /
    to 2.2.2.2
    enable
  vll MCT_VLL1 101
     vll-peer 3.3.3.3 [4.4.4.4]
     [vll-pw-redundancy-active]
     vlan 101
       tagged eth 1/1
cluster mct_l2vpn 1
     rbridge-id 1
     l2vpn-peer 2.2.2.2 rbridge-id 2
    deploy
    client MCT_CLIENT1
      rbridge-id 101
      client-interface e 1/1
         [vll-pw-redundancy-active]
      deploy
```

MCT Snooping

All VLAN IGMP and PIM snooping features are supported on the MCT links. The device supports both IPv4 and IPv6 snooping over MCT links.

Events Handling

CCEP down event to MCT

CCEP down event to MCT peer is generated only if,

- The CCEP port is replaced in the SG entry's OIF to ICL ports, if CCEP ports exists in the OIF list. The new periodic IGMP joins now coming through the remote CCEP ports will reach us as MDUP messages on the ICL ports. This will ensure that the ICL ports in the SG/WG OIF list and IGMPv3 DB are refreshed.

MCT remote CCEP down event

MCT remote CCEP down event is generated only if,

The CCEP ports of the MCT peer (remote CCEP ports) goes down. Two possible scenarios exist:

- SG entry with ICL as IIF already exists (because there was a local CEP receiver for that SG). In this case, a CCEP link (only if it exists in WG entry or IGMPv3 DB due to joins received) will be added to SG OIF list. If no joins were received from CCEP, then the CCEP will not be added to the OIF list, unless it is a flooding scenario.
- SG entry with ICL as IIF did not exist because there are only remote CCEP receivers and no local CEP receivers. In this case when remote CCEP goes down we'll start getting SG flows through ICL and we'll create SG and add local CCEP ports to the OIF (copied from WG entry).

MCT local CCEP up event

MCT local CCEP up event is generated only:

After the local CCEP comes up, two possible scenarios exist:

- IGMP joins may start coming through the local CCEP links or
- IGMP joins may continue to come through the remote CCEP link as before.

Case 1: You stop receiving MDUP message from the MCT peer through the ICL link and the ICL ports are eventually aged out from the WG/SG OIF list. Local CCEP ports are added to the OIF list.

Case 2: You will continue to get MDUP messages from MCT peer through the ICL link and the local CCEP is up and the local CCEP is added to the OIF. The ICL ports will eventually age out, if the MDUP join messages with remote CEP flag set were not received. The CCEP up event is sent to the MCT PEER.

MCT remote CCEP up event

CCEP up event from the MCT PEER is generated only if,

- IGMP joins may start coming through the remote CCEP links or
- IGMP joins may continue to come through the local CCEP link as before.

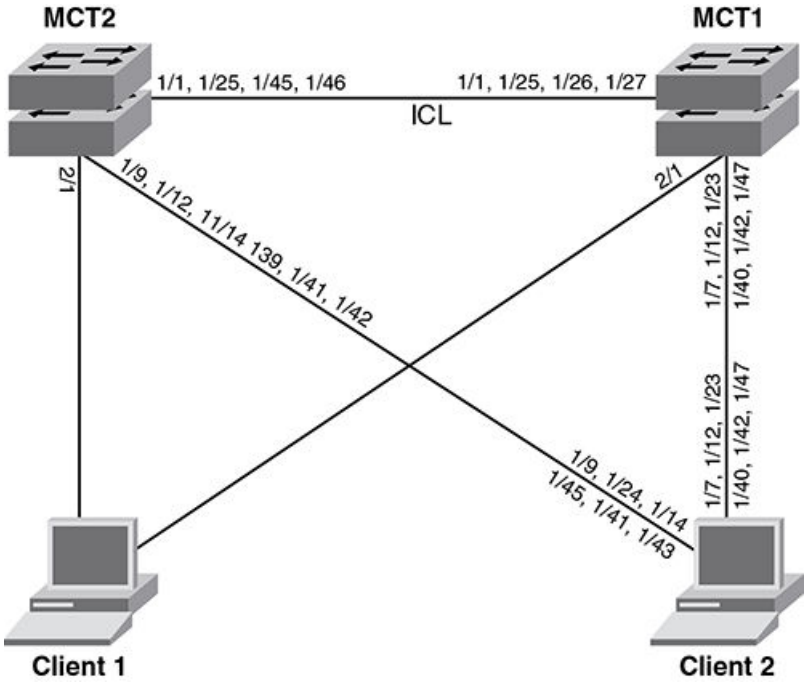
In both cases, upon receiving MCT CCEP up event from peer, the local CCEP ports will be removed from the OIF list of those SG entries with ICL as IIF. No other SG entries are affected. They'll continue to have the membership information from CCEP ports.

In the second case, it will continue to send MDUP join messages (with CCEP flag set) to the MCT peer, as usual. This will ensure that, in the MCT peer, those SG's sourced from its CEP ports will have its CCEP links as OIF (efficient path).

Configuration considerations

- IGMP and PIM proxy configurations need to be configured on both the MCT peers.
- Both MCT nodes should have same snooping mode (either both should be active or both should be passive).
- Static IGMP membership configurations for CCEP ports need to configured on both the MCT peers.
- Static uplink configurations need to be configured on both the MCT peers.
- MDUP does not support fragmentation of control packets over ICL.
- The IP MTU of any non-NetIron OS PIM router connected to a NetIron OS MCT PIM Snooping node needs to be configured to <=1450 bytes. This is due to the CCP payload size limit.

FIGURE 152 MCT snooping topology example



There are no specfic MCT commands for configuring MCT snooping. Below is a sample MCT snooping configuration .

Sample MCT snooping configuration

```
!
lag "CLIENT-1" dynamic id 6
ports ethernet 2/1
 primary-port 2/1
deploy
!
!
lag "CLIENT-2" static id 5
ports ethernet 1/7 ethernet 1/12 ethernet 1/23 ethernet 1/40 ethernet 1/42 ethernet 1/47
 primary-port 1/47
deploy
!
!
lag "ICL-1" static id 4
ports ethernet 1/1 ethernet 1/25 to 1/27
 primary-port 1/1
deploy
!
!
vlan 21
 untagged ethe 1/7 ethe 1/12 ethe 1/23 ethe 1/40 ethe 1/42 ethe 1/47 ethe 2/1
 tagged ethe 1/1 ethe 1/25 to 1/27
router-interface ve 21
multicast active
!
vlan 31
 tagged ethe 1/1 ethe 1/7 ethe 1/12 ethe 1/23 ethe 1/25 to 1/27 ethe 1/40 ethe 1/42 ethe 1/47 ethe 2/1
router-interface ve 31
multicast6 passive
!
!
vlan 4090
```

```
 tagged ethe 1/1 ethe 1/25 to 1/27
 router-interface ve 49
!
interface ve 21
ip address 21.1.1.1/24
!
interface ve 31
ip address 31.31.31.3/24
ipv6 address 2031:31::2/112
!
interface ve 49
ip address 49.49.49.1/30
!
!
cluster CLUSTER-1 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 20 to 50
icl ICL-1 ethernet 1/1
peer 49.49.49.2 rbridge-id 35535 icl ICL-1
deploy
 client CLIENT-1
  rbridge-id 3
  client-interface ethernet 2/1
  deploy
 client CLIENT-2
  rbridge-id 2
  client-interface ethernet 1/47
  deploy
!
Config on MCT2:-
=====================
!
lag "CLIENT-1" dynamic id 6
ports ethernet 2/1
 primary-port 2/1
deploy
!
lag "CLIENT-2" static id 5
ports ethernet 1/9 ethernet 1/12 ethernet 1/14 ethernet 1/39 ethernet 1/41 to 1/42
 primary-port 1/39
deploy
!
!
lag "ICL-1" static id 1
ports ethernet 1/1 ethernet 1/25 ethernet 1/45 to 1/46
 primary-port 1/1
deploy
!
!
vlan 21
 untagged ethe 1/9 ethe 1/12 ethe 1/14 ethe 1/39 ethe 1/41 to 1/42 ethe 2/1
 tagged ethe 1/1 ethe 1/25 ethe 1/45 to 1/46
router-interface ve 21
multicast active
!
!
vlan 31
 tagged ethe 1/1 ethe 1/9 ethe 1/12 ethe 1/14 ethe 1/25 ethe 1/39 ethe 1/41 to 1/42 ethe 1/45 to 1/46 ethe
2/1
 router-interface ve 31
multicast6 passive
!
!
vlan 4090
 tagged ethe 1/1 ethe 1/25 ethe 1/45 to 1/46
 router-interface ve 49
!
interface ve 21
ip address 21.1.1.2/24
!
interface ve 31
```

```
ip address 31.31.31.2/24
ipv6 address 2031:31::1/112
!
!
interface ve 49
ip address 49.49.49.2/30
!
!cluster CLUSTER-1 1
 rbridge-id 35535
 session-vlan 4090
 member-vlan 20 to 50
icl ICL-1 ethernet 1/1
peer 49.49.49.1 rbridge-id 1 icl ICL-1
deploy
 client CLIENT-1
  rbridge-id 3
  client-interface ethernet 2/1
  deploy
 client CLIENT-2
  rbridge-id 2
  client-interface ethernet 1/39
  deploy
```

Displaying IP multicast information

The following sections show how to display IP multicast information.

Displaying multicast information

The existing show commands have been modified to include the MCT specific information in the outgoing interfaces.

The example below displays if IGMP/MLD snooping is enabled on VLAN 20 and what mode of snooping is configured (active/passive).

Cluster peer #1:

```
device# show ip multicast vlan 20
----+-----+---------+---------------+-----+-----+------
VLAN State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+---------------+-----+-----+------
20   I-Ena Active    Self            40    1     1
----+-----+---------+---------------+-----+-----+------
Router ports:
Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join
  1    (*, 224.1.1.1 ) 00:09:04 NumOIF: 2 profile: none
          Outgoing Interfaces:
               CCEP rbr-id 200 vlan 20 ( V2) 00:00:22/22s
               ICL  vlan 20 ( V2) 00:07:37/16s
  1    (1.1.1.1, 224.1.1.1) in e2/1 vlan 20 00:00:03 NumOIF: 2 profile: none
          Outgoing Interfaces:
               ICL e2/15 vlan 20 ( V2) 00:00:03/0s
               CCEP rbr-id 200 e2/7 vlan 20 ( V2) 00:00:03/0s
          FID: 0x8006 MVID: None
```

Cluster Peer 2:

```
DUT2#show ip multicast vlan 20
----+-----+---------+---------------+-----+-----+------
VLAN State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+---------------+-----+-----+------
20   I-Ena Passive   None            28    1     1
----+-----+---------+---------------+-----+-----+------
Router ports:
Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join
  1    (*, 224.1.1.1 ) 00:09:24 NumOIF: 2 profile: none
```

```
          Outgoing Interfaces:
               CCEP rbr-id 200 vlan 20 ( V2) 00:00:42/42s
               e2/2 vlan 20 ( V2) 00:05:15/36s
  1    (1.1.1.1, 224.1.1.1) in e2/15 vlan 20 00:00:23 NumOIF: 1 profile: none
          Outgoing Interfaces:
               e2/2 vlan 20 ( V2) 00:00:23/0s
          FID: 0x8006 MVID: None
```

MCT Client:

```
device# show ip multicast vlan 20
----+-----+---------+---------------+-----+-----+------
VLAN State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+---------------+-----+-----+------
20   I-Ena Passive   77.77.77.1      81    1     1
----+-----+---------+---------------+-----+-----+------
Router ports: 4/3 (97s)
Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join
  1    (*, 224.1.1.1 ) 00:07:09 NumOIF: 2 profile: none
          Outgoing Interfaces:
               e4/3 vlan 20 ( R) 00:02:42/97s
               e3/2 vlan 20 ( V2) 00:07:09/91s
  1    (1.1.1.1, 224.1.1.1) in e4/3 vlan 20 00:02:20 NumOIF: 1 profile: none
          Outgoing Interfaces:
               e3/2 vlan 20 ( V2) 00:02:20/0s
          FID: 0x8005 MVID: None
```

Syntax: show ip multicast vlan *vlan-id*

The **vlan***vlan-id* parameter displays IP multicast VLAN information for a specified VLAN.

Displaying IP multicast statistics

To display IP multicast statistics on a device, enter the following commands at any level of the CLI.

```
device##show ip multicast vlan 20 statistics
VLAN ID 20
  Receive stats:                          Transmit stats:
  General query           : 0             General query           : 3
  Group specific query    : 0             Group specific query    : 0
  IGMP Report             : 14            IGMP V2 Proxy Sent      : 0
  IGMP Leave              : 1             IGMP V3 Proxy Sent      : 0
  IGMPV3 Report           : 0             PIM Proxy Sent          : 0
  IGMPV3 Error            : 0             MCT MDUP msg recvd      : 10
  PIMV2 hello             : 0
  PIMV2 join/prune        : 0
  PIMV2 J/P pkt error     : 0
  MCT MDUP msg sent       : 5
  MCT MDUP msg error      : 0
```

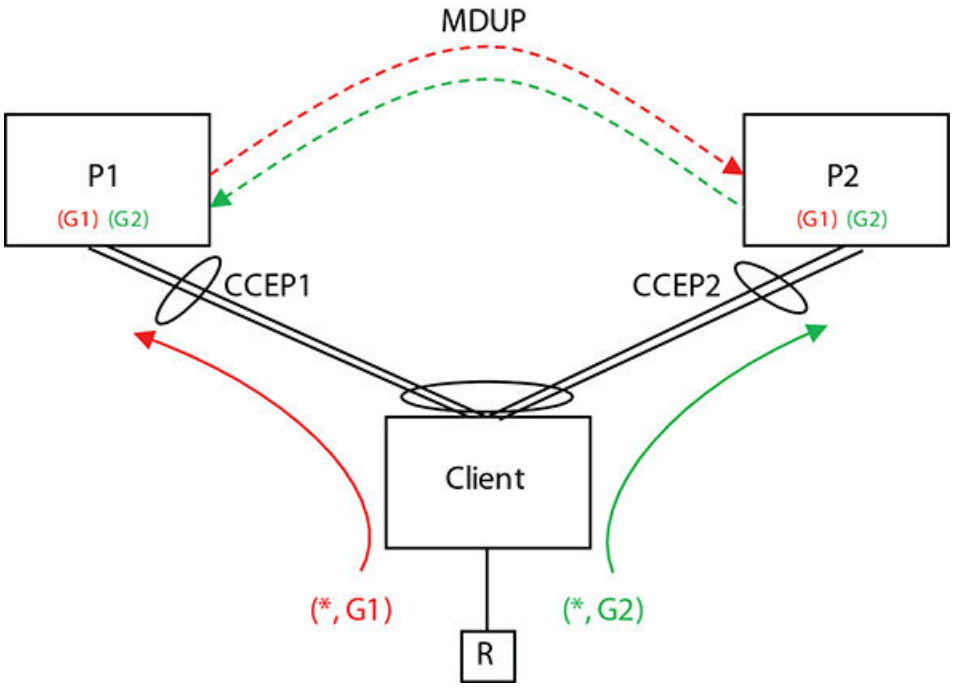
Syntax: show ip multicast statistics

For information regarding the output fields, refer to "Displaying IP multicast information".

PIM Over MCT

Figure 155 is an example of the IGMP synchronization.

FIGURE 153 Synchronizing IGMP State on the CCEPs



Synchronizing IGMP State on the CCEPs

The MDUP channel sends an IGMP report of messages received on the local CCEP to the remote MCT peer. This ensures that regardless of which MCT peer the client forwards the messages, both will recieve it and process it, ensuring identical state in both MCT Peers.

PIM Over MCT on page 507 displays R sending IGMP report for (*, G1).

NOTE

If the multicast source lies on the uplink, and receivers lie on the MCTclients, both MCT nodes will forward traffic to the clients if the ICL goes down. To avoid this situation, the keepalive VLAN must always be configured under the cluster to enable Master-Slave election and prevent duplicate traffic to CCEP receivers.

Synchronization method.

Client forwards it to P1 on CCEP1.

P1 sends this to P2 via MDUP.

P2 adds (*, G1) to CCEP2 upon processing it.

R sends IGMP report for (*, G2).

Client forwards it to P2 on CCEP2.

P2 sends this to P1 via MDUP.

P1 adds (*, G2) to CCEP1 upon processing it.

Final result, both P1 and P2 has both (*, G1) and (*, G2) on their local CCEPs.

Traffic Load sharing on the CCEPs

For a stream that has a receiver behind the Client, both MCT Peers recieve the traffic, and each has to decide whether to forward the traffic to the local CCEP or let the remote peer take care of it. The following simple decision function is used to determine this.

- If the CCP is down, it will forward locally.
- If the remote CCEP is down, it will forward locally.
- If the local CCEP is down, it will not forward locally.
- If the ingress is the CEP, it will forward locally.
- If the ingress is the ICL, it will not forward locally.
- If the ingress is a different CCEP, it will forward locally.

Use the following expression to determine the traffic load sharing.

Forward locally =

((Src_addr + Grp_addr) & 0x00000001) ^ ((UINT32)(local_brdige_id > remote_bridge_id))

Sending IGMP Queries on CCEPs

Since there are two chassis connected to the same MCT VLAN and since the Client is not going to flood the incoming IGMP Queries to the other chassis, both chassis could end up electing themselves as the IGMP Querier and sending queries on the CCEPs and the hosts behind the client will end up responding to both. To avoid this, the IGMP surpresses queries going out of the CCEP port on one of the MCT Peers. The following algorithm is used:

If (CCP connection is down)

Send Queries on local CCEP.

Else if (remote CCEP is down)

Send Queries on local CCEP.

Else if (local CCEP is down)

Suppress Queries on local CCEP.

Else if (local-bridge-id > remote-bridge-id)

Send Queries on local CCEP.

Else

Suppress Queries on local CCEP.

Enabling PIM over MCT scaling optimization

By default, PIM over MCT has a scaling limit of 2K mcache (S,G) entries and 512 IGMP entries. You can increase the maximum scaling of these entries to 16K mcache entries and 4K IGMP entries.

The increase in scaling optimizes the MDUP between the MCT peers to minimize the processing of packets received from the CCEP ports downstream. The MDUP from one MCT peer to another peer occurs when one of the MCT peer sends the General Query (GS), Group Specific Query (GSQ), and Group Source Specific Query (GSSQ) queries. This peer is the querier in the MCT VLAN.

NOTE
This feature is not supported for snooping over MCT.

Perform the following steps to enable scaling optimization on both MCT peers.

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

```
device# configure terminal
```

2. Enable IPv4 PIM over MCT scaling optimization.

```
device(config)# ip multicast-routing optimization mct-scaling
```

   IPv6 MCT scaling optimization is also supported by the **ipv6 multicast-routing optimization mct-scaling** command.

3. Verify that the feature is enabled.

```
device(config)# show ip pim global
Global IPv4 PIM Settings
  ...
  MCT Scaling Optimization  : enabled
```

The following example is the configuration of the previous steps.

```
device# configure terminal
device(config)# ip multicast-routing optimization mct-scaling
```

Displaying IGMP and MLD cluster group information

To display the IGMP cluster groups, use the **show ip igmp cluster-client group** command.

```
device# show ip igmp cluster-client group
Total 1 groups
--------------------------------------------------------------------------------
Idx   Group Address    Port   Intf   GrpCmpV Mode    Timer Refreshed MDUPReq Srcs
----+---------------+------+------+-------+-------+------+---------+-------+------
   1  10.1.1.1        e1/7   v10       Ver2 exclude   237      N        N     0

Total number of groups 1

Groups having cluster clients 1
```

To display the MLD cluster group, use the **show ipv6 mld cluster-client group** command.

```
device# show ipv6 mld cluster-client group 123::3
Total 1 groups
--------------------------------------------------------------------------------------------------------------
Idx  Group Address                              Port   Intf   GrpCmpV Mode    Timer Refreshed MDUPReq Srcs
----+---------------------------------------+------+------+-------+-------+-----+---------+-------+----
   1 123::3                                  e1/7   v10       Ver1 exclude   253     Y        N      0
```

To display the number of IGMP cluster groups, use the **show ip igmp group count** command.

```
device# show ip igmp group count
Total IGMP groups : 4096
```

Displaying PIM mcache table information

Use the **show ip pim mcache** command to display if a CCEP port is being blocked or being forwarded to.

```
device#show ip pim mcache
IP Multicast Mcache Table
Entry Flags    : SM  - Sparse Mode, SSM - Source Specific Mutlicast, DM - Dense Mode
                 RPT    - RPT Bit, SPT - SPT Bit, LSRC - Local Source, LRCV - Local Receiver, JOIN - Join
Upstream
                 HW - HW Forwarding Enabled, FAST - Resource Allocated, TAG - Need For Replication Entry
                 REGPROB - Register In Progress, REGSUPP - Register Suppression Timer
```

```
                MSDPADV - Advertise MSDP, NEEDRTE - Route Required for Src/RP,  PRUN - DM Prune Upstream
Interface Flags: IM - Immediate, IH - Inherited, WA - Won Assert
                MJ - Membership Join, MI - Membership Include, ME - Membership Exclude
                BR - Blocked RPT, BA - Blocked Assert, BF - Blocked Filter, BI - Blocked IIF,
            BM - Blocked MCT
Total entries in mcache: 1
1     (2.2.2.101, 239.0.1.3) in v200 (e2/15), Uptime 00:01:08, Rate 42229 (SM)
      Source is directly connected. RP 2.2.2.1
      Flags (0x3046cec1) SM SPT L2REG LSRC LRCV JOIN HW FAST MSDPADV
      fast ports: ethe 2/1
      AgeSltMsk: 00000002,  FID: 0x8006, MVID: NotReq, RegPkt: 0, AvgRate: 41688, profile: none
      Forwarding_oif: 1, Immediate_oif: 1, Blocked_oif: 1
      L2 (HW) 1:
          TR(e2/1,e2/1),  00:01:08/181, Flags: IM IH
      Blocked OIF 1:
          TR(e1/5,e1/5)(VL200), 00:01:08/0, Flags: MJ BM
Number of matching entries: 1
device#
```

Use the **show ipv6 pim mcache** command to display if a CCEP port is being blocked or being forwarded to.

```
MLX#show ipv6 pim mcache
IP Multicast Mcache Table
Entry Flags    : SM  - Sparse Mode, SSM - Source Specific Mutlicast, DM - Dense Mode
                 RPT    - RPT Bit, SPT - SPT Bit, LSRC - Local Source, LRCV - Local Receiver, JOIN - Join
Upstream
                 HW - HW Forwarding Enabled, FAST - Resource Allocated, TAG - Need For Replication Entry
                 REGPROB - Register In Progress, REGSUPP - Register Suppression Timer
                 MSDPADV - Advertise MSDP, NEEDRTE - Route Required for Src/RP,  PRUN - DM Prune Upstream
Interface Flags: IM - Immediate, IH - Inherited, WA - Won Assert
                 MJ - Membership Join, MI - Membership Include, ME - Membership Exclude
                 BR - Blocked RPT, BA - Blocked Assert, BF - Blocked Filter, BI - Blocked IIF,
             BM - Blocked MCT
Total entries in mcache: 1
1 (2062:62:62:62::11, ff1a::2) in v62 (tag e1/1), Uptime 00:00:58 (SM)      upstream neighbor is L2
fe80::21b:edff:fea4:a441. RP 2001:1:2:3:4::b      Flags (0x304680c1) SM SPT LSRC LRCV HW FAST      fast
ports:       AgeSltMsk: 00000003, FID: 0xffff (D), DIT: NotReq, profile: none, KAT Timer value:
240      Forwarding_oif: 0, Immediate_oif: 0, Blocked_oif: 1       Blocked OIF 1:
             TR(e1/39,e1/39)(VL62), 00:00:01/0, Flags: MJ BM
Number of matching entries: 1
MLX#
```

Displaying MCT PIM Counters

To display the statistics and error counters for the Multicast MDUP channel between the MCT peers, use the **show ip pim counter mct** command.

```
device#show ip pim count mct
Multicast MCT Statistics for IPv4 (UP):
Messages assembled into the send buffer  : 11811
Messages processed out of the recv buffer: 0
Segments sent successfully to TCP        : 11762
Segments failed to be accepted by TCP    : 0
Segments assembled into the receive buffer : 0
Messages dropped because (size > 1500)     : 0
Messages dropped because it won't fit into available space in send buffer   : 0
Segments dropped because it won't fit into available space in receive buffer: 0
Received messages dropped because of cluster-id mismatch                    : 0
Received messages dropped because the peer was not recognized               : 0
Received messages dropped because cluster not active                        : 0
Received messages dropped because MCT VLAN unrecognized                     : 0
Received messages dropped because of bad message type                       : 0
Received messages dropped because of bad checksum                           : 0
Received bytes skipped because of sync or checksum errors                   : 0
PIM Hello Messages sent  : 0
PIM J/P Messages sent    : 0
PIM Assert Messages sent : 0
PIM Unknown not sent     : 0
PIM Hello Messages received     : 0
```

```
PIM J/P Messages received       : 0
PIM Assert Messages received   : 0
PIM Unknown received & dropped : 0
IGMPv1 reports sent  : 0
IGMPv2 reports sent  : 0
IGMPv3 reports sent    : 0
IGMP leaves sent : 0
IGMP queries sent : 0
IGMP unknown not sent     : 0
IGMPv1 reports received : 0
IGMPv2 reports received : 0
IGMPv3 reports received : 0
IGMP leaves received   : 0
IGMP queries received   : 0
IGMP unknown received & dropped : 0
device#
```

To display the MCT IPv6 PIM counter, use the **show ipv6 pim counter mct** command.

```
device#show ipv6 pim count mct
Multicast MCT Statistics for IPv6 (UP):
Messages assembled into the send buffer  : 279
Messages processed out of the recv buffer: 523
Segments sent successfully to TCP        : 279
Segments failed to be accepted by TCP    : 0
Segments assembled into the receive buffer : 293
Messages dropped because (size > 1500)     : 0
Messages dropped because it won't fit into available space in send buffer   : 0
Segments dropped because it won't fit into available space in receive buffer: 0
Received messages dropped because of cluster-id mismatch                    : 0
Received messages dropped because the peer was not recognized               : 0
Received messages dropped because cluster not active                        : 0
Received messages dropped because MCT VLAN unrecognized                     : 0
Received messages dropped because of bad message type                       : 0
Received messages dropped because of bad checksum                           : 0
Received bytes skipped because of sync or checksum errors                   : 0
PIM Hello Messages sent  : 0
PIM J/P Messages sent    : 0
PIM Assert Messages sent : 0
PIM Unknown not sent     : 0
PIM Hello Messages received     : 0
PIM J/P Messages received       : 0
PIM Assert Messages received   : 0
PIM Unknown received & dropped : 0
MLDv1 reports sent  : 0
MLDv2 reports sent    : 0
MLD leaves sent : 0
MLD queries sent : 0
MLD unknown not sent     : 0
MLDv1 reports received  : 0
MLDv2 reports received  : 0
MLD leaves received   : 0
MLD queries received   : 0
MLD unknown received & dropped : 0
device#
```

Displaying IGMP and MLD interfaces

Use the **show ip igmp interface** command to show the IGMP Query suppression state on CCEP ports.

```
device#show ip igmp interface
---------+------+---------+---------------+---------+-----+-----+---------
Intf/Port|Groups| Version |Querier        |  Timer  |V1Rtr|V2Rtr|Tracking
         |      |Oper  Cfg|               |OQrr GenQ|     |     |
---------+------+----+----+---------------+----+----+-----+-----+---------
v200           2    2    -                                        Disabled
    e2/15           2    - Self               0   59 No    No
     e2/1           2    - Self               0   59 No    Yes
```

```
     e1/5           2    - Self (MCT-Blk)     0   40 No    No
device#
```

Refer to "Displaying the IGMP status of an interface" for output descriptions.

Use the **show ipv6 mld interface** command to show the IGMP Query suppression state on CCEP ports.

```
device# show ipv6 mld interface
---------+------+---------+--------------------------------------+---------+-----+---------
Intf/Port|Groups| Version |Querier                               |  Timer  |V1Rtr|Tracking
         |      |Oper  Cfg|                                      |OQrr GenQ|     |
---------+------+----+----+--------------------------------------+----+----+-----+---------
v62            0    2    2                                                    Disabled
     e2/1           2    - Self (MCT-Blk)                          0   79 No
    e1/37           2    - Self                                    0  108 No
    e1/33           2    - Self                                    0  108 No
device#
```

Refer to "Displaying IPv6 PIM interface information" for output descriptions.

Sample configuration

MCT Peer 1

```
lag "AMERICAS-CCEP-1-1" static id 11
 ports ethernet 1/5 to 1/6
 primary-port 1/5
 deploy
lag "AMERICAS-ICL" static id 1
 ports ethernet 2/1 to 2/2
 primary-port 2/1
 deploy
vlan 200
 untagged ethe 2/15
 tagged ethe 1/5 to 1/6 ethe 2/1 to 2/2
 router-interface ve 200
vlan 4090
 tagged ethe 2/1 to 2/2
 router-interface ve 100
router pim
 rp-address 2.2.2.1
interface management 1
 ip address 10.25.109.6/21
 enable
interface ve 100
 ip address 1.1.1.1/24
interface ve 200
 ip address 2.2.2.1/24
 ip pim-sparse
cluster AMERICAS 1
 rbridge-id 100
 session-vlan 4090
 member-vlan 200
 icl AMERICAS-ICL ethernet 2/1
 peer 1.1.1.2 rbridge-id 200 icl AMERICAS-ICL
 client-interfaces sync_ccep_early lacp-delay 5
 deploy
 client AMERICAS-CLIENT-1
  rbridge-id 300
  client-interface ethernet 1/5
  deploy
```

MCT Peer 2

```
lag "AMERICAS-CCEP-2-1" static id 21
 ports ethernet 3/5 to 3/6
```

```
 primary-port 3/5
 deploy
lag "AMERICAS-ICL" static id 1
 ports ethernet 4/1 to 4/2
 primary-port 4/1
 deploy
vlan 200
 untagged ethe 4/23
 tagged ethe 3/5 to 3/6 ethe 4/1 to 4/2
 router-interface ve 200
vlan 4090
 tagged ethe 4/1 to 4/2
 router-interface ve 100
router pim
 rp-address 2.2.2.1
interface management 1
 ip address 10.25.109.5/21
 enable
interface ve 100
 ip address 1.1.1.2/24
interface ve 200
 ip address 2.2.2.2/24
 ip pim-sparse
cluster AMERICAS 1
 rbridge-id 200
 session-vlan 4090
 member-vlan 200
 icl AMERICAS-ICL ethernet 4/1
 peer 1.1.1.1 rbridge-id 100 icl AMERICAS-ICL
 client-interfaces sync_ccep_early lacp-delay 5
 deploy
 client AMERICAS-CLIENT-1
  rbridge-id 300
  client-interface ethernet 3/5
  deploy
```

BFD over MCT

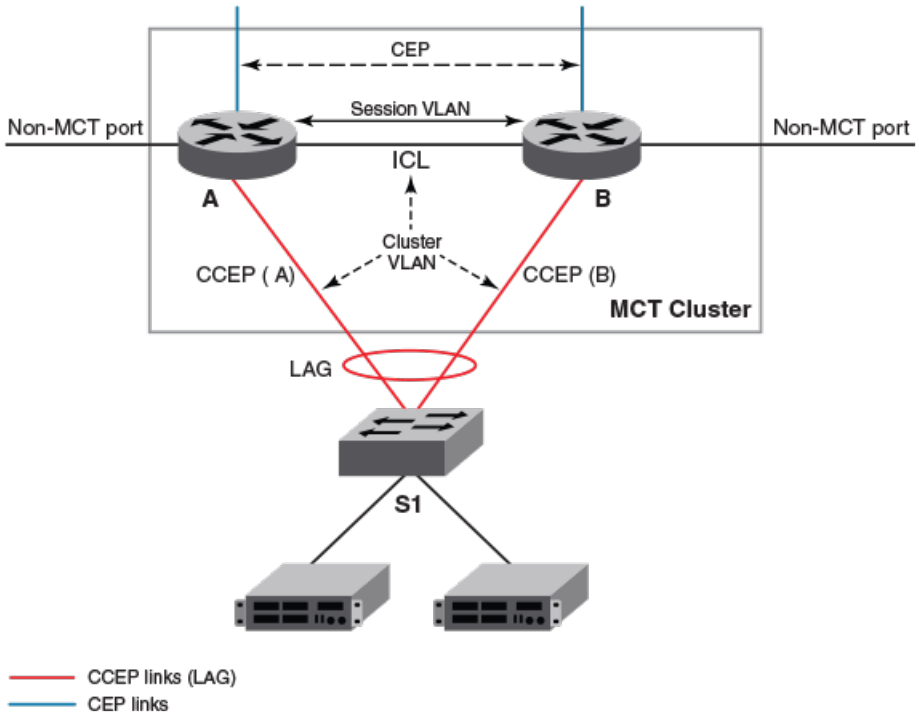
Bidirectional Forwarding Detection (BFD) over MCT detects forwarding path failures in different applications that are configured on MCT cluster devices.

BFD sessions are formed between clients of MCT clusters and MCT nodes. BFD sessions over an MCT member virtual ethernet (VE) interface detect forwarding path failures in the following applications:

- Static routes
- OSPFv2 and OSPFv3
- IS-IS
- BGP4 and BGP4+

For more information about BFD, refer to the *Bidirectional Forwarding Detection* chapter.

FIGURE 154 BFD over MCT



In the network diagram, applications on router A send requests to BFD to detect the status of the protocol peers on S1 and clients beyond S1. The following events are observed when BFD is implemented on router A.

BFD at the Rx port:

- Successfully receives periodic BFD packets sent by the MCT client and maintains the BFD state.
- The receiving port can be either ICL or CCEP.
- Detects the loss of connectivity with the application peer on the MCT client within BFD time constraints.
- Differentiates a CCEP or ICL failure from a complete breakdown.

BFD at the Tx port:

- Sends periodic BFD control packets to the MCT client directly via CCEP or via ICL ports.
- Moves BFD Tx to ICL before BFD times out on the MCT client when CCEP is down.
- Moves BFD Tx to CCEP before BFD times out on the MCT client when the ICL is down.

Use case: BFD over MCT with multiple LAGs

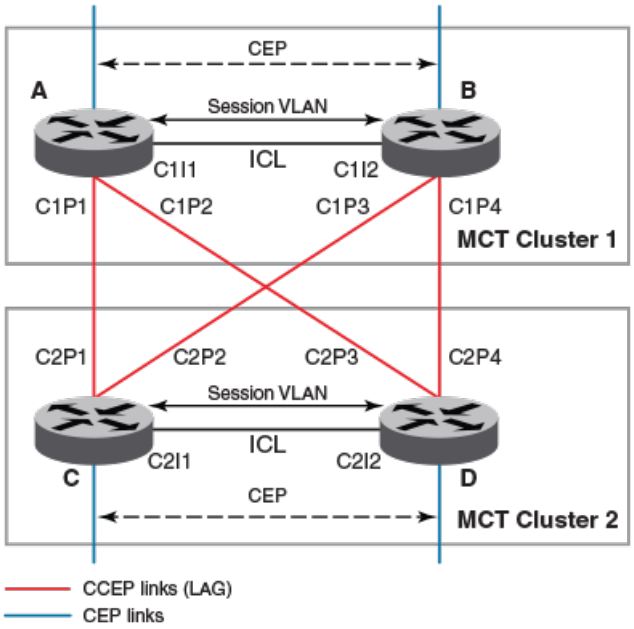
BFD sessions are formed between MCT cluster clients and MCT nodes. The VEs span ICL and CCEP ports; thus BFD packets from clients can reach MCT nodes on either the CCEP port or the ICL link. The local BFD Tx port can be on either of these ports. To support this configuration, BFD session processing is modified to accept and send packets on VEs with multiple LAGs as members.

Let us consider that a BFD session formed between A and C, as shown in the following network diagram. The following scenarios describe how the BFD Tx port is switched when a port goes down to ensure that there is no impact to the ongoing BFD session.

- Router A's Tx port is C1P1, and router C's Rx port is C2P1. When C1P1 is down, router A's Tx port switches to ICL port C1I1. When router C receives the BFD packet on port C2I1, it also switches its Rx port to the C2I1 port.

- Router A's Tx port is C1P2, and router C's Rx port is C2I1. When C1P2 goes down, router A's Tx port switches to ICL port C1I1, while router C's Rx port does not change.
- Router A's Tx port is C1I1, and router C's Rx port is C2P2. When C1I1 goes down, router A's Tx port changes to either the C1P1 port or the C1P2 port. When router C receives the BFD packet on C2P1 or C2I1, its Rx port also changes to C2P1 or C2I1, respectively.

FIGURE 155 BFD over MCT with multiple LAGs



BFD over MCT limitations

- BFD sessions may go down or flap when MAC learning occurs on MCT devices.
- BFD over MCT does not support VE over Virtual Private LAN Service (VPLS).

Upgrade and downgrade considerations

- During a network upgrade procedure, BFD sessions are established between the MCT client and the upgraded MCT device.
- BFD should be disabled on MCT member VEs before the downgrade to prevent impact on certain applications.

BFD over MCT scalability

- BFD over MCT supports all members of the LAG.
- BFD over MCT supports BFD timers for values between 100 and 30000 milliseconds.

NOTE

When CER 2000 Series and CES 2000 Series devices are heavily loaded with many BFD sessions, the BFD sessions may flap if the configured BFD interval is less than 500 ms with a multiplier value of 3. For multihop IPv4 session, BFD sessions are stable at 500 ms with a multiplier value of 3. For multihop IPv6 session, BFD sessions are stable at 600 ms with a multiplier value of 3.

- On the MLX Series and XMR Series devices, 250 BFD sessions are supported. However, 40 BFD sessions are supported on an LP.
- On the CER 2000 Series and CES 2000 Series devices, 40 BFD sessions are supported. 40 BFD sessions are supported on an LP.
- When BFD supports a maximum of 40 session per LP or a LAG, BFD sessions may switch to ICL LAG from CCEP LAG when CCEP is down or an interface in CCEP LAG is down. Hence BFD can support a maximum of 40 sessions over MCT cluster.
- For the configuration tabulated below on a CER 2000 Series and CES 2000 Series device, BFD is stable at a timer value of 5000 msec * 3. At timer value lesser than 5000 msec * 3, BFD session may flap.

Scenario or protocol	Session type	Session type
OSPF	**IPv4 session**	**IPv6 session**
Neighbor	5	5
Routes	130K	1.5K
BGP	**IPv4 session**	**IPv6 session**
Neighbor	5	5
Routes	100K	4K
MPLS	**Configured**	**Peer**
VLL	100	300
VPLS	256	768
Number of MPLS tunnels allocated	30	
Number of MPLS cross-connects allocated	56	
BFD		
Number of BFD sessions	20	

Configuring BFD over MCT

BFD over MCT is configured to detect forwarding path failures in different applications that are configured on MCT cluster devices.

1. Configure the MCT cluster on the NetIron OS device.
   a) Create LAG for CCEP ports

```
device(config-lag-ccep)#
device(config-lag-ccep)# lag ccep dynamic id 2
device(config-lag-ccep)# ports ethernet 4/17 ethernet 4/18
device(config-lag-ccep)# primary-port 4/17
device(config-lag-ccep)# deploy
```

   b) Create LAG for ICL ports

```
device(config)# lag icl dynamic id 1
device(config-lag-icl)# ports ethernet 1/2 ethernet 3/3
device(config-lag-icl)# primary-port 1/2
device(config-lag-icl)# deploy
```

   c) Configure member VLAN

```
device(config)# vlan 7 name member_vlan
device(config-vlan-7)# tagged ethernet 1/2 ethernet 3/3 ethernet 4/17 ethernet 4/18
device(config-vlan-7)# router-interface ve 7
```

   d) Configure session VLAN

```
device(config)# vlan 10 name session_vlan
device(config-vlan-10)# tagged ethernet 1/2 ethernet 3/3
device(config-vlan-10)# router-interface ve 10
```

   e) Assign IP address to session VLAN virtual interface

```
device(config)# interface ve 10
device(config-vif-10)# ip address 10.10.10.1/24
```

   f) Configure the cluster

```
device(config)# cluster bfd 10
device(config-cluster-bfd)# rbridge-id 11
device(config-cluster-bfd)# session-vlan 10
device(config-cluster-bfd)# member-vlan 7
device(config-cluster-bfd)# icl icl ethernet 1/2
device(config-cluster-bfd)# peer 10.10.10.2 rbridge-id 10 icl icl
device(config-cluster-bfd)# deploy
device(config-cluster-bfd)# client "client_1"
device(config-cluster-bfd-client-client_1)# rbridge-id 2
device(config-cluster-bfd-client-client_1)# client-interface ethernet 4/17
device(config-cluster-bfd-client-client_1)# deploy
device(config-cluster-bfd-client-client_1)# exit
device(config-cluster-bfd)# exit
```

2. Configure applications that use BFD on the device.

   Configure OSPF on global mode

```
device(config)# router ospf
device(config-ospf-router)# area 0.0.0.0
device(config-ospf-router)# bfd all-interfaces
```

3. Configure BFD parameters on the device.

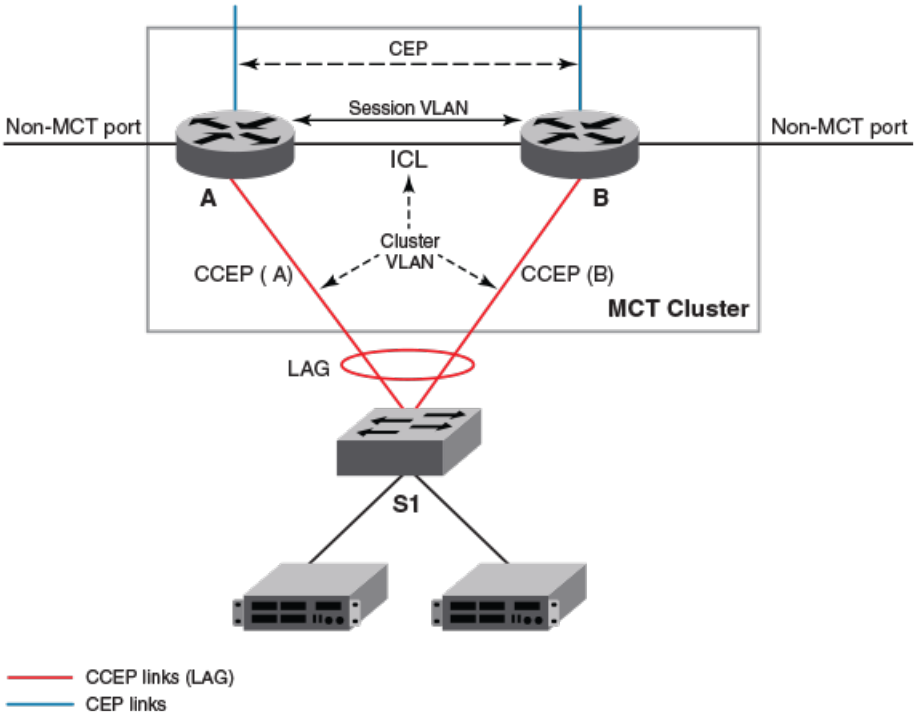
   Assign the IP address and enable BFD on the member VLAN virtual interface

```
device(config)# interface ve 7
device(config-vif-7)# ip address 10.20.20.2/24
device(config-vif-7)# bfd interval 300 min-rx 300 multiplier 3
device(config-vif-7)# ip ospf area 0.0.0.0
device(config-vif-7)# ip ospf bfd
```

BFD over MCT configuration example

The following example configures BFD over MCT on a NetIron OS device.

FIGURE 156 BFD over MCT



The following example illustrates the MCT configuration at router A of the network diagram.

```
lag "ccep" static id 2
 ports ethernet 1/1 ethernet 1/4
 primary-port 1/1
 deploy
!
lag "icl" static id 1
 ports ethernet 1/2 ethernet 4/5
 primary-port 1/2
 deploy
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/2 ethe 4/5
!
vlan 7 name CLIENT
 tagged ethe 1/1 to 1/2 ethe 1/4 ethe 4/5
 router-interface ve 7
 static-mac-address 0024.3843.6d41 ethernet 1/1
!
vlan 10 name ICL
 tagged ethe 1/2 ethe 4/5
 router-interface ve 10
!
no route-only
!
router ospf
 area 0.0.0.0
 bfd all-interfaces
!
interface ve 7
 bfd interval 300 min-rx 300 multiplier 3
 ip ospf area 0.0.0.0
 ip ospf bfd
 ip address 192.0.6.0/24
!
interface ve 10
 ip address 10.10.10.1/24
!
cluster "bfd" 10
 rbridge-id 11
 session-vlan 10
 cluster-client-static-mac-move
 member-vlan 7
 icl icl ethernet 1/2
 peer 10.10.10.2 rbridge-id 10 icl icl
 deploy
 client "CES-3"
  rbridge-id 2
  client-interface ethernet 1/1
  deploy
!
end
```

The following example illustrates the MCT configuration at router B of the network diagram.

```
lag "ccep" static id 2
 ports ethernet 1/1 ethernet 1/3
 primary-port 1/1
 deploy
!
lag "icl" static id 1
 ports ethernet 1/2 ethernet 3/5
 primary-port 3/5
 deploy
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/2 ethe 3/5
!
vlan 7
 tagged ethe 1/1 to 1/3 ethe 3/5
 router-interface ve 7
 static-mac-address 0024.3843.6d41 ethernet 1/1
!
vlan 10
 tagged ethe 1/2 ethe 3/5
 router-interface ve 10
!
router ospf
 area 0.0.0.0
!
interface ve 7
 bfd interval 300 min-rx 300 multiplier 3
 ip ospf area 0.0.0.0
 ip ospf bfd
 ip address 192.0.16.0/24
!
interface ve 10
 ip address 10.10.10.2/24
!
cluster "bfd" 10
 rbridge-id 10
 session-vlan 10
 cluster-client-static-mac-move
 member-vlan 7
 icl icl ethernet 3/5
 peer 10.10.10.1 rbridge-id 11 icl icl
 deploy
 client "CES-3"
  rbridge-id 2
  client-interface ethernet 1/1
  deploy
!
end
```

The following example illustrates the MCT configuration at S1 of the network diagram.

```
lag "client" static id 1
 ports ethernet 1/13 to 1/16
 primary-port 1/13
 deploy
!
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/5 to 1/6
!
vlan 7
 tagged ethe 1/13 to 1/16
 router-interface ve 7
!
router ospf
 area 0.0.0.0
!
interface ve 7
 bfd interval 300 min-rx 300 multiplier 3
 ip ospf area 0.0.0.0
 ip ospf bfd
 ip address 192.0.10.0/24
!
end
```

Displaying BFD information

Various show commands can be used to display BFD information.

Use one or more of the following commands to display BFD information. The commands do not have to be entered in this order.

1. Enter the **show bfd** command to display current registered protocol, BFD state and the number of BFD sessions available.

```
device(config)# show bfd
 BFD State: ENABLED Version: 1 Use PBIF Assist: Y SH setup delay 180 MH setup delay 0
 Current Registered Protocols: ospf/0
 All Sessions: Current: 2 Maximum Allowed: 250 Maximum Exceeded Count: 0
 Maximum TX/RX Sessions Allowed on LP: 80  Maximum Session Exceeded Count for LPs: 0
   LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions
   1   0/0           2   0/0           3   1/1           4   1/1
 BFD Enabled ports count: 1
 Port       MinTx       MinRx       Mult Sessions
 ve 7       300         300         3    2
```

2. Enter the **show bfd neighbor** command to display the total number of neighbor entries, neighbor address, and the current state of BFD.

```
device(config)# show bfd neighbors
Total Entries:2 R:RXRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
NeighborAddress                 State  Interface       Holddown   Interval  R/H
20.20.20.3                      UP     ve 7            900000     300000    Y/S
20.20.20.2                      UP     ve 7            900000     300000    Y/S
```

3. Enter the **show bfd neighbor details** command to display information about the Tx and Rx ports where BFD control messages are received from the remote peer.

```
device(config)# show bfd neighbors details
Total Entries:2 R:RXRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
NeighborAddress                 State  Interface       Holddown   Interval  R/H
20.20.20.3                      UP     ve 7            900000     300000    Y/S
  Registered Protocols(Protocol/VRFID): ospf/0
  Local: Disc: 5, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
  Remote: Disc: 1, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
```

```
  Stats: RX: 3648 TX: 4249 SessionUpCount: 1 at SysUpTime: 0:16:29:10.591
  Session Uptime: 0:0:16:36.666, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 4/17,RX: eth 4/17,Vlan Id: 7
  Using PBIF Assist: Y
NeighborAddress                       State  Interface       Holddown   Interval  R/H
20.20.20.2                            UP     ve 7            900000     300000    Y/S
  Registered Protocols(Protocol/VRFID): ospf/0
  Local: Disc: 6, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
  Remote: Disc: 6, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
  Stats: RX: 4530 TX: 4071 SessionUpCount: 1 at SysUpTime: 0:16:29:10.591
  Session Uptime: 0:0:16:36.666, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 3/3,RX: eth 3/3,Vlan Id: 7
  Using PBIF Assist: Y
```

4. Enter the **show bfd applications** command to display information about the registered protocol count.

```
device(config)# show bfd applications
  Registered Protocols Count: 1
     Protocol  VRFID     Parameter HoldoverInterval
     ospf      0         0         0
```

Multiple VLAN Registration Protocol (MVRP)

Multiple VLAN Registration Protocol

Multiple VLAN Registration Protocol (MVRP) is an MRP application that provides IEEE 802.1ak-compliant VLAN pruning and dynamic VLAN creation on switch ports. It allows dynamic configuration of a VLAN over intermediate switches joining a set of access switches declaring a particular VLAN. MVRP aware switches exchange VLAN configuration information and maintains a dynamic reachability tree connecting all devices interested in a particular VLAN. MVRP VLAN pruning, using the reach ability tree, limits the scope of unnecessary broadcast and unknown unicast to a set of interested end devices only.

MVRP allows the propagation of VLAN information from device to device. With MVRP, an access switch is manually configured with all the desired VLANs for the network, and all other switches on the network learn those VLANs dynamically.

Enabling MVRP

MVRP must be enabled globally to allow the device to participate in the protocol.

1. On any device on which you want to configure MVRP service, from privileged EXEC mode, enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enable MVRP globally.

   ```
   device(config)# mvrp enable
   ```

3. Configure the join, leave, and leave-all timer for MVRP.

   ```
   device(config)# mvrp timer join 400 leave 2000 leave-all 10000
   ```

4. Configure MVRP at the interface level.

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

5. Enable MVRP on the interface.

   ```
   device(config-if-e1000-1/1)# mvrp enable
   ```

6. Configure port applicant mode as non-participant.

   ```
   device(config-if-e1000-1/1)# mvrp applicant-mode non-participant
   ```

7. (Optional) Configure point-to-point over a port.

   This configuration is required when a port is connected to a shared media device.

```
device(config-if-e1000-1/1)# mvrp point-to-point
```

8. (Optional) Configure MVRP timers at the interface level.

```
device(config-if-e1000-1/1)# mvrp timer join 400 leave 1500 leave-all 8000
```

9. Configure registration mode for an interface.

```
device(config-if-e1000-1/1)# mvrp registration-mode forbidden vlan 10
```

10. Return to privileged EXEC mode.

```
device(config-if-e1000-1/1)# end
```

11. Verify the MVRP configuration.

```
device# show mvrp config
mvrp enable
mvrp timer join 400 leave 2000 leave-all 10000
!
interface ethernet 1/5
  mvrp enable
  mvrp registration-mode forbidden vlan 10
  mvrp timer join 400 leave 1500 leave-all 8000
  mvrp point-to-point
  mvrp applicant-mode non-participant
```

12. Display the MVRP statistics.

```
device# show mvrp statistics
Port  : ethe 1/1
    ------------------------------------------------------------
    Message type                  Received     Transmitted
    ------------------------------------------------------------
    New                           0            0
    In                            0            0
    Join In                       0            0
    Join Empty                    0            0
    Empty                         0            0
    Leave                         0            0
    Leave-all                     0            0
    ------------------------------------------------------------
    Total PDUs                    0            0
    ------------------------------------------------------------
```

Clearing MVRP statistics

MVRP session counters can be cleared using a CLI command.

Ensure that MVRP is enabled in your network.

To determine the effect of clearing the MVRP statistics, an appropriate **show** command is entered before and after the **clear** command.

1. From the privileged EXEC mode, enter the **show mvrp statistics** command for Ethernet 1/1.

```
device# show mvrp statistics ethernet 1/1
Port  : ethe 1/1
    ------------------------------------------------------------
    Message type                  Received     Transmitted
    ------------------------------------------------------------
    New                           10           10
    In                            0            0
    Join In                       0            0
    Join Empty                    0            0
    Empty                         0            0
    Leave                         5            3
    Leave-all                     0            0
    ------------------------------------------------------------
    Total PDUs                    0            0
    ------------------------------------------------------------
```

2. Enter the **clear mvrp statistics** command for the interface Ethernet 1/1.

```
device# clear mvrp statistics ethernet 1/1
```

3. Enter the **show mvrp statistics** command for Ethernet 1/1.

```
device# show mvrp statistics ethernet 1/1
Port  : ethe 1/1
    ------------------------------------------------------------
    Message type                  Received     Transmitted
    ------------------------------------------------------------
    New                           0            0
    In                            0            0
    Join In                       0            0
    Join Empty                    0            0
    Empty                         0            0
    Leave                         0            0
    Leave-all                     0            0
    ------------------------------------------------------------
    Total PDUs                    0            0
    ------------------------------------------------------------
```

In this show output for a specified interface after the **clear mvrp statistics** command has been entered, you can see that the statistical counters have been reset.

MVRP configuration examples

Single interface MVRP configuration example

```
device(config)#int e 1/1
device(config-eth-1/1)#mvrp enable
device(config-eth-1/1)#mvrp registration-mode forbidden vlan 10
device(config-eth-1/1)#mvrp timer join 400 leave 1400 leave-all 10000
```

Multiple Interface (consecutive) MVRP configuration example

```
device(config)#int e 1/1 to e 1/2
device(config-mif-1/1-1/2)#mvrp enable
device(config- mif-1/1-1/2)#mvrp registration-mode forbidden vlan 10
device(config- mif-1/1-1/2)#mvrp timer join 400 leave 1400 leave-all 10000
```

Multiple Interface (non-consecutive) MVRP Configuration

```
device(config)#int e 1/1  e 1/3 e 1/5
device(config-mif-1/1,1/3,1/5)#mvrp enable
device(config-mif-1/1,1/3,1/5)#mvrp registration-mode forbidden vlan 10
device(config-mif-1/1,1/3,1/5)#mvrp timer join 400 leave 1400 leave-all 10000
```

Error messages

Deletion of dynamically learned VLAN

Assume VLAN 10 is dynamically learned over port 1/1. Now when VLAN 10 is manually removed, the following error message is displayed.

```
Error - Dynamic VLAN 10 cannot be deleted manually.
```

Deletion of dynamically created PORT-VLAN membership

Assume VLAN 10 is dynamically learned over port 1/1. Now when port 1/1 is manually removed from the VLAN, the following error message is displayed.

```
Error - Dynamically added port 1/1 cannot be deleted from VLAN 10 manually
```

Adding a VLAN to forbidden list over MVRP enabled port when it is statically configured

For example, assume port 1/1 is statically tagged to VLAN 10. Now when VLAN 10 is added to the forbidden list on an MVRP enabled port 1/1, the following error is displayed.

```
Error - Forbidden vlan configuration not allowed as VLAN 10 is statically configured on port 1/1.
```

Enabling MVRP when per VLAN instance of STP or RSTP is running

For example, assume per VLAN STP or RSTP is running. While enabling MVRP on the system following error is displayed.

```
Error - Please remove all per vlan instances of STP and RSTP.
```

Enabling MVRP while MSTP is configured

When MVRP is enabled while MSTP is running, the following error message is displayed.

```
Error - Please remove all MSTP configurations before running MVRP.
```

Enabling MVRP when metro ring is configured

When MVRP is enabled while metro ring is configured, the following error message is displayed.

```
Error - Please remove all metro-ring configurations before running MVRP
```

Enabling MVRP when ERP is configured

When MVRP is enabled while ERP is configured, the following error message is displayed.

```
Error - Please remove all ERP configurations before running MVRP
```

Enabling MVRP when VSRP is configured

When MVRP is enabled while VSRP is configured, the following error message is displayed.

```
Error - Please remove all VSRP configurations before running MVRP
```

Enabling MVRP when topology group is configured

When MVRP is enabled while a topology group is configured, the following error message is displayed:

```
Error - Please remove all topology group configurations before running MVRP
```

Enabling MVRP when VLAN group is configured

When MVRP is enabled while VLAN group is configured, the following error message is displayed.

```
Error - Please remove all VLAN group configurations before running MVRP
```

Enabling MVRP over VPLS end point port

When MVRP is configured over a port which is also a VPLS end point, the following error message is displayed.

```
Error - MVRP cannot be enabled on port 1/1 as it is configured as a VPLS end point
```

Enabling MVRP over port with non-default port-type

When MVRP is enabled over a port with non-default port-type, the following error message is displayed.

```
Error - MVRP cannot be enabled on port 1/1 as its port-type has been changed for PB-PBB network
```

Running per VLAN instance of STP or RSTP when MVRP is enabled

For example, assume per VLAN STP or RSTP is running. Now, while enabling MVRP on the system, the following error message is displayed.

```
Error - Please disable mvrp protocol before running STP on vlan 10.
```

Enabling MSTP when MVRP is configured

When MSTP is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running MSTP on vlan 10.
```

Enabling metro ring when MVRP is configured

When metro ring is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running MRP on VLAN 10.
```

Enabling ERP when MVRP is configured.

When ERP is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running ERP on VLAN 10.
```

Enabling VSRP when MVRP is configured

When VSRP is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running VSRP on VLAN 10.
```

Enabling topology group when MVRP is configured

When a topology group is configured while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp before configuring topology group.
```

Enabling VLAN group when MVRP is configured

When vlan-group is configured while MVRP is configured, the following error message is displayed.

```
Error - Please disable MVRP before configuring vlan group.
```

Adding port to VPLS vlan when MVRP is configured over the same port

When an MVRP enabled port is added to a VPLS VLAN, the following error message is displayed.

```
Error - Port %p cannot be a VPLS end-point, due to mvrp configuration.
```

Changing port type to non-default when MVRP is configured over the same port

When a port type of MVRP enabled port is changed to non-default, the following error message is displayed.

```
Error: Cannot change port type to non-default when MVRP is enabled on the same port.
```

Adding port to VPLS vlan when MVRP is configured over the same port

When an MVRP enabled port is added to a VPLS VLAN, the following error message is displayed.

```
Error:  Please disable mvrp before running %s over any vpls vlan.
```

Removing single spanning tree instance when MVRP is enabled

When a single STP or RSTP is disabled while MVRP is running, the following error message is displayed.

```
Error - single stp cannot be disabled when MVRP is running.
Error - single rstp cannot be disabled when MVRP is running.
```

When MVRP enabled port is added as secondary port of a trunk

When an MVRP enabled port is added as a secondary port of a trunk, the following error message is displayed.

```
Error - mvrp is enabled on secondary port 1/1.
```

When a VLAN having dynamic ports is removed from base spanning tree

For example, assume statically created VLAN 11 is dynamically learned on port 1/1. Now when VLAN 11 is removed from the base spanning tree, the following error message is displayed.

```
Error - Cannot remove spanning tree from VLAN: 11 as it has dynamically tagged ports.
```

Syslog Messages

You can enable the syslog messages by using the **logging enable mvrp-vlan** command.

When a VLAN is created dynamically

```
Jan  9 03:31:42:AM: MVRP: VLAN 100 dynamically added.
```

When a VLAN is removed dynamically

```
Jan  9 03:31:42:AM: MVRP: VLAN 100 dynamically removed.
```

When a VLAN is added on a port dynamically

```
Jan  9 03:31:42:AM: MVRP: Port 1/2 dynamically added to VLAN 100.
```

When a VLAN is removed from a port dynamically

```
Jan  9 03:31:42:AM: MVRP: Port 1/2 dynamically removed from VLAN 100.
```

When a dynamic Vport changes to static

```
Jan  9 03:31:42:AM: MVRP: Port VPORT type changed to static for 1/1 and VLAN 100.
```

When a dynamic VLAN changes to static

```
Jan  9 03:31:42:AM: MVRP: Dynamic VLAN 100 changed to static.
```

When VLAN creation threshold is reached

```
Jan  9 03:31:42:AM: MVRP: System threshold reached for creation of VLANs. MVRP could not add VLAN 100 on
port 1/1.
```

Multiple MAC Registration Protocol (MMRP)

Overview

Multiple MAC Registration Protocol (MMRP) provides a mechanism for end-stations and bridges to dynamically register or declare group membership or individual MAC addresses to bridges attached in the same LAN. Any given declaration is propagated to all application participants, and registered in each bridge on those ports that are closest to the source or sources of the declaration within the active topology. Registration of group membership information makes bridges aware that frames destined for the group MAC address concerned should only be forwarded in the direction of the registered members of the group. Therefore, forwarding of frames destined for the address associated with that group occurs only on ports on which such membership registration has been received.

MMRP networks

MMRP on NetIron provides the ability for flood containment for multi-point services in a PBB network.

Limitations

MMRP is not supported on untagged ports.

Propagation of Group Membership

MMRP uses the Registration Entries in the Filtering Database to ensure that the frames are transmitted to those ports on which group members are attached, thereby avoiding flooding of these frames on all the ports.

Any node required to receive frames for this group has to declare the attribute. When a bridge is required to receive frames for a group it will declare the group, membership to all the ports in the VLAN based on the active topology, so that it sends this information to all connected nodes. Each bridge receiving this declaration will register it on the incoming port and will in turn send it on all other ports there by propagating the declaration to all the nodes in the LAN. Such propagation will result in the formation of the reachability tree. When a bridge has to send frames to this particular group, it will be sent to ports on which the registration entry exists.

Definition of MRP protocol elements

Use of MAP context

MMRP, as defined in the standard, operates within the set of VLAN Contexts that correspond to the VLANs that are supported by the VLAN Bridged Local Area Network. The MAP Context Identifier used to identify a VLAN Context shall be equal to the VID used to identify the corresponding VLAN.

The set of ports defined to be part of the active topology for a given VLAN Context shall be equal to the set of ports for which the following are true:

- The port is a member of the member set of that VLAN; and
- The port is one of the ports that are part of the active topology for the spanning tree that supports that VLAN.

Context identification in MMRP

The ingress rules for MMRPDUs on the port receiving such frames.

- MMRP frames with no VLAN classification (that is, untagged or priority-tagged MMRPDUs) are discarded if the port is tagged port in the vlan.
- VLAN-tagged MMRP frames are classified according to the VID carried in the tag header.
- If the port is not in the member set for the MMRP frame's VLAN classification, then the frame is discarded.
- MMRPDUs transmitted by MMRP Participants are VLAN classified according to the VLAN Context associated with that Participant. The following rules apply when the MMRPDUs are transmitted.
- MMRPDUs are transmitted through a given port only if the port is the member of the VLAN.
- MMRPDUs are transmitted as VLAN-tagged frames or as untagged frames in accordance with the port is tagged or untagged Port for the VLAN concerned. Where VLAN-tagged frames are transmitted, the VID field of the tag header carries the VLAN Context Identifier value.

MMRPDU

The MMRPDU frames will use the destination MAC of 00-00-00-00-00-20 and Ether-type of 88-F6 and protocol version of 0x00

MMRP PDU Forwarding

- If MMRP is not enabled globally on the system, then the MMRP PDUs will be flooded in the hardware.
- If MMRP is enabled globally, then MMRP PDUs are captured to CPU for further processing.
  - In the LP, if MMRP is enabled on the source port of the PDU, then it is forwarded to MP for further processing.
  - If MMRP is not enabled on the source port of the PDU, then it is flooded to other ports from the LP.
- Flooding of MMRP PDUs occurs on the forwarding ports of the VLAN on which the PDU is received.

MMRP Operation Overview

MMRP defines an MRP application that provides the extended filtering services. MMRP makes use of the following:

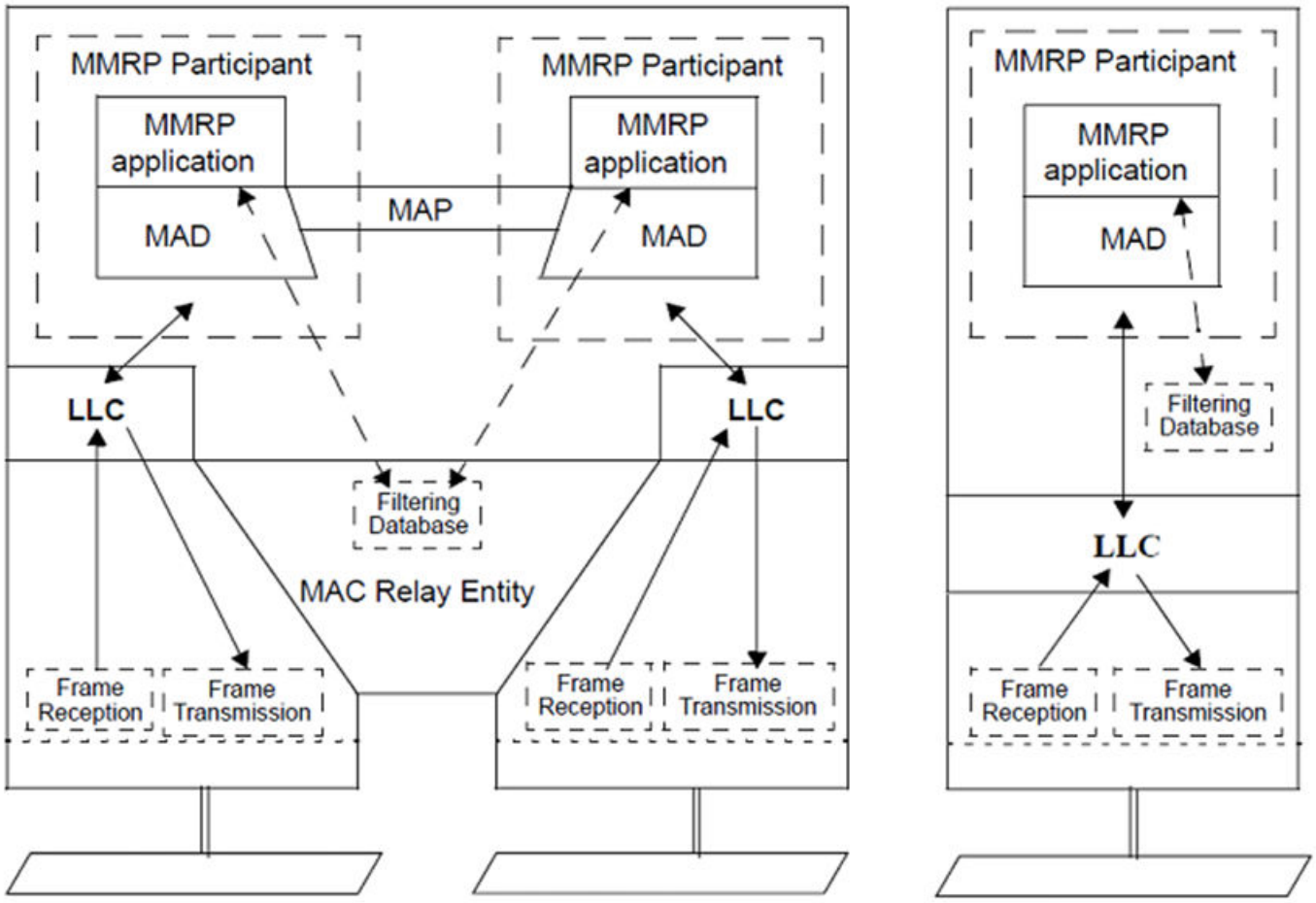
- The declaration and propagation services offered by MAD and MRP to declare and propagate Group membership, Group service requirement, and individual MAC address information within the LAN
- The registration services offered by MAD to allow Group membership, Group service requirement, and individual MAC address information to control the frame filtering behavior of participating devices.

Figure 159 illustrates the architecture of MMRP in the case of a two-Port Bridge and an end station, for a given VLAN Context. Where MMRP is used in multiple VLAN Contexts, an instance of the MMRP Participant exists for each VLAN context.

As shown in the diagram, the MMRP Participant consists of the following components:

- The MMRP application
- MRP Attribute Propagation
- MRP Attribute Declaration

FIGURE 157 MMRP components



Configuring MMRP

MMRP must be enabled globally to allow the device to participate in the MMRP (IEEE 802.1ak) protocol.

1. On any device on which you want to configure MMRP service, from privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Enable MMRP globally.

```
device(config)# mmrp enable
```

3. Configure a set of VLANs on which MMRP is allowed to participate.

```
device(config)# mmrp include-vlan 500 600 700
```

Only 256 B-VLANs are allowed to participate in MMRP. The global configuration is applicable to all the MMRP enabled ports unless an explicit configuration is made on the port.

4. Configure the join, leave, and leave-all timers at the global level.

```
device(config)# mmrp timer join 400 leave 1400 leave-all 10000
```

The Leave timer should be greater than or equal to twice the join timer plus 600ms. Leave-all timer should be large relative to the Leave timer; recommended value is at least three times the value of Leave timer.

5. Enable MMRP on the interfaces.

By default, MMRP is disabled on all interfaces. After MMRP is enabled globally, use the **mmrp enable** command on the interfaces on which MMRP is required.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# mmrp enable
```

6. (Optional) Configure a set of VLANs on which MMRP is allowed to participate on an interface.

```
device(config-if-e1000-1/1)# mmrp include-vlan 100 to 300
```

By default, if no port level configuration exists, then MMRP will operate on a globally configured include-vlan.

7. (Optional) Configure the join, leave, and leave-all timers at the interface level.

```
device(config-if-e1000-1/1)# mmrp timer join 200 leave 1500 leave-all 8000
```

8. (Optional) Configure registration mode for MACs to be forbidden.

```
device(config-if-e1000-1/1)# mmrp registration-mode vlan 100 forbidden 011E.8300.3001
```

9. Configure an interface as point-to-point for MMRP.

```
device(config-if-e1000-1/1)# mmrp point-to-point
```

10. Verify the configuration.

```
device(config-if-e1000-1/1)# end
device# show mmrp configuration
mmrp enable
mmrp include-vlan 500 600 700
mmrp timer join 400 leave 1400 leave-all  10000
!
interface ethernet 1/1
 mmrp enable
 mmrp include-vlan 100 to 300
 mmrp timer join 200 leave 1500 leave-all 8000
 mmrp registration-mode vlan 100 forbidden 011E.8300.3001
 mmrp point-to-point
!
```

11. Display the MMRP statistics.

```
device# show mmrp statistics
Vlan 100 - Ports 1/1 to 1/5
-------------------------------------
Message type   Received   Transmitted
-------------------------------------
In             0          0
Join In        0          0
Join Empty     0          0
Empty          0          156
Leave          0          0
Leave All      40         41
-------------------------------------
Total PDUs     2          826
-------------------------------------
Vlan 200 - Ports 2/1 to 2/5
-------------------------------------
Message type   Received   Transmitted
-------------------------------------
In             0          0
Join In        0          0
Join Empty     0          0
Empty          0          156
Leave          0          0
Leave All      40         41
-------------------------------------
Total PDUs     2          826
-------------------------------------
```

Clearing MMRP statistics

MMRP session counters can be cleared using a CLI command.

Ensure that MMRP is enabled in your network.

To determine the effect of clearing the MMRP statistics, an appropriate **show** command is entered before and after the **clear** command.

1. From the privileged EXEC mode, enter the **show mmrp statistics** command.

```
device# show mmrp statistics vlan 100
Vlan 100 - Ports 1/1 to 1/6
-------------------------------------
Message type   Received   Transmitted
-------------------------------------
In             0          0
Join In        0          0
Join Empty     0          0
Empty          0          156
Leave          0          0
Leave All      40         41
-------------------------------------
Total PDUs     2          826
-------------------------------------
```

2. Enter the **clear mmrp statistics** command.

```
device# clear mvrp statistics vlan-id 100
```

3. Enter the **show mvrp statistics** command for Ethernet 1/1.

```
device# show mvrp statistics ethernet 1/1
Vlan 100 - Ports 1/1 to 1/6
-------------------------------------
Message type   Received   Transmitted
-------------------------------------
In             0          0
Join In        0          0
Join Empty     0          0
Empty          0          0
Leave          0          0
Leave All      0          0
-------------------------------------
Total PDUs     0          0
-------------------------------------
```

In this show output for a specified interface after the **clear mmrp statistics** command has been entered, you can see that the statistical counters have been reset.

Syslog messages

The following syslog messages may occur when using MMRP feature.

Message `MMRP is disabled globally.`
Explanation MMRP is disabled globally.
Message Level Informational

Message `MMRP Mac 011e.8300.2710 registered on port 1/1 vlan 100.`
Explanation A new MAC is registered.
Message Level Informational

Message `MMRP Mac 011e.8300.2710 is removed from port 1/1 vlan 100.`
Explanation MMRP MAC is removed.
Message Level Informational

CLI Error Messages

The following Error messages are introduced for this feature

1. Timer configuration: Leave timer should be greater than or equal to twice the join timer plus 600ms. If this condition is not satisfied then the following error message will be displayed.

```
Error: leave timer value must be greater than twice the join timer plus 600ms
```

2. Timer configuration: Leave-all timer should be large relative to leave timer; recommended value is at least three times the value of leave timer.

```
Error: leave-all timer value must be greater than three times the leave timer value
```

3. If MMRP is disabled globally any MMRP command is issued then the following error will be displayed.

```
Error: MMRP is disabled globally, operation rejected.
```

4. If MMRP is enabled on non-existing VLAN then following error will be displayed.

```
Error: Vlan <vlan id> not configured
```

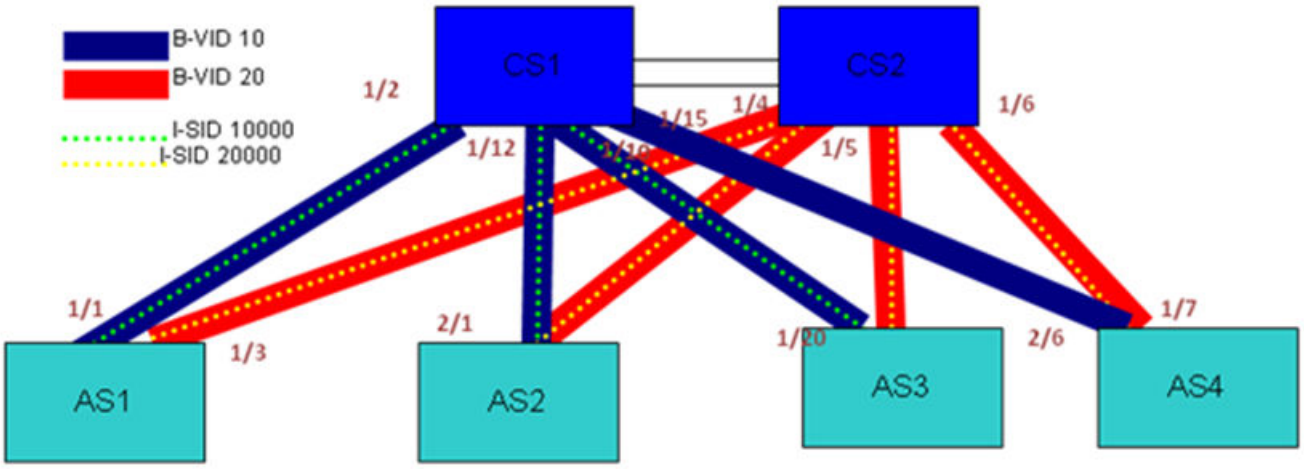
5. If at the interface level if MMRP is enabled on the VLAN is which is not present in the global configuration then following error message will be displayed.

```
device#(config)# mrp enable
device(config)# mmrp include-vlan 10
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# mmrp include-vlan 800
Error - mmrp vlan 800  must be configured at global level first
```

Configuration Example

In the network shown in Figure 160 there are two E-LANS one for B-VLAN10 and ISID 10000 and other of B-VLAN20 and ISID 20000.

FIGURE 158 MMRP with PBB enabled



Sample MMRP configuration on AS1

```
device_AS1(config)# mmrp enable
device_AS1(config)# mmrp include-vlan 10 20
device_AS1(config)# interface ethernet 1/1 ethernet 1/3
device_AS1(config-mif-1/1,1/3)# mmrp enable
device_AS1(config-mif-1/1,1/3)# mmrp include-vlan 10
```

Sample configuration on AS2

```
device_AS2(config)# mmrp enable
device_AS2(config)# mmrp include-vlan 10 20
device_AS2(config)#interface ethernet 2/1 ethernet 2/3
device_AS2(config-mif-2/1,2/3)# mmrp enable
device_AS2(config-mif-2/1,2/3)# mmrp include-vlan 10
```

PBB configuration MLX for B-VLAN10

```
device_AS1(config)# vlan 10
device_AS1(config-vlan-10)# tagged ethernet 1/1
```

```
device_AS1(config-vlan-10)# exit
device_AS1(config)# router mpls
device_AS1(config-mpls)# vpls vinst 2000
device_AS1(config-mpls-vpls-vinst)# pbb
device_AS1(config-mpls-vpls-vinst-pbb)# vlan 10 isid 10000
device_AS1(config-mpls-vpls-vinst-vlan-10-isid-10000)# tagged ethernet 1/1
```

PBB configuration on Metro for B-VLAN10

```
device_AS1(config)# interface ethernet 1/1
device_AS1(config-if-e10000-1/1)# port-type backbone-network
device_AS1(config-if-e10000-1/1)# exit
device_AS1(config)# esi iptv-service encapsulation isid
device_AS1(config-esi-iptv-service)# isid 10000
device_AS1(config-esi-iptv-service-isid-10000)# exit
device_AS1(config)# esi iptv-carrier encapsulation bvlan
device_AS1(config-esi-iptv-carrier)# vlan 10
device_AS1(config-esi-iptv-carrier-vlan-10)# tagged ethernet 1/1
device_AS1(config-esi-iptv-carrier-vlan-10)# esi-client  iptv-service
```

Sample configuration on CS1

```
device_CS1(config)# mmrp enable
device_CS1(config)# mmrp include-vlan 10 20
device_CS1(config)# interface ethernet 1/2 ethernet 1/10 ethernet 1/12 ethernet 1/15
device_CS1(config-mif-1/2, 1/10, 1/12, 1/15)# mmrp enable
```

PBB configuration MLX for B-VLAN 10

```
device_CS1(config)# vlan 10
device_CS1(config-vlan-10)# tagged ethernet 1/2 ethernet 1/10 ethernet 1/12 ethernet 1/15
device_CS1(config)#
```

PBB configuration on Metro for B-VLAN 10

```
device_CS1(config)# interface ethernet 1/2 ethernet 1/10 ethernet 1/12 ethernet 1/15
device_CS1(config-if-e10000-1/1)# port-type backbone-network
device_CS1(config-if-e10000-1/1)# exit
device_CS1(config)# esi iptv-carrier encapsulation bvlan
device_CS1(config-esi-iptv-carrier)# vlan 10
device_CS1(config-esi-iptv-carrier-vlan-10)# tagged ethernet 1/2 ethernet 1/10 ethernet 1/12 ethernet 1/15
```

MMRP is used with PBB for the registration and declaration of Multicast B-DA MAC.

Declaration of MAC

The declaration of the multicast MAC is done by the BEB. In the topology described above, the declaration is sent if AS1 is

- CER 2000 Series and CES 2000 Series
  - when an ISID ESI IPTV-service is added as client to the B-VLAN ESI IPTV-carrier
  - when MMRP is enabled on the port belonging to a B-VLAN with ISID clients associated
  - when a port enabled with MMRP is tagged to B-VLAN with ISID clients associated
- MLX Series and XMR Series
  - when B-VLAN 10 ISID 10000 endpoint is created in the VPLS instance
  - when MMRP is enabled on the port where B-VLAN and ISID is configured

When MMRP sends Join PDU on ports in the B-VLAN. The attribute declared here would be the multicast flood MAC (011e. 8300.2710).

Displaying the declared MAC on AS1

```
device_AS1# show mmrp attributes
Port        Vlan        Mac-address                      Registrar
Registrar                    Applicant
                                    State                  Mgmt                  State
-------------------------------------------------------------------------------
1/1         10        011e.8300.2710           IN                              Fixed
Quiet Active
```

Registration of MAC

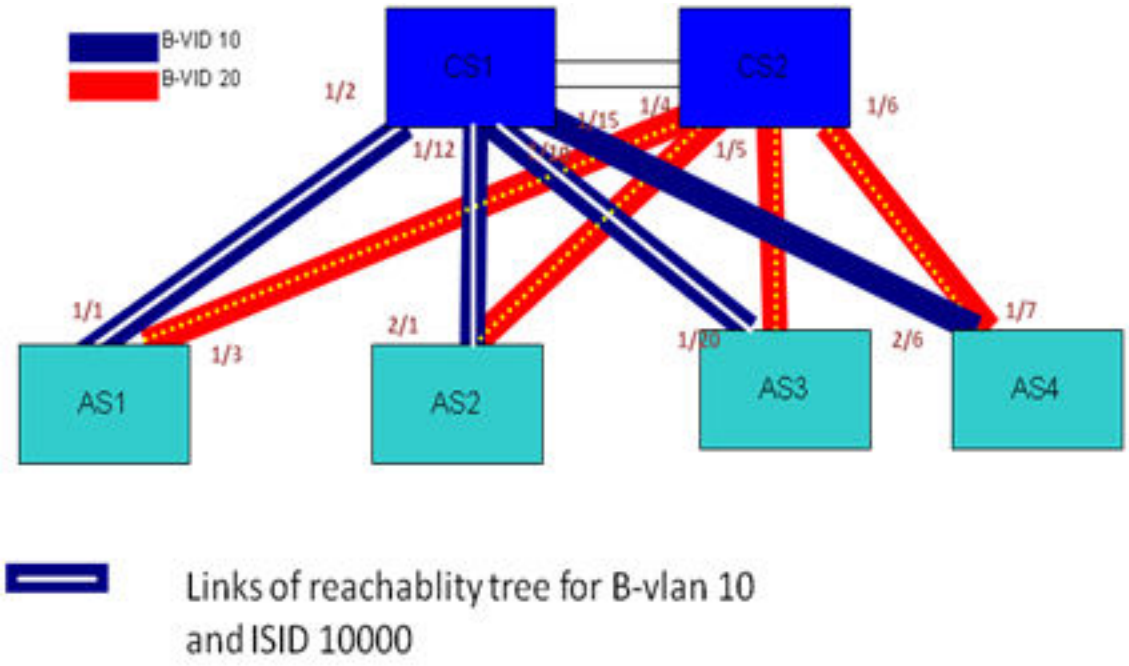
CS1 receives this declaration on port 1/2 and will register it. It will then propagate the declaration to all the ports of the B-VLAN that are in the forwarding state.

This dissemination will result in the registration of this MAC on AS2, AS3, and AS4 and reachability tree is formed. Similarly AS2, AS3 also declare for the flood MAC. AS4 will not declare because the I-SID is not terminated on AS4. CS1 now has ports connecting to AS1, AS2, and AS3 registered to the flood MAC for B-VLAN10. The Figure 161 shows the reachability tree for the B-VLAN10.

Displaying the Registered MAC on CS1

```
device_CS1#show mmrp attributes
Port        Vlan         Mac-address               Registrar
Registrar                       Applicant
                                    State                 Mgmt                      State
-------------------------------------------------------------------------------
1/2         10        011e.8300.2710      IN                              Normal
Quiet Active
1/10         10        011e.8300.2710      IN                               Normal
Quiet Active
1/12         10         011e.8300.2710      IN                               Normal
Quiet Active
```

FIGURE 159 Reachability tree for B-VLAN 10 and ISID 10000



When packets with this MAC address reach the CS, it will multicast only on ports on which registration of the MAC address is present (for example: it will multicast to AS1, AS2 and AS3 but not to AS4).

Unknown Unicast Multicasting in the above Topology:

1. AS2 is trying to send to a customer MAC C.
2. AS2 will flood the packet towards on port 2/1 the PBB network with the B-SA as the MS2 (Base MAC of AS2) and B-DA as Multicast flood MAC, B-VLAN as 10 and ISID as 10000.
3. CS1 on receiving this packet will multicast to the ports on which it has registered the flood MAC, in this case 1/2 and 1/10 (it will not send on 1/12 because of source port suppression).
4. If AS1 has been learnt the customer MAC C then it will strip the PBB header do Layer 2 forwarding towards the customer.

Remote Fault Notification (RFN)

10G WAN PHY fault and performance management

This feature provides fault and performance management features such as alarm detection, alarm generation, and performance monitoring on 10 GbE WAN PHY interfaces. It only applies to 10 GbE interfaces configured in the WAN PHY mode.

Using this feature, you can gather fault and performance management information and display it for the current 15 minute interval or for any of the previous 15 minute intervals. In addition, this feature allows you to create a path trace between WAN PHY interfaces to ensure correct connection.

- 8x10G WAN PHY has been supported in releases before 5.8.00b.
- 20x10G WAN PHY support is available starting with 5.8.00b.

Setting a 10 GbE interface to WAN PHY mode

To set a 10 GbE interface to WAN PHY mode, use the following command.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# phy-mode wan
```

Syntax: [no] phy-mode [**wan** | **28k**]

The **wan** parameter sets the PHY mode to WAN.

The **28k** parameter sets the PHY mode to 28k (to allow interoperability with other devices).

The default setting is LAN PHY mode; to reset PHY mode to LAN, use the command **no phy-mode** .

Turning alarm interfaces on and off

When a 10 GbE port is to WAN PHY mode, alarm monitoring is set on by default. You can turn alarm monitoring off for an individual port as shown in the following.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# no alarm-monitoring
```

Syntax: [no] alarm-monitoring

Configuring path trace

You can configure a character string to be carried in the SONET overhead as a method of detecting mis-connection of ports between two devices connected over the WAN PHY. The devices compare the configured character string with the received character string to determine if the connection is valid.

You can configure the Extreme device with the character string "test1" using the following command.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# overhead j0-transmit test1
```

Syntax: [no] overhead [**j0-transmit string**] | [**j1-transmit string**]

The *string* variable is the character string used to detect mis-connection of ports it must be configured with the same value on each side of the WAN PHY connection.

Displaying status of alarms on an interface

You can display the current status of WAN PHY alarms as shown in the following.

```
device# show controller curr15min e 3/1
----------------------------------------------------------
10g wan phy alarms statistics  - PORT e3/1
----------------------------------------------------------
ACTIVE ALAMRS : LOS
ACTIVE DEFECTS : LOF-S AIS-L AIS-P
Elapsed time [0 min 13 secs]
Format [alarm type = count]
FM-PARAMS
 Section
  LOS = 1  LOF = 1
Line
  AIS-L = 1 RDI-L = 0
Path
  AIS-P = 1  LOP = 0 PLM = 0 AIS-PFE = 0 PLM-PFE = 0
PM-PARAMS
 Section
  CV = 2  ES = 1  SES = 0  SEFS = 0
 Line
  CV = 3  ES = 1  SES = 0 UAS = 0
  CV-FE  = 0  ES-FE = 0  SES-FE = 0 UAS-FE = 0
 Path
   CV = 4  ES = 1  SES = 0 UAS = 0
   CV-FE  = 0  ES-FE = 0  SES-FE = 0 UAS-FE = 0
```

Syntax: show controller [**curr15min port no** | **slot no**] | [**day port no** | **slot no**] | [**prev15min interval port no** | **slot no**]

The **curr15min** parameter specifies that you want to display WAN PHY alarm and performance information for the current 15 minute interval for either the port number *portno* or slot number *slot no* specified.

The **day** parameter specifies that you want to display WAN PHY alarm and performance information for the current day for either the port number *port no* or slot number *slot no* specified.

The **prev15min** parameter specifies that you want to display WAN PHY alarm and performance information for the a past 15 minute interval as specified by the *interval* variable for either the port number *port no* or slot number *slot no* specified. Possible values for *interval* are 1 - 31 and indicates which previous 15 minute interval you want to display information from. The closest previous interval is 1 and the farthest is 31.

The *port no* variable specifies the port that you want to display WAN PHY alarm and performance information for.

The *slot no* variable specifies the slot that you want to display WAN PHY alarm and performance information for.

This display shows the following information.

TABLE 51 WAN PHY display parameters

Parameter	Description.
Loss of signal (LOS)	LOS is raised when the synchronous signal (STS-N) level drops below the threshold at which a BER of 1 in 103 is predicted. It could be due to a cut cable, excessive attenuation of the signal, or equipment fault. LOS state clears when two consecutive framing patterns are received and no new LOS condition is detected.
Out of frame (OOF) alignment or SEF (Severely errored Frame)	OOF state occurs when four or five consecutive SONET frames are received with invalid (errored) framing patterns (A1 and A2 bytes). The maximum time to detect OOF is 625 microseconds. OOF state clears when two consecutive SONET frames are received with valid framing patterns.
Loss of frame (LOF) alignment	LOF state occurs when the OOF state exists for a specified time in milliseconds. LOF state clears when an in-frame condition exists continuously for a specified time in milliseconds.
Loss of pointer (LOP)	LOP state occurs when N consecutive invalid pointers are received or N consecutive new data flags (NDFs) are received (other than in a concatenation indicator), where N = 8, 9, or 10. LOP state clears when three equal valid pointers or three consecutive AIS indications are received. LOP can be identified as follows: • STS path loss of pointer (SP-LOP) • VT path loss of pointer (VP-LOP)
Alarm indication signal (AIS)	The AIS is an all-ones characteristic or adapted information signal. It is generated to replace the normal traffic signal when it contains a defect condition in order to prevent consequential downstream failures being declared or alarms being raised. Line AIS defect is detected as a "111" pattern in bits 6, 7, and 8 of the K2 byte in five consecutive frames. Line AIS defect is terminated when bits 6, 7, and 8 of the K2 byte do not contain the code "111" for five consecutive frames. STS-Path AIS defect is detected as all ones in bytes H1 and H2 in three contiguous frames. STS-Path AIS defect is terminated when a valid STS Pointer is detected with the NDF set to "1001" (inverted) for one frame, or "0110" (normal) for three contiguous frames. AIS can also be identified as follows: • Line alarm indication signal (AIS-L) • STS path alarm indication signal (SP-AIS) • VT path alarm indication signal (VP-AIS)
Remote error indication (REI)	This is an indication returned to a transmitting node (source) that an errored block has been detected at the receiving node (sink). This indication was formerly known as far end block error (FEBE). REI can also be identified as the following: • Line remote error indication (REI-L) • STS path remote error indication (REI-P) • VT path remote error indication (REI-V)
Remote defect indication (RDI)	This is a signal returned to the transmitting terminating equipment upon detecting a loss of signal, loss of frame, or AIS defect. RDI was previously known as FERF. RDI can also be identified as the following: • Line remote defect indication (RDI-L)

TABLE 51 WAN PHY display parameters (continued)

Parameter	Description.
	• STS path remote defect indication (RDI-P) • VT path remote defect indication (RDI-V)
B1 error (coding violation, CV)	Parity errors evaluated by byte B1 (BIP-8) of an STS-N are monitored. If any of the eight parity checks fail, the corresponding block is assumed to be in error.
B2 error (coding violation, CV)	Parity errors evaluated by byte B2 (BIP-24 x N) of an STS-N are monitored. If any of the N x 24 parity checks fail, the corresponding block is assumed to be in error.
B3 error (coding violation, CV)	Parity errors evaluated by byte B3 (BIP-8) of a VT-N (N = 3, 4) are monitored. If any of the eight parity checks fail, the corresponding block is assumed to be in error.
Errored Seconds (ES)	At each layer, an Errored Second (ES) is a second with one or more Coding Violations at that layer OR one or more incoming defects (e.g., SEF, LOS, AIS, LOP) at that layer has occurred. Far end - This is an indication returned to a transmitting node (source) that an errored block has been detected at the receiving node (sink). And Errored seconds - far end indicate this error in terms of errored seconds. ES can be identified as follows: • Section Errored seconds (ES-S)· • Line Errored seconds (ES-L), Line Errored seconds- Far end (ES-LFE)· • Path Errored seconds (ES-P), Path Errored seconds- Far end (ES-PFE)
Severely Errored seconds (SES)	At each layer, an Severely Errored Second (SES) is a second with x or more CVs at that layer, or a second during which at least one or more incoming defects at that layer has occurred. Values of x vary depending on the line rate and the Bit Error Rate. SES can be identified as follows: • Section Severely Errored seconds (SES-S)· • Line Severely Errored seconds (SES-L), Line Errored seconds-Far end (SES-LFE)· • Path Severely Errored seconds (SES-P), Path Errored seconds-Far end (SES-PFE)
Severely errored frame seconds (SEFS)	A Severely Errored Framing Second (SEFS) is a seconds with containing one or more SEF events. This counter is only counted at the Section Layer.
Unavailable seconds (UAS)	At the Line, Path, and VT layers, an unavailable second is calculated by counting the number of seconds that the interface is unavailable. At each layer, the SONET or SDH interface is said to be unavailable at the onset of 10 contiguous SESs. The 10 SESs are included in unavailable time. Once unavailable, the SONET or SDH interface becomes available at the onset of 10 contiguous seconds with no SESs. The 10 seconds with no SESs are excluded from unavailable time. With respect to the SONET or SDH error counts at each layer, all counters at that layer are incriminated while the SONET or SDH interface is deemed available at that layer. While the interface is deemed unavailable at that layer, the only count that is incriminated is UASs at that layer. UAS can be identified as follows: • Line Unavailable seconds (UAS-L), Line Unavailable seconds at far end (UAS-LFE)· • Path Unavailable seconds (UAS-P), Path Unavailable seconds (UAS-PFE)

Wait for all cards feature

During a system reload, an Interface module comes up after it completes its initialization process. After an Interface module is up, its ports can come up. Since 10G modules have more packet processors to initialize, 1G ports are up earlier than 10G ports.

> NOTE
> Rebooting interface modules manually is not supported. The wait for all cards feature will only take effect when the entire router or switch is rebooted.

The **wait-for-all-cards** command directs all ports to come up at the same time. This is done by waiting for all Interface modules to come up first, before allowing for ports to come up. This command is shown in the following.

```
device(config)# wait-for-all-cards
```

Syntax: [no] wait-for-all-cards

> NOTE
> With the **wait-for-all-cards** command enabled,10G ports will come up before 1G ports because NetIron software processes 10G port's state changes first.

Link fault signaling

You can enable link fault signaling on 10 or 100 gigabit interfaces. Link fault signaling (LFS) is a physical layer protocol that enables communication on a link between two 10 or 100 Gigabit Ethernet devices. When configured on the Extreme 10 or 100 Gigabit Ethernet port, the port can detect and report fault conditions on transmit and receive ports.

If LFS is configured on an interface, the following Syslog messages are generated when that interface goes up or down or when the TX or RX fiber is removed from one or both sides of the link that has LFS configured:

- SYSTEM: port 2/1 is down (remote fault)
- SYSTEM: Interface ethernet 2/1, state down - remote fault
- SYSTEM: Interface ethernet 2/1, state up

Traditionally, in MLX Series and XMR Series devices, LFS was disabled in both TX and RX directions. The **link-fault-signaling** command was used to enable LFS in both TX and RX directions. When RX LFS is enabled, a port will be brought up only when the PHY-MAC link is up, and there is no link fault received by the MAC. When RX LFS is disabled, a port will be brought up as long as the PHY-MAC link is up, regardless of any RX fault indication to MAC.

The RX LFS is always enabled by default and cannot be disabled. The **link-fault-signaling** command only applies to enabling or disabling the TX LFS. While RX LFS is recommended to be enabled at all times, for some applications it is requested to have the means to disable RX LFS.

There are two independent link-fault signaling commands **link-fault-signaling** and **link-fault-signaling ignore-rx.** These commands are applicable at both the global (system-level) and per-port level. Both global and per-port configurations are considered jointly to determine the resulting per-port configuration. When a global configuration is applied, it will override the corresponding per-port configuration already present. It is recommended to configure the global configuration prior to applying per-port configurations.

To configure LFS, enter the following commands.

```
device(config)# interface ethernet 1/4
device(config-if-e1000-1/4)# link-fault-signaling
```

Syntax: [no] link-fault-signaling

LFS is disabled by default.

NOTE
Ensure both sides are LFS ON when using LFS with RX (always on) and another router (which can be configured ON or OFF). Do not assume all boxes have LFS ON or OFF by default. Be sure and check.

To to disable RX LFS on a specified port, enter the **link-fault-signaling ignore-rx** command.

```
device(config)# interface ethernet 1/4
device(config-if-e1000-1/4)# link-fault-signaling ignore-rx
```

Syntax: [no] link-fault-signaling ignore-rx

RX LFS is ignored on the specified port.

Displaying and clearing remote fault counters

To display Remote Fault Notification (RFN) counters on 10GbE LAN physical interface, enter the following command.

```
device # show remote-fault ethernet 1/1 to 1/4
Port RFN Detected Remote-fault count time last RFN detected
---- ---------------- ---------------------- --------------------------
1/1 Yes 15 Sep 29 22:03:03
1/2 No 0 -
1/3 No 12 Aug 20 13:22:14
1/4 No 0 -
    ** remote-fault counters are only supported for ports in LAN PHY mode on 10GE modules. **
```

The example above displays remote fault notification counters with slot 1 as a 10GbE module, and ports 1/1, 1/2, 1/3, and 1/4 in LAN mode.

If the user enters a slot number that is not a 10 GbE port, or if any port in the port range is not a 10GbE port in LAN mode, the following error message is displayed.

```
device# show remote-fault slot 3
remote-fault counters are only supported for ports in LAN PHY mode on 10 GE modules.
```

To clear remote fault notification counters on a 10 GbE LAN physical interface, enter the following command.

```
device#clear remote-fault slot 1
```

Syntax: show or clear remote-fault [**ethernet slot#/port#** [**to slot#/port**] | **slot slot#**]

You can display information for remote fault notification counters in the Extreme device by using the **show remote-fault** command without options,

Use the ethernet *<slot#/port#>* option to limit the display to a single ethernet port.

Use the to *<slot#/port>* option for a range of ports.

Use the slot *<slot#>* option to limit the display to a single slot.

The following table describes the output of the **show remote-fault** command

TABLE 52 Display of show remote-fault output

This field...	Displays...
Port	The *<port#>* variable specifies the port number for the interface module.
RFN Detected	The remote-fault notification is detected on a given interface. If "Yes" is displayed, then the remote-fault notification is detected on the given port at the time of inquiry. If "No" is displayed, then no remote-fault notification is detected on the given port at the time of inquiry.
Remote-fault count	The Remote-fault count displays the number of times the remote-fault notification is detected on a given interface. The number of times, include: • The time since the Interface Module was last powered on. • The time since the count was last cleared by the user. • The time since the interface was last configured as a LAN mode.
time last RFN detected	The time the remote-fault notification was last detected on a given interface.

Reverse Path Forwarding

A number of common types of denial-of-service (DoS) attacks, including Smurf and Tribe Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. Reverse Path Forwarding (RPF) is designed to prevent such a malicious user from spoofing a source IP address by checking that the source address specified for a packet is received from a network to which the device has access. Packets with invalid source addresses are not forwarded. Optionally, you can log packets that fail the RPF test.

RPF is supported for IPv6 packets. Differences in RPF support in IPv4 and IPv6 are noted within this chapter where necessary.

RPF configuration

Before you begin to configure Reverse Path Forwarding (RPF), review the following sections:

- Configuration considerations for RPF on page 551
- Special considerations for configuring RPF on CES 2000 Series and CER 2000 Series devices on page 552
- Special considerations for configuring RPF with ECMP routes on page 552
- RPF support for IP over MPLS routes on page 552
- RPF-compatible CAM profiles on page 552
- Configuring the global RPF command on page 553
- Enabling RPF on individual ports on page 553
- Configuring a timer interval for IPv6 session logging on page 554
- Suppressing RPF for packets with specified address prefixes on page 554

Configuration considerations for RPF

Consider the following points when you configure Reverse Path Forwarding:

- IP packets with a source IP address of 0.0.0.0 will always fail RPF check.
- If you attempt to enable the global RPF command on a system with incompatible CAM settings, the command will be rejected and you will receive a console message.
- Because the RPF feature requires that the entire IP route table is available in hardware, the feature must work in conjunction with Foundry Direct Routing (FDR). FDR is the default mode of operation for the device.
- You cannot configure RPF on a physical port that has VRF configured on it, or if the physical port belongs to a virtual interface with a VRF configuration.
- Only RPF loose mode is supported for GRE routes.
- If a default route is present on the router, loose mode will permit all traffic.
- RPF can only be configured at the physical port level. It should not be configured on virtual interfaces on the MLX Series and XMR Series.
- CER 2000 Series and CES 2000 Series devices provide support for uRPF for VE interfaces.
- IPv6 packets with a link-local source address are not subject to IPv6 RPF check.

- IPv6 RPF check is not supported for 6-to-4 tunnel routes.

Special considerations for configuring RPF on CES 2000 Series and CER 2000 Series devices

- CES 2000 Series and CER 2000 Series devices do not support IPv6.
- Unlike the XMR Series and MLX Series devices, port level granularity is not supported on CES 2000 Series and CER 2000 Series devices; therefore, RPF must be configured on the entire device either all in loose mode or all in strict mode.
- If the logging feature is enabled, it is enabled on the entire device. If the logging feature is disabled, logging for the entire device is disabled.
- You cannot configure RPF on a physical port that belongs to a virtual LAN (VLAN).
- If a combination of RPF, PBR, and ACL are configured on an interface, RPF takes precedence.
- The section Special considerations for configuring RPF with ECMP routes on page 552 does not apply to the CES 2000 Series and CER 2000 Series devices.

Special considerations for configuring RPF with ECMP routes

NOTE
This section applies only to the XMR Series and MLX Series devices.

RPF for IPv6 is not subject to the special considerations for configuring RPF with ECMP routes described in this section.

For a source IP address matching an ECMP route, RPF permits the packet if it arrives on any of the next-hop interfaces for that route. For example, if there are two best next hops for a network route 10.11.11.0/24, one pointing to 10.10.10.1 (Gigabit Ethernet 7/1) and the other to 10.10.30.1 (Gigabit Ethernet 7/12), then incoming packets with a source address matching 10.11.11.0/24 will be permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.

A disadvantage of this configuration is that if some other route shares any of these next hops, the packets with a source IP address matching that route are also permitted from any of the interfaces associated with those next hops. For example, if 10.12.12.0/24 has the next hop 10.10.10.1, then packets from 10.12.12.0/24 are also permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.

RPF support for IP over MPLS routes

For IPv4 routes over MPLS tunnels, the physical interface for an outgoing tunnel on which a route is assigned may not be the same as the one from which you receive packets. Consequently, only RPF loose mode is supported on MPLS uplinks. IPv6 is not currently supported over MPLS. When it is supported, it will only support RPF loose mode on MPLS uplinks.

RPF-compatible CAM profiles

NOTE
This section applies only to the XMR Series and MLX Series devices.

Not all CAM profiles are compatible with RPF. Table 53 lists all of the RPF-compatible CAM profiles by software release. Refer to "CAM partition profiles" for a description of each of the available CAM profiles.

TABLE 53 RPF compatible and non-compatible CAM profiles

Software release	Compatible CAM profiles	Non-compatible CAM profiles
XMR or MLX	default	ipv4-ipv6
	ipv4	ipv4-vpls
	ipv4-vpn	l2-metro
	ipv6	l2-metro-2
	mpls-l3vpn	mpls-vpls
	mpls-l3vpn-2	mpls-vpls-2
	ipv4-ipv6-2	mpls-vpn-vpls
	multi-service-2	multi-service
	multi-service-4	

Configuring the global RPF command

Before you can enable RPF to operate on a device, you must first configure RPF globally. There are separate commands for IPv4 and IPv6, as shown in the following examples.

> **NOTE**
> IPv6 configurations are not supported on CES 2000 Series and CER 2000 Series devices.

For IPv6 configurations, use the following command.

Syntax: ipv6 reverse-path-check

```
device(config-if-e1000-1/4)# rpf-mode ?
  loose    Allow packets forwarding if there is a route to source
  strict   Allow packets forwarding if route to source is towards the incoming
           port
device(config-if-e1000-1/4)# rpf-mode strict ?
  log   Log packets that fail RPF check and are to be dropped
device(config-if-e1000-1/4)# rpf-mode strict
```

For IPv4 configurations, use the following command.

```
device(config)# reverse-path-check
```

Syntax: reverse-path-check

```
device(config)# ipv6 reverse-path-check
```

Enabling RPF on individual ports

After RPF has been configured globally for a device, it must be configured on every interface that you want it to operate. The RPF feature can be configured on physical Ethernet interfaces. There are two modes, "strict" and "loose," that can be configured to enforce RPF on IP addresses for packets arriving on a given interface:

- In **loose** mode, RPF permits a packet as long as the source address matches a known route entry in the routing table. It will drop a packet if it does not match a route entry. Note that if a default route is present, loose mode will permit all traffic.

- In **strict** mode, RPF requires that a packet matches a known route entry as described in loose mode and also that it arrives at the interface as described in the router table's next hop information. It will drop a packet that does not match both of these criteria.

Configuring RPF on a port requires separate commands for IPv4 and IPv6. To configure RPF on a port, use the IPv4 or IPv6 command, as shown in the following examples.

For IPv4 configurations, use the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e1000-3/1)# rpf-mode strict log
```

Syntax: [no] rpf-mode [**loose** | **strict**] [**log**]

For IPv6, use the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e1000-3/1)# rpf-mode-ipv6 strict log
```

Syntax: [no] rpf-mode-ipv6 [**loose** | **strict**] [**log**]

There are two modes in which you can enforce RPF on IP sources address for packets that arrive on a configured interface:

- **The loose** option configures RPF in the loose mode.
- The **strict** option configures RPF in the strict mode.

The **log** option directs RPF to log packets that fail the RPF test. Enabling RPF logging may lead to high CPU utilization on the interface module because packets that fail the RPF check test are dropped in software. Only syslog entries are created by this option. No SNMP traps are issued by this option.

The ACL or RPF logging mechanism on the interface modules log a maximum of 256 messages per minute, and send these messages to the management module. A rate-limiting mechanism has been added to rate-limit the number of messages from the interface module CPU to the management module CPU to 5 messages per second. Because this delays the delivery of messages to the management module, in the worst case scenario with all 256 packets arriving at the same time on the interface module, the time values stamped by the management module on the messages will vary by as much as 60 seconds.

Configuring a timer interval for IPv6 session logging

You can use the **ipv6 session-logging-age** command to globally configure a timer interval for IPv6 session logging. The timer interval is set for 3 minutes in the following example.

```
device(config)# ipv6 session-logging-age 3
```

Syntax: [no] ipv6 session-logging-age minutes

The *minutes* variable sets the timer interval for logging. Configurable values are from 1 through 10 minutes. The default value is 5 minutes.

You can use the **show log** command to view RPF messages, as shown in the following example.

```
device# show log
Dec 18 19:32:52:I:IPv6 RPF: Denied 1 packet(s) on port 1/2 tcp fec0:1::2(0) -> 4500:1::2(0)
```

Suppressing RPF for packets with specified address prefixes

NOTE

This section is not applicable for the CES 2000 Series and CER 2000 Series devices because, with these devices, RPF takes precedence over PBR and ACLs.

You can suppress RPF packet drops for a specified set of packets using inbound ACLs. To suppress RPF packets:

1. Create an IPv4 or IPv6 ACL that identifies the address range that you do not want dropped.
2. Specify the flag to the ACL permit clause of the **suppress-rpf-drop** command.

   When a packet that fails the RPF check and matches the specified ACL permit clause with the suppress-rpf-drop flag set, it is forwarded as a normal packet and it is accounted as a "**unicast RPF suppressed drop packet,**" as described in Displaying RPF statistics on page 556.

   NOTE
   The **suppress-rpf-drop** command is not supported on CES 2000 Series and CER 2000 Series devices.

   The following example demonstrates the configuration of the IPv4 ACL named **"access-list 135"** which permits traffic from the source network 10.4.4.0/24 even if the RPF check test fails.

```
device(config)# access-list 135 permit ip 10.4.4.0.0.0.255 any suppress-rpf-drop
device(config)# access-list 135 permit ip any any
```

   The following example demonstrates the configuration of the IPv6 ACL named "**rpf1"** which permits traffic from the source host 2002::1 even if the RPF check test fails.

```
device(config)# ipv6 access-list rpf1
device(config-ipv6-access-list rpf1)# permit tcp host 2002::1 any suppress-rpf-drop
```

   Syntax: suppress-rpf-drop

   In the following example, the IPv4 ACL 135 is applied as an inbound filter on Ethernet interface 7/5.

```
device(config)# interface ethernet 7/5
device(config-if-e1000-7/5)# rpf-mode strict
device(config-if-e1000-3/1)# ip access-group 135 in
```

   NOTE
   If the physical port is a member of a virtual interface, the ACL will have to be applied to the virtual interface instead of the physical port.

Excluding packets that match the routers default route

The **urpf-exclude-default** and **ipv6 urpf-exclude-default** commands direct the Extreme router to drop packets whose source address matches the routers default route and increment the RPF drop counter. Using this feature requires that RPF be configured globally first. This feature is configured separately for IPv4 and IPv6 as described in the following examples.

For IPv4, use the following commands.

```
device(config)# reverse-path-check
device(config)# urpf-exclude-default
```

Syntax: urpf-exclude-default

For IPv6, use the following commands.

```
device(config)# ipv6 reverse-path-check
device(config)# ipv6 urpf-exclude-default
```

Syntax: ipv6 urpf-exclude-default

Displaying RPF statistics

To display information about RPF configuration and packets that have been dropped because they failed the RPF check, use the **show ip interface** or the **show ipv6 interface** command as shown.

For IPv4, use the following command.

```
device# show ip interface ethernet 7/1
Interface Ethernet 7/1 (384)
    port enabled
    port state: UP
    ip address: 10.2.3.4/8
    Port belongs to VRF: default
    encapsulation: Ethernet, mtu: 1500
    MAC Address 000c.db24.a6c0
    directed-broadcast-forwarding: disabled
    No inbound ip access-list is set
    No outbound ip access-list is set
    No Helper Addresses are configured
    RPF mode: strict RFP Log: Disabled
    376720 unicast RPF drop 36068 unicast RPF suppressed drop
```

For IPv6 configurations, use the following command.

```
device#show ipv6 interface ethernet 3/1
Interface Ethernet  3/1 is down, line protocol is down
  IPv6 is enabled, link-local address is
  Global unicast address(es):
  Joined group address(es):
    ff02::2
    ff02::1
  MTU is 1500 bytes
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30 seconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1 seconds
  ND advertised retransmit interval is 0 seconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  No Inbound Access List Set
  No Outbound Access List Set
  IPv6 RPF mode: Strict  IPv6 RPF Log: Enabled
  RxPkts:       0                     TxPkts:   0
  RxBytes:      0                     TxBytes:  0
  IPv6 unicast RPF drop: 0
  IPv6 unicast RPF suppressed drop: 0
```

NOTE

The RPF accounting information is always available through the physical interface, even if the physical port belongs to one or more VE's

Table 54 describes the RPF statistics displayed when using the **show ip interface** or **show ipv4 interface** command. They are displayed in **bold font** .

TABLE 54 RPF statistics by port

Field	Description
RPF mode:	This display parameter can have one of the following two values: • loose - RPF will permit a packet as long as the source address matches a known route entry in the routing table. It will drop a packet if it does not match a route entry. • strict - RPF requires that a packet matches a known route entry as described for loose mode and also that it arrives at the configured interface as described in the router table's next hop

TABLE 54 RPF statistics by port (continued)

Field	Description
	information. It will drop a packet that does not match both of these criteria.
RPF Log:	This display parameter displays the RPF Log Configuration Status: • Enabled - The RPF log feature has been configured. • Disabled - The RPF log feature has not been configured
number unicast RPF drop	The number of packets that have been dropped due to failure of the RPF test.
number unicast RPF suppressed drop	The number of packets that would have been dropped due to failure of the RPF test but were not dropped because they matched conditions set in an ACL with the flag set in the **suppress-rpf-drop** command.

Clearing RPF statistics for a specified IPv4 interface

To clear RPF statistics on a specific IPv4 physical interface, use the **clear ip interface ethernet** command.

Syntax: clear ip interface ethernet slot/port

Use the **ethernet** parameter to specify the Ethernet or port.

The *slot/port* variables specify the interface for which you want to clear RPF statistics.

Clearing RPF statistics for all IPv4 interfaces within a router

To clear RPF statistics on all IPv4 physical interfaces within a router, use the **clear ip interface counters** command.

Syntax: clear ip interface counters

Clearing RPF statistics for a specified IPv6 interface

To clear RPF statistics on a specific IPv6 physical interface, use the **clear ipv6 interface** command.

Syntax: clear ipv6 interface ethernet slot/port

Use the **ethernet** parameter to specify the Ethernet port.

The *slot/port* variables specify the interface for which you want to clear RPF statistics.

Clearing RPF statistics for all IPv6 interfaces within a router

To clear RPF statistics on all IPv6 physical interfaces within a router, use the **clear ipv6 interface counters** command.

Syntax: clear ipv6 interface counters

Displaying RPF logging

If you set the log option of the **rpf-mode** command, the packets are saved to the system log. To display the log, use the following command.

```
device# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 1305 overruns)
    Buffer logging: level ACDMEINW, 50 messages logged
```

```
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Dynamic Log Buffer (50 lines):
May 11 12:12:54:I:RPF: Denied 1 packets on port 7/5 tcp 10.4.4.1(0) -> 10.6.7.8(0)
```

NOTE

A maximum of 256 RPF log messages are logged per minute.

Unidirectional Link Detection

Unidirectional Link Detection overview

Unidirectional Link Detection (UDLD) monitors a link between two Extreme devices and blocks the ports on both ends of the link if there is a unidirectional failure.

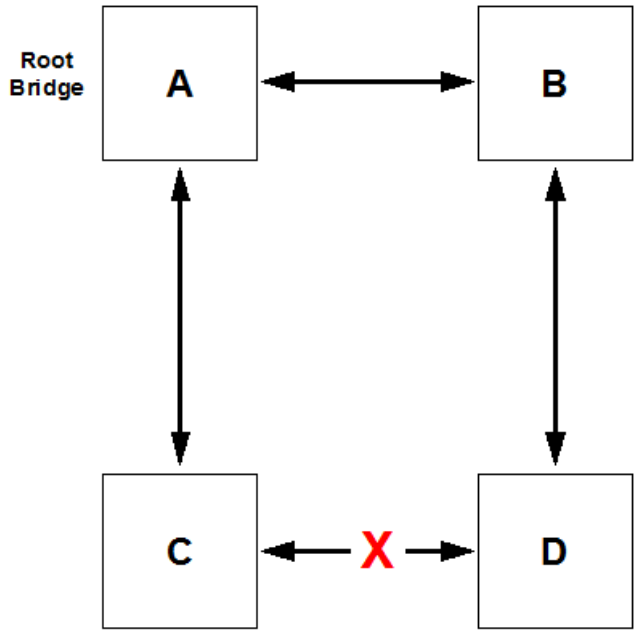
UDLD protocol detects and blocks broken unidirectional links in the network. This is done through the exchange of UDLD protocol data units (PDU) between devices on a physical link. Both ends of the link must support the same proprietary UDLD protocol to detect the unidirectional link condition.

A unidirectional link is assumed when the UDLD stops receiving UDLD PDUs from the other end of the link. The device then blocks the physical link. The physical link will still be up but the line protocol will be down. UDLD PDUs continue to be transmitted and received on the link.

How UDLD works

The following shows a simple four-switch network in which two paths connect to each switch. STP blocks traffic on as many ports as necessary so that only one operational path exists from the STP root bridge to all nodes in the network.

FIGURE 160 Four-switch example for UDLD



In the previous figure, STP detects that the port on Switch D that is connected to Switch C should be put into a blocked state. Therefore, no data traffic gets transmitted or received on this port. Data traffic remains blocked as long as Switch D receives bridge protocol data units (BPDUs) from both switches C and B.

If the link between Switch C and Switch D becomes unidirectional (for reasons such as hardware failure or incorrect cabling) in the direction from D to C, Switch D ages out the status that it was receiving BPDUs from Switch C. This eventually causes STP to put the port in a forwarding state, thus allowing all data traffic. This creates a loop for all BUM traffic that enters the network. BUM traffic can go from Switch B to Switch D to Switch C to Switch A, and then back to Switch B.

To prevent this loop from forming, UDLD can be used to detect that the link between Switch C and Switch D has become unidirectional.

Keepalive interval

By default, ports enabled for UDLD exchange proprietary health-check packets once every 500 ms (the keepalive interval). If a port does not receive a health-check packet from the port at the other end of the link within the keepalive interval, the port waits for four more intervals. If the port still does not receive a health-check packet after waiting for five intervals, the port concludes that the link has failed and takes the port down.

The keepalive interval and keepalive retries can be configured to values other than the default.

- Keepalive interval — You can change the link health-check packet send interval. By default, ports enabled for UDLD send a link health-check packet once every 500 ms. You can change the interval to a value from 1 through 60, where 1 equals 100 ms, 2 equals 200 ms, and so on.
- Keepalive retries — You can set how many retries a port makes when sent health-checks do not receive a reply. By default, a port waits one second to receive a health-check reply packet from the port at the other end of the link. If the port does not receive a reply, the port tries four more times by sending up to four more health-check packets. If the port still does not receive a reply after the maximum number of retries, the port goes down. You can change the maximum number of keepalive retries to a value from 3 through 64.

UDLD for tagged ports

The default implementation of UDLD sends the packets untagged, even across tagged ports. If the untagged UDLD packet is received by a third-party switch, that switch may reject the packet. As a result, UDLD may be limited to only NetIron OS devices, because UDLD may not function on third-party switches.

You can configure ports to send out UDLD control packets that are tagged with a specific VLAN ID as tagged UDLD control packets. Third-party switches are allowed to receive the control packets that are tagged with the specified VLAN.

NOTE
You must configure the same VLANs that will be used for UDLD on all devices across the network; otherwise, the UDLD link cannot be maintained.

Configuration and feature notes for UDLD

- UDLD is supported only on Ethernet ports.
- To configure UDLD on a LAG group, you must configure the feature on each port of the group individually. Configuring UDLD on a LAG group's primary port enables the feature on that port only.
- Dynamic LAG is not supported.
- If you want to configure UDLD on a static LAG group that contains ports on which UDLD is enabled, you must remove the UDLD configuration from the ports. After you create the LAG group, you can again add the UDLD configuration.

- UDLD must be configured on both sides of the link.

Enabling UDLD

UDLD allows you to monitor a link between two devices and bring the ports on both ends of the link down if the link fails at any point between the two.

1. Enter global configuration mode.

```
device# configure terminal
```

2. With the device in global configuration mode, use one or any combination of the following commands to enable UDLD on Ethernet ports.
   - Enable UDLD on a single port.

```
device(config)# link-keepalive ethernet 1/1
```

   - Enable the feature on a LAG group.

```
device(config)# link-keepalive ethernet 1/1 ethernet 1/2
device(config)# link-keepalive ethernet 1/3 ethernet 1/4
```

   - Enable ports to receive and send UDLD control packets tagged with a specific VLAN ID.

```
device(config)# link-keepalive ethernet 1/18 vlan 22
```

3. Set the keepalive interval.

```
device(config)# link-keepalive interval 4
```

4. Set the number of keepalive retries.

```
device(config)# link-keepalive retries 10
```

5. Return to privileged exec mode.

```
device(config)# exit
```

6. Verify the configuration.
   - Verify the configuration for all ports.

```
device# show link-keepalive
Total link-keepalive enabled ports: 2
Keepalive Retries: 5 Keepalive Interval: 5 * 100 MilliSec.
Port   Physical Link   Link-keepalive  Logical link   Link-vlan
1/1    down            down            down           2
1/2    down            down            down
1/18   down            down            down           22
```

   - Verify the configuration for a single port.

```
device# show link-keepalive ethernet 1/1
Current State   : down          Remote MAC Addr  : 0000.0000.0000
Local Port      : 1/1           Remote Port      : n/a
Local System ID : 1bb3d340      Remote System ID : 00000000
Packets sent    : 0             Packets received : 0
Transitions     : 5             Link-Vlan        : 2
```

   - Verify the UDLD state for an individual port. The line protocol state listed in the first line will be "down" if UDLD has brought the port down.

```
device# show interface ethernet 1/1
GigabitEthernet1/1 is disabled, line protocol is down, link keepalive is enabled
  Hardware is GigabitEthernet, address is 000c.dbe2.5900 (bia 000c.dbe2.5900)
  Configured speed 1Gbit, actual unknown, configured duplex fdx, actual unknown
  Configured mdi mode AUTO, actual unknown
  Member of 2 L2 VLANs, port is tagged, port state is Disabled
  STP configured to ON, Priority is level7, flow control enabled
  Force-DSCP disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  MTU 1522 bytes, encapsulation ethernet
  300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 0 packets
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions, DMA transmitted 0 packets
```

7. Save the configuration.

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

8. Clear the UDLD statistics.

```
device# clear link-keepalive statistics
```

UDLD configuration example

```
device# configure terminal
device(config)# link-keepalive ethernet 1/1
device(config)# link-keepalive interval 4
device(config)# link-keepalive retries 10
device(config)# exit
device# show link-keepalive
device# copy running-config startup-config
```

Virtual Switch Redundancy Protocol (VSRP)

VSRP overview

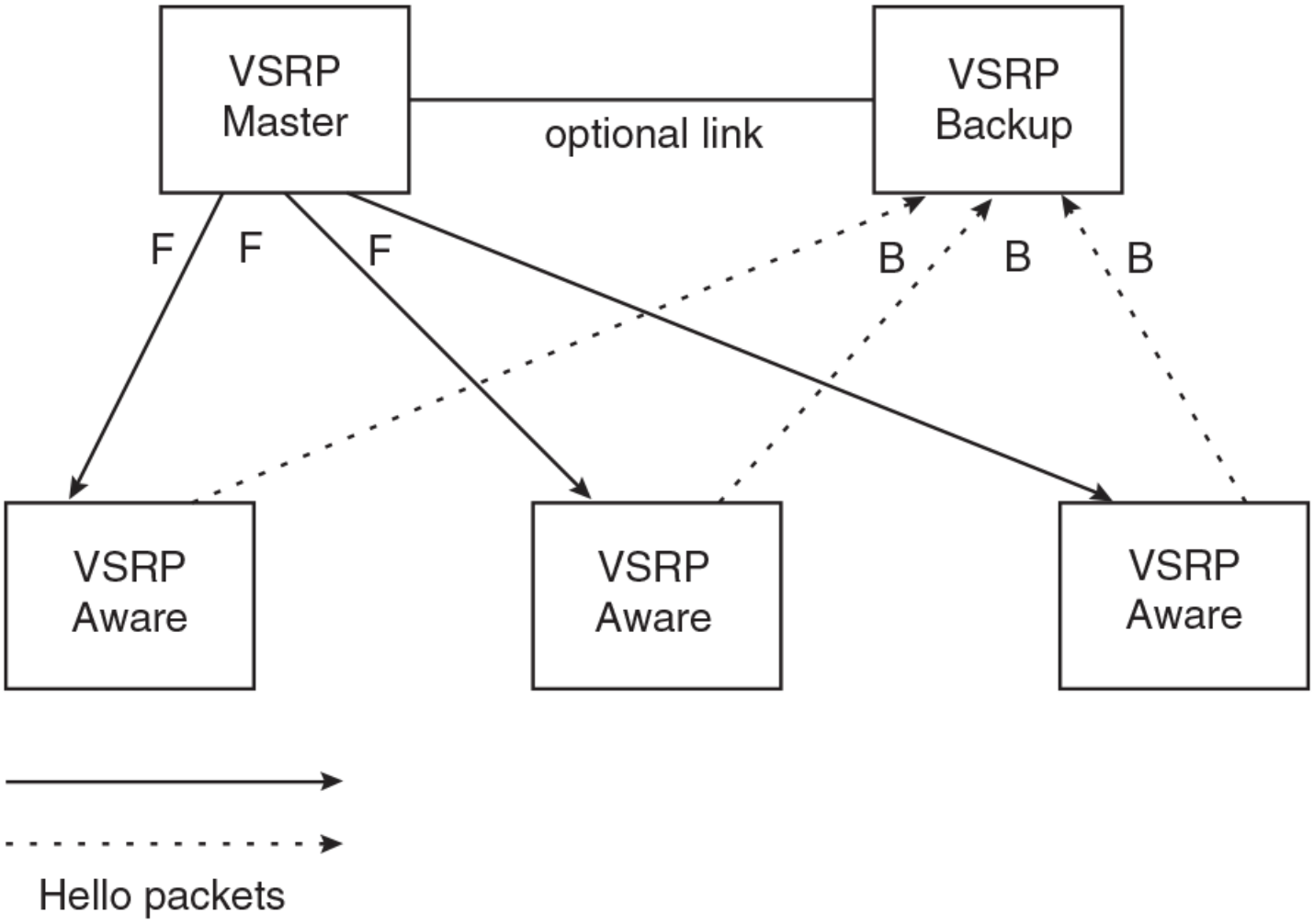
Virtual Switch Redundancy Protocol (VSRP) is an Extreme proprietary protocol that provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Based on the Extreme Virtual Router Redundancy Protocol Extended (VRRP-E), VSRP provides one or more backups for a device. If the active device becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network.

The device support full VSRP as well as VSRP-awareness . An Extreme device that is not itself configured for VSRP but is connected to an Extreme device that is configured for VSRP, is considered to be VSRP aware.

You can use VSRP for Layer 2, Layer 3, or for both layers. On Layer 3 devices, Layer 2 and Layer 3 share the same VSRP configuration information.

The following example shows an example of a VSRP configuration.

FIGURE 161 VSRP mesh - redundant paths for the traffic



In this example, two Extreme devices are configured as redundant paths for VRID 1. On each of the devices, a Virtual Router ID (VRID) is configured on a port-based VLAN. Since VSRP is primarily a Layer 2 redundancy protocol, the VRID applies to the entire VLAN. However, you can selectively remove individual ports from the VRID if needed.

Following Master election (described below), one of the Extreme devices becomes the Master for the VRID and sets the state of all the VLAN ports to Forwarding. The other device is a Backup and sets all the ports in its VRID VLAN to Blocking.

If a failover occurs, the Backup becomes the new Master and changes all its VRID ports to the Forwarding state.

Other Extreme devices can use the redundant paths provided by the VSRP devices. In this example, three Extreme devices use the redundant paths. An Extreme device that is not itself configured for VSRP but is connected to an Extreme device that is configured for VSRP, is VSRP aware . In this example, the three Extreme devices connected to the VSRP devices are VSRP aware. An Extreme device that is VSRP aware can failover its link to the new Master in sub-second time, by changing the MAC address associated with the redundant path.

When you configure VSRP, make sure each of the non-VSRP Extreme devices connected to the VSRP devices has a separate link to each of the VSRP devices.

VSRP configuration notes and feature limitations

- VSRP and 802.1Q-n-Q tagging are not supported together on the same device.
- VSRP and Super Aggregated VLANs are not supported together on the same device.
- The VLAN supports IGMP snooping version 2 and version 3 when VSRP or VSRP-aware is configured on a VLAN.
- VSRP is supported only for VLANs that are part of the default ESI. VSRP is not supported for VLANs configured under user-defined ESIs.

VSRP redundancy

You can configure VSRP to provide redundancy for Layer 2 and Layer 3:

- Layer 2 only - The Layer 2 links are backed up but specific IP addresses are not backed up.
- Layer 2 and Layer 3 - The Layer 2 links are backed up and a specific IP address is also backed up. Layer 3 VSRP is the same as VRRP-E. However, using VSRP provides redundancy at both layers at the same time.

The NetIron OS device supports Layer 2 and Layer 3 redundancy. You can configure the device for either Layer 2 only or Layer 2 and Layer 3. To configure for Layer 3, specify the IP address you are backing up.

NOTE
If you want to provide Layer 3 redundancy only, disable VSRP and use VRRP-E.

Master election and failover

Each VSRP device advertises its VSRP priority in Hello messages. During Master election, the VSRP device with the highest priority for a given VRID becomes the Master for that VRID. After Master election, the Master sends Hello messages at regular intervals to inform the Backups that the Master is healthy.

If there is a tie for highest VSRP priority, the device whose virtual routing interface has a higher IP address becomes the master.

VSRP failover

Each Backup listens for Hello messages from the Master. The Hello messages indicate that the Master is still available. If the Backups stop receiving Hello messages from the Master, the election process occurs again and the Backup with the highest priority becomes the new Master.

Each Backup waits for a specific period of time, the Dead Interval, to receive a new Hello message from the Master. If the Backup does not receive a Hello message from the Master by the time the Dead Interval expires, the Backup sends a Hello message of its own, which includes the Backup's VSRP priority, to advertise the Backup's intent to become the Master. If there are multiple Backups for the VRID, each Backup sends a Hello message.

When a Backup sends a Hello message announcing its intent to become the Master, the Backup also starts a hold-down timer. During the hold-down time, the Backup listens for a Hello message with a higher priority than its own.

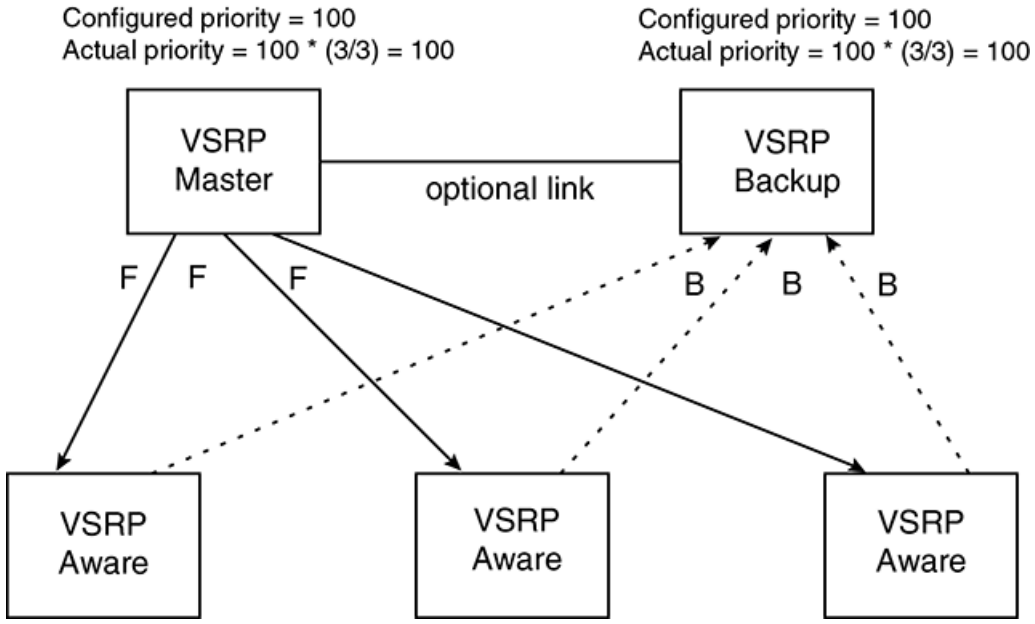
- If the Backup receives a Hello message with a higher priority than its own, the Backup resets its Dead Interval and returns to normal Backup status.
- If the Backup does not receive a Hello message with a higher priority than its own by the time the hold-down timer expires, the Backup becomes the new Master and starts forwarding Layer 2 traffic on all ports.

If you increase the timer scale value, each timer value is divided by the scale value. To achieve sub-second failover times, you can change the scale to a value up to 10. This shortens all the VSRP timers to 10 percent of their configured values.

VSRP priority calculation

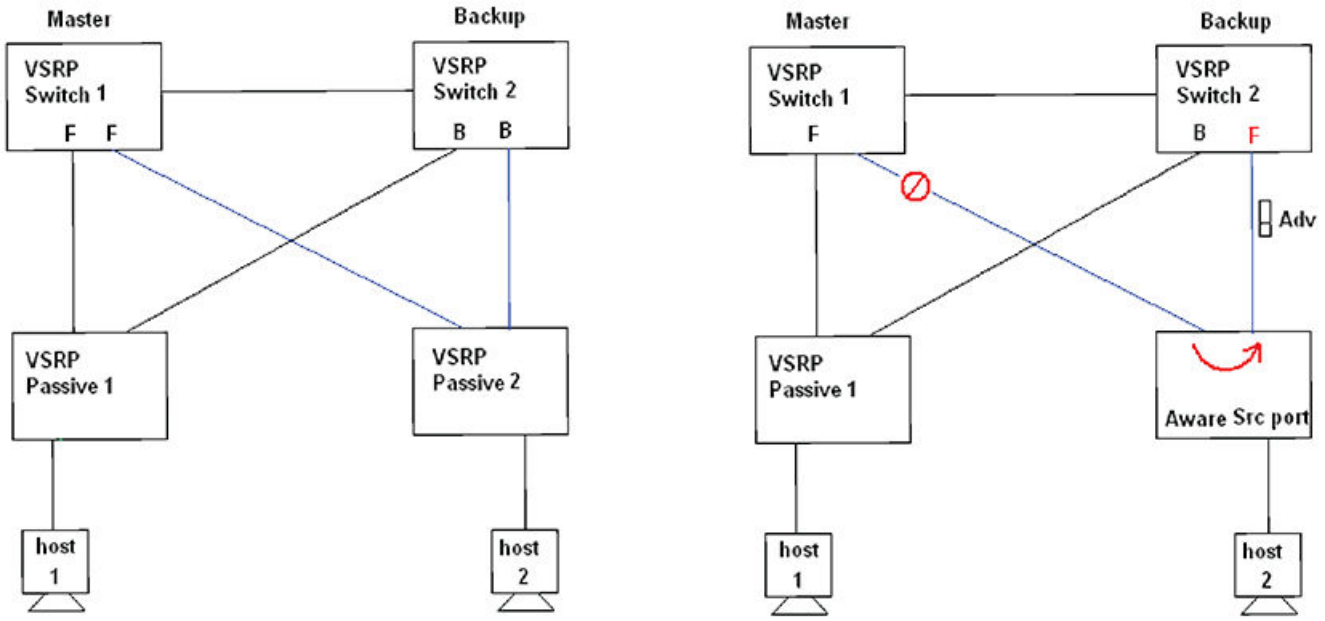
Each VSRP device has a VSRP priority for each VRID and its VLAN. The VRID is used during Master election for the VRID. By default, a device VSRP priority is the value configured on the device (which is 100 by default). However, to ensure that a Backup with a high number of up ports for a given VRID is elected, the device reduces the priority if a port in the VRID VLAN goes down. For example, if two Backups each have a configured priority of 100, and have three ports in VRID 1 in VLAN 10, each Backup begins with an equal priority, 100. This is shown in the following figure.

FIGURE 162 VSRP priority



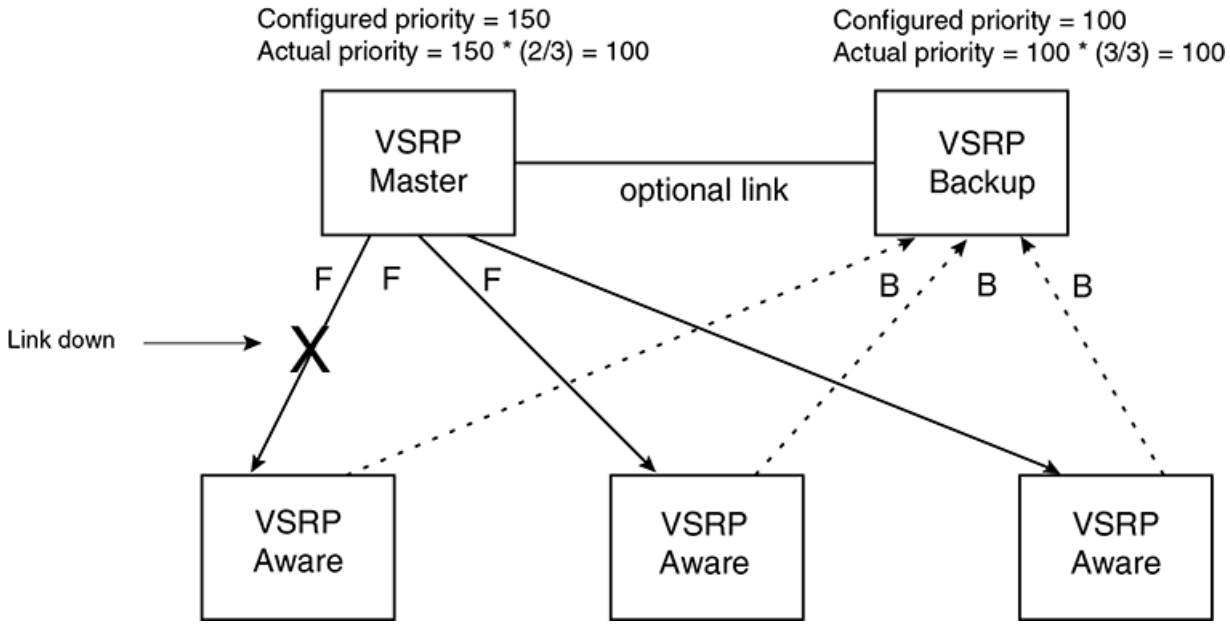
However, if one of the VRID ports goes down on one of the Backups, that Backup priority is reduced. If the Master priority is reduced enough to make the priority lower than a Backup priority, the VRID fails over to the Backup. The following figure shows an example.

FIGURE 163 VSRP priority recalculation



You can reduce the sensitivity of a VSRP device to failover by increasing its configured VSRP priority. For example, you can increase the configured priority of the VSRP device on the left in Figure 165 to 150. In this case, failure of a single link does not cause failover. The link failure caused the priority to be reduced to 100, which is still equal to the priority of the other device. This is shown in the following figure.

FIGURE 164 VSRP priority bias

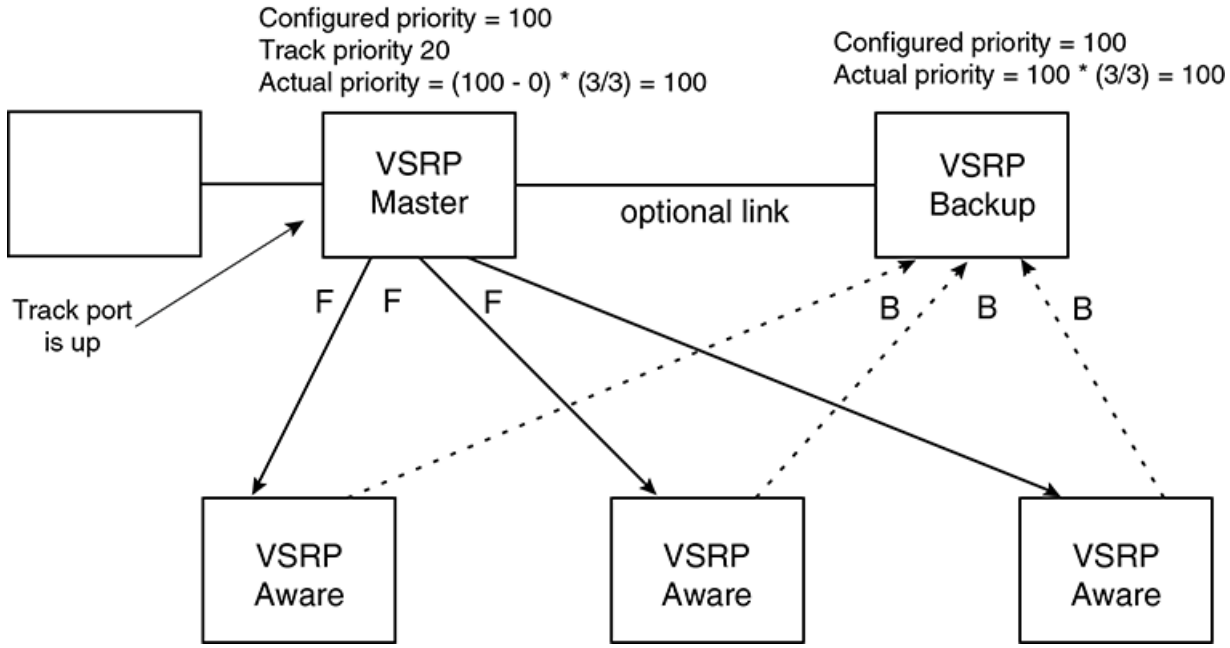


Track ports

Optionally, you can configure track ports to be included during VSRP priority calculation. In VSRP, a track port is a port that is not a member of the VRID VLAN, but whose state is nonetheless considered when the priority is calculated. Typically, a track port represents the exit side of traffic received on the VRID ports. By default, no track ports are configured.

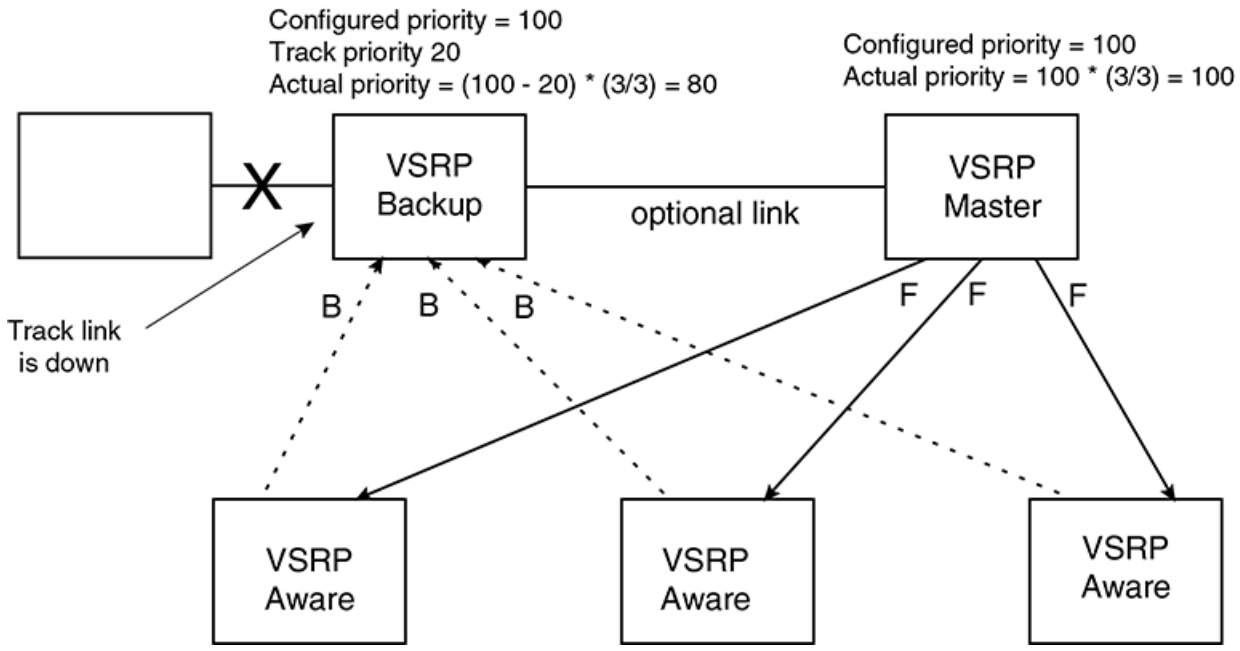
When you configure a track port, you assign a priority value to the port. If the port goes down, VSRP subtracts the track port priority value from the configured VSRP priority. For example, if the you configure a track port with priority 20 and the configured VSRP priority is 100, the software subtracts 20 from 100 if the track port goes down, resulting in a VSRP priority of 80. The new priority value is used when calculating the VSRP priority. The following figure shows an example.

FIGURE 165 Track port priority



In Figure 167, the track port is up. SInce the port is up, the track priority does not affect the VSRP priority calculation. If the track port goes down, the track priority does affect VSRP priority calculation, as shown in the following figure.

FIGURE 166 Track port priority subtracted during priority calculation



MAC address failover on VSRP-aware devices

VSRP-aware devices maintain a record of each VRID and its VLAN. When the device has received a Hello message for a VRID in a given VLAN, the device creates a record for that VRID and VLAN and includes the port number in the record. Each subsequent time the device receives a Hello message for the same VRID and VLAN, the device checks the port number:

- If the port number is the same as the port that previously received a Hello message, the VSRP-aware device assumes that the message came from the same VSRP Master that sent the previous message.
- If the port number does not match, the VSRP-aware device assumes that a VSRP failover has occurred to a new Master, and moves the MAC addresses learned on the previous port to the new port.

The VRID records age out if unused. This can occur if the VSRP-aware device becomes disconnected from the Master. The VSRP-aware device will wait for a Hello message for the period of time equal to the following.

VRID Age = Dead Interval + Hold-down Interval + (3 x Hello Interval)

The values for these timers are determined by the VSRP device sending the Hello messages. If the Master uses the default timer values, the age time for VRID records on the VSRP-aware devices is as follows.

3 + 2 + (3 x 1) = 8 seconds

In this case, if the VSRP-aware device does not receive a new Hello message for a VRID in a given VLAN, on any port, the device assumes the connection to the Master is unavailable and removes the VRID record.

Configuring device redundancy using VSRP

Virtual Switch Redundancy Protocol (VSRP) provides device redundancy for specific ports in a port-based VLAN. Configuring VSRP device redundancy in your network leads to faster failover times if an interface goes offline.

VSRP is enabled after assigning a Virtual Routing ID (VRID) on specific ports in a port-based VLAN and setting a backup priority for the device. Repeat this task on each device selected for VSRP redundancy.

NOTE
VSRP is enabled by default on the device, but may be disabled if Virtual Router Redundancy Protocol (VRRP) or VRRP Extended (VRRP-E) is currently enabled.

1. On any device on which you want to configure VSRP service, from privileged EXEC mode, enter global configuration mode by issuing the **configure terminal** command.

   ```
   device# configure terminal
   ```

2. Optionally, globally enable the VSRP protocol.

   This is required only if VSRP was disabled earlier and you want to re-enable it.

   ```
   device(config)# router vsrp
   ```

3. Configure a VLAN by assigning an ID to the VLAN.

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

4. Configure the interfaces on which VSRP service is to be enabled by adding ports to the VLAN.

   ```
   device(config-vlan-200)# tagged ethernet 1/1 to 1/8
   ```

   In this example, a range of tagged Ethernet interfaces is configured.

5. Assign a VSRP VRID to the VLAN.

   ```
   device(config-vlan-200)# vsrp vrid 1
   ```

6. (Optional) Add additional ports to the VSRP instance.

   ```
   device(config-vlan-200-vrid-1)# include-port ethernet 1/10
   ```

7. (Optional) Configure VRID IP address if you are configuring Layer 3 redundancy.

   ```
   device(config-vlan-200-vrid-1)# ip-address 10.10.10.1
   ```

   VSRP does not require you to specify an IP address. If you do not specify an address, VSRP provides Layer 2 redundancy. If you specify an IP address, VSRP provides Layer 2 and Layer 3 redundancy.

8. Designate this device as a backup VSRP device with a priority higher than the default priority.

   ```
   device(config-vlan-200-vrid-1)# backup priority 110
   ```

   The priority is used to determine the initial VSRP master device. If a VSRP master device goes offline, the backup device with the highest priority will assume the role of master device.

9. Enable a backup router to send hello messages to the master VSRP device.

   ```
   device(config-vlan-200-vrid-1)# advertise backup
   ```

   By default, backup VSRP devices do not send hello messages to advertise themselves to the master.

10. Enable the VRRP session.

    You can also use the **enable** command to enable the VRRP session.

```
device(config-vlan-200-vrid-1)# activate
```

11. Return to privileged EXEC mode.

```
device(config-vlan-200-vrid-1)# end
```

12. Display VSRP information about the VRID to verify the configuration steps in this task.

```
device# show vsrp vrid 100

VLAN 100
Auth-type no authentication
VRID 100
========
State          Administrative-status       Advertise-backup      Preempt-mode
Master         Enabled                     Disabled              True
Parameter      Configured   Current        Unit/Formula
Priority       100          100            (100-0)*(3.0/3.0)
Hello-interval 1            1              sec/1
Dead-interval  3            3              sec/1
Hold-interval  3            3              sec/1
Initial-ttl    2            2              hops
Next hello sent in 00:00:00
Member ports:      ethe 1/1 ethe 2/1 ethe 2/10
Operational ports: ethe 1/1 ethe 2/1 ethe 2/10
```

    This is an optional step. Before entering the **show vsrp vrid** command, you may need to activate several VSRP backup devices.

The following example configures VSRP service for VRID 1 on Ethernet interfaces 1/1 to 1/8 of VLAN 200.

```
device# configure terminal
device(config)# router vsrp
device(config)# vlan 200
device(config-vlan-200)# tagged ethernet 1/1 to 1/8
device(config-vlan-200)# vsrp vrid 1
device(config-vlan-200-vrid-1)# backup priority 110
device(config-vlan-200-vrid-1)# advertise backup
device(config-vlan-200-vrid-1)# activate
device(config-vlan-200-vrid-1)# end
device# show vsrp vrid 1
```

Configuring optional VSRP parameters

You can configure several optional VSRP parameters.

VSRP is configured and enabled.

VSRP is enabled after assigning a Virtual Routing ID (VRID) on specific ports in a port-based VLAN and setting a backup priority for the device. You can configure a number of optional parameters once VSRP is enabled.

NOTE

VSRP is enabled by default on the device, but may be disabled if Virtual Router Redundancy Protocol (VRRP) or VRRP Extended (VRRP-E) is currently enabled.

NOTE
All the steps in this section are optional.

1. On any device on which you want to configure, from privileged EXEC mode, enter global configuration mode by issuing the **configure terminal** command.

```
device# configure terminal
```

2. Configure a VLAN by assigning an ID to the VLAN.

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

3. Assign a VSRP VRID to the VLAN.

```
device(config-vlan-200)# vsrp vrid 1
```

4. Configure a Backup to save the VSRP timer values received from the Master instead of the timer values configured on the Backup.

```
device(config-vlan-200-vrid-1)# save-current-values
```

5. Configure how many hops the packet can traverse before being dropped.

```
device(config-vlan-200-vrid-1)# initial-ttl 5
```

6. Configure the number of seconds between hello messages from the master to the backups for a given VRID.

```
device(config-vlan-200-vrid-1)# hello-interval 10
```

7. Configure the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is offline.

```
device(config-vlan-200-vrid-1)# dead-interval 15
```

8. Configure the interval for the backup to send hello messages to the master when the advertisement is enabled.

```
device(config-vlan-200-vrid-1)# backup-hello-interval 180
```

9. Change the hold-down time interval.

   The hold-down interval prevents Layer 2 loops from occurring during failover, by delaying the new Master from forwarding traffic long enough to ensure that the failed Master is really unavailable.

```
device(config-vlan-200-vrid-1)# hold-down-interval 4
```

Configuring authentication on VSRP interfaces

If the interfaces on which you configure the VRID use authentication, the VSRP packets on those interfaces also must use the same authentication.

A VSRP session must be configured and running.

If you configure your device interfaces to use a simple password to authenticate traffic, VSRP interfaces can be configured with the same simple password, and VSRP packets that do not contain the password are dropped. If your interfaces do not use authentication, neither does VSRP. Repeat this task on all interfaces on all devices that support the VRID.

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

```
device# configure terminal
```

2. Configure the VLAN on which a VSRP VRID is assigned.

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

3. Enter the simple text password configuration.

```
device(config-vlan-100)# vsrp auth-type simple-text-auth ourpword
```

4. Verify the password.

```
device# show vsrp
VLAN 200
Auth-type simple text password
VRID 1
========
State    Administrative-status        Advertise-backup    Preempt-mode
Master    Enabled                        Disabled            True
Parameter             Configured        Current        Unit/Formula
Priority                 100                100              (100-0)*(3.0/3.0)
Hello-interval        1                    1                sec/1
Dead-interval         3                    3                sec/1
Hold-interval         3                    3                sec/1
Initial-ttl              2                    2                hops
Next hello sent in 00:00:00
Member ports:      ethe 1/1 ethe 2/1 ethe 2/10
Operational ports: ethe 1/1 ethe 2/1 ethe 2/10
```

Tracking ports and setting the VSRP priority

Configuring port tracking on an exit path interface and setting a priority on a VSRP device enables VSRP to monitor the interface. If the interface goes down, the VRID's VSRP priority is reduced by the amount of the track port priority you specify.

This capability is useful for tracking the state of the exit interface for the path for which the VRID is providing redundancy

1. Enter the global configuration mode.

```
device# configure terminal
```

2. Optionally, globally enable VSRP.

```
device(config)# router vsrp
```

3. Configure a VLAN by assigning an ID to the VLAN.

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

4. Configure the interfaces on which VSRP service is to be enabled by adding ports to the VLAN.

```
device(config-vlan-200)# tagged ethernet 1/1 to 1/8
```

5. Assign a VSRP VRID to the VLAN.

```
device(config-vlan-200)# vsrp vrid 1
```

6. Configure the track port and priority.

```
device(config-vlan-200-vrid-1)# track-port ethernet 1/2 priority 4
```

The priority value is used when a tracked port goes down and the new priority is set to this value. Ensure that the priority value is lower than the priority set for any existing master or backup device to force a renegotiation for the master device.

Disabling backup pre-emption setting

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the VRID. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, because the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

Disabling VSRP backup preemption

VRRP backup preemption prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

A VSRP session must be globally enabled using the **router vsrp** command in global configuration mode.

1. Configure a VLAN by assigning an ID to the VLAN.

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

2. Configure the interfaces on which VSRP service is to be enabled by adding ports to the VLAN.

```
device(config-vlan-200)# tagged ethernet 1/1 to 1/8
```

3. Assign a VSRP VRID to the VLAN.

```
device(config-vlan-200)# vsrp vrid 1
```

4. Disable preemption on a Backup.

```
device(config-vlan-200-vrid-1)# non-preempt-mode
```

VSRP fast start

VSRP fast start allows non-Extreme or non-VSRP aware devices that are connected to an Extreme device that is the VSRP Master to quickly switchover to the new Master when a VSRP failover occurs

This feature causes the port on a VSRP Master to restart when a VSRP failover occurs. When the port shuts down at the start of the restart, ports on the non-VSRP aware devices that are connected to the VSRP Master flush the MAC address they have learned for the VSRP master. After a specified time, the port on the previous VSRP Master (which now becomes the Backup) returns back online. Ports on the non-VSRP aware devices switch over to the new Master and learn its MAC address.

Special considerations when configuring VSRP fast start

Consider the following when configuring VSRP fats start:

- VSRP is sensitive to port status. When a port goes down, the VSRP instance lowers its priority based on the port up fraction. Since the VSRP fast start feature toggles port status by bringing ports down and up it can affect VSRP instances because their priorities get reduced when a port goes down. To avoid this, the VSRP fast start implementation keeps track of ports that it brings down and suppresses port down events for these ports (as concerns VSRP).
- Once a VSRP restart port is brought up by a VSRP instance, other VSRP instances (in Master state) that have this port as a member do not go to forwarding immediately. This is a safety measure that is required to prevent transitory loops. This could happen if a peer VSRP node gets completely cut off from this node and assumed Master state. In this case, where there are 2 VSRP instances that are in Master state and forwarding, the port comes up and starts forwarding immediately. This would cause a forwarding loop. To avoid this, the VSRP instance delays forwarding.

Recommendations for configuring VSRP fast start

The following recommendations apply to configurations where multiple VSRP instances are running between peer devices sharing the same set of ports:

- Multiple VSRP instances configured on the same ports can cause VSRP instances to be completely cut off from peer VSRP instances. This can cause VSRP instances to toggle back and forth between master and backup mode. For this reason, we recommend that you configure VSRP fast start on a per port basis rather than for the entire VLAN.
- We recommend that VSRP peers have a directly connected port without VSRP fast start enabled on it. This allows protocol control packets to be received and sent even if other ports between the master and standby are down.
- The VSRP restart time should be configured based on the type of connecting device since some devices can take a long time to bring a port up or down (as long as several seconds). In order to ensure that the port restart is registered by neighboring device, the restart time may need to be changed to a value higher than the default value of 1 second.

Configuring VSRP fast start globally

VSRP fast start enables non-NetIron OS or non-VSRP aware devices that are connected to a NetIron OS device which is the VSRP Master to quickly switch over to the new Master when VSRP failover occurs.

VSRP is enabled.

VSRP fast start can be enabled on a VSRP-configured device, either on a VALN to which the VRID of the VSRP-configured device belongs (globally) or on a port that belongs to the VRID.

1. On any device on which you want to configure, from privileged EXEC mode, enter global configuration mode by issuing the **configure terminal** command.

   ```
   device# configure terminal
   ```

2. Configure a VLAN by assigning an ID to the VLAN.

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

3. Assign a VSRP VRID to the VLAN.

```
device(config-vlan-100)# vsrp vrid 100
```

4. Enable VSRP fast start. Globally configure a VSRP-configured device to shut down its ports when a failover occurs, and restart after a specified time. This will shutdown all the ports, with the specified VRID, that belong to the VLAN when failover occurs.

```
device(config-vlan-100-vrid-100)# restart-ports 5
```

5. Verify the configuration using **show vsrp vrid** command.

```
# show vsrp vrid 100
VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode save-current
   master     enabled               disabled         true         false
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops
   next hello sent in 00:00:00.3
   Member ports:      ethernet 1/2/5 to 1/2/8
   Operational ports: ethernet 1/2/5 ethernet 1/2/8
   Forwarding ports:  ethernet 1/2/5 ethernet 1/2/8
   Restart ports:     1/2/5(1) 1/2/6(1) 1/2/7(1) 1/2/8(1)

device# show vsrp vrid 100
VLAN 100
auth-type no authentication
VRID 100
========
State        Administrative-status    Advertise-backup    Preempt-mode    save-current
master        enabled                    disabled            true            false
Parameter             Configured        Current        Unit/Formula
priority              100                    50              (100-0)*(2.0/4.0)
hello-interval     1                      1                sec/1
dead-interval      3                      3                sec/1
hold-interval      3                      3                sec/1
initial-ttl           2                      2                hops
next hello sent in 00:00:00.3
Member ports:      ethe 2/5 to 2/8
Operational ports: ethe 2/5 ethe 2/8
Forwarding ports:  ethe 2/5 ethe 2/8
Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

VSRP slow start

You can configure the VSRP slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup.

In a VSRP configuration, if a Master router goes down, the Backup router with the highest priority takes over. When the Master comes back up again, it takes over from the Backup. By default, this transition from Backup back to Master takes place immediately. You can configure the VSRP slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup. (This range is currently set to between 1 to 600 ticks (1/10 second to 60 seconds). This interval allows time for VSRP convergence when the Master is restored.

When the VSRP slow start timer is enabled, if the Master goes down, the Backup takes over immediately. If the Master subsequently comes back up again, the amount of time specified by the VSRP slow start timer elapses (in this example, 30 seconds) before the Master takes over from the Backup.

Configuring VSRP slow start

You can configure the VSRP slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup.

VSRP is enabled.

When the VSRP slow start timer is enabled, if the Master goes down, the Backup takes over immediately. If the Master subsequently comes back up again, the amount of time specified by the VSRP slow start timer elapses (in this example, 30 seconds) before the Master takes over from the Backup.

1. On any device on which you want to configure, from privileged EXEC mode, enter global configuration mode by issuing the **configure terminal** command.

   ```
   device# configure terminal
   ```

2. Enter the VSRP router configuration mode.

   ```
   device(config)# router vsrp
   ```

3. Configure VSRP slow start.

   ```
   device(config-vsrp-router)# slow-start 300
   ```

VSRP 2

In VSRP setup, there are always at least two VSRP switches for each VSRP instance. A passive device should always have either one access link or one trunk link connected with each VSRP switch for each VSRP instance. This can create a black hole scenario. A black hole is when VSRP failover causes data traffic from the switches/hosts which connect to VSRP passive switch to go nowhere.

VSRP 2 can detect the health of each pair/set of links for each VSRP instance.VSRP 2 detects which link of VSRP backup switch not receiving any advertisement for specific time duration. The VSRP backup switch can treat the link of the pair on the master side is down or broken and set its own link of the pair in forwarding state. The VSRP backup switch sends out gratuitous ARP for VSRP master only to this link. Other links of other VSRP instances in VSRP backup switch are still in blocking state as shown in figure Figure 169,Figure 170, and Figure 171.

FIGURE 167 Black hole scenario 1

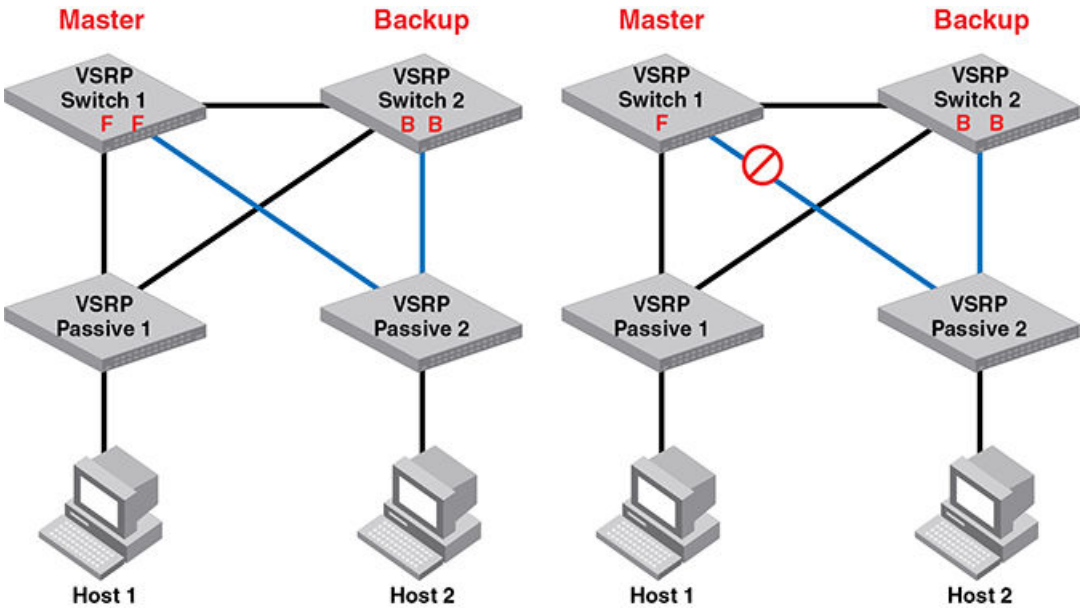


FIGURE 168 Black hole scenario 2

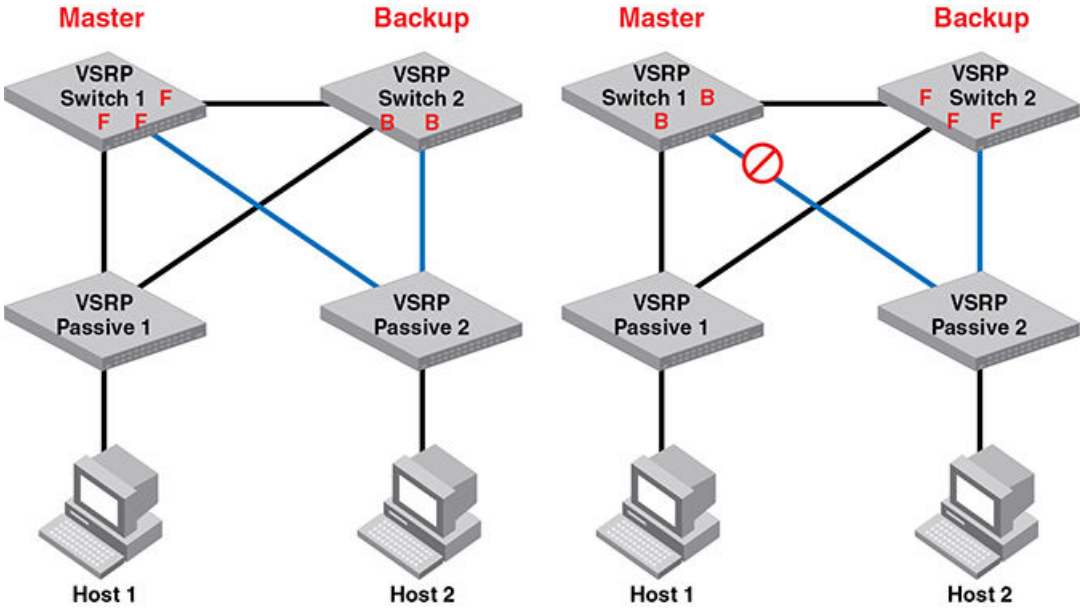
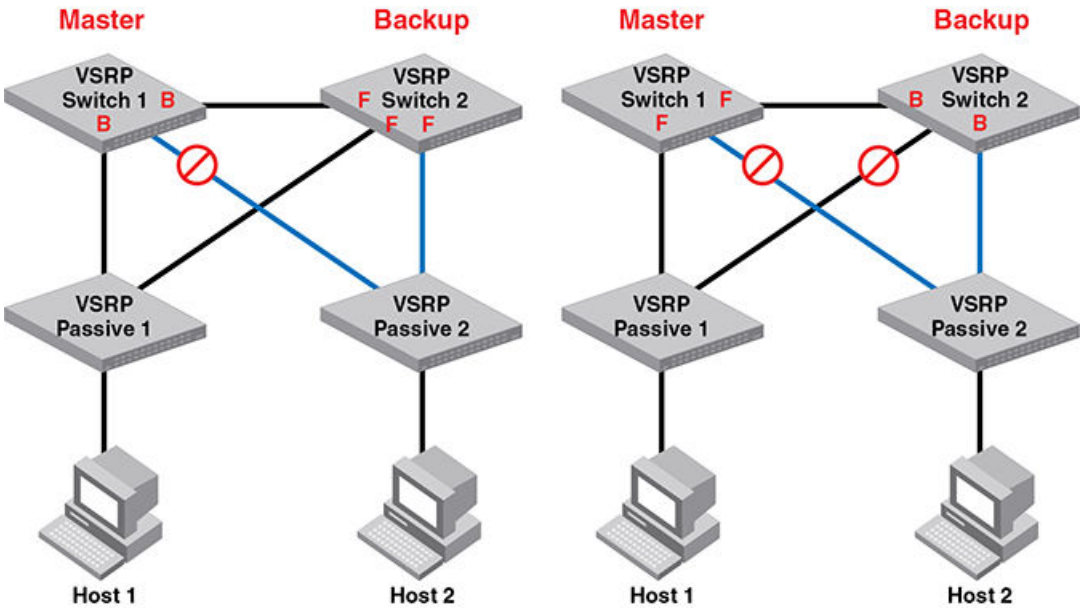


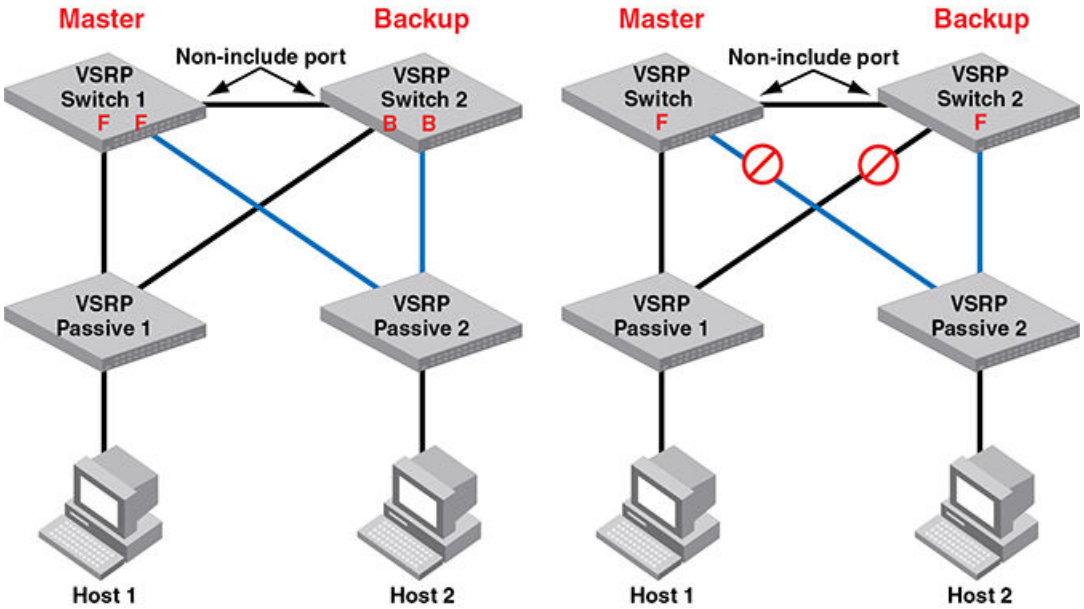
FIGURE 169 Black hole scenario 3



VSRP failover:

- VSRP backup set all include links in blocking state. Blocking ports drop data traffic.
- VSRP failover changes master state by current priority change.
- Current priority changes by link failure and track port failure.

FIGURE 170 Correct VSRP behavior



VSRP is switch redundancy, VSRP 2 is link redundancy.

When VSRP backup changes an include port from blocking state to forwarding state, to make the aware session changes, VSRP backup will send advertisements on the forwarding include ports every 3*hello time.

VSRP aware switches are able to change the src port of aware session and flush MACs.

For VSRP non-aware switches (other vendors), the non-aware switch will flush MACs because the link connecting VSRP master is failed.

VSRP backup will toggle the interface when it sets an include port to forwarding state by VSRP 2.

VSRP 2 doesn't change the master/backup state, only changes the port state.

The change of Master/backup state (VSRP failover) still follows the rules of current priority of VSRP.

VSRP 2 supports:

- hold-down time
- track port
- preempt-mode
- restart port
- topology groups
- VLAN groups.
- VPLS VLAN by topology group.
- Layer 3 VSRP with a condition: non-include link in between two VSRP routers is a must.

Configuration considerations:

- If multiple VSRP instances on multiple VLAN are configured the on one side of link pair, the same VSRP instances of same VLANs must configure on another side of link pair.
- Link-pair has to be enabled or disabled on all VSRP switches for the same VSRP instance.
- Currently, VSRP 2 only supports two VSRP switches in the topology. Multiple VSRP switches may cause a loop when the link redundancy set the VSRP ports in the forwarding state in the link failed cases.
- For VSRP 2 supporting Layer 3 VSRP, it is necessary to have a non-include link in between two VSRP switches. VSRP virtual router is still in VSRP master. Layer 3 data traffic is switched by VSRP backup to VSRP master. The traffic is routed by VSRP master (virtual router).

Configuring VSRP 2

VSRP 2 detects which link of VSRP backup switch not receiving any advertisement for specific time duration.

VSRP is enabled.

1. From global configuration mode, configure a VLAN by assigning an ID to the VLAN

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

2. Assign a VSRP VRID to the VLAN.

   ```
   device(config-vlan-200)# vsrp vrid 1
   ```

3. Enable link redundancy.

   ```
   device(config-vlan-200-vsrp-1)# link-redundancy
   ```

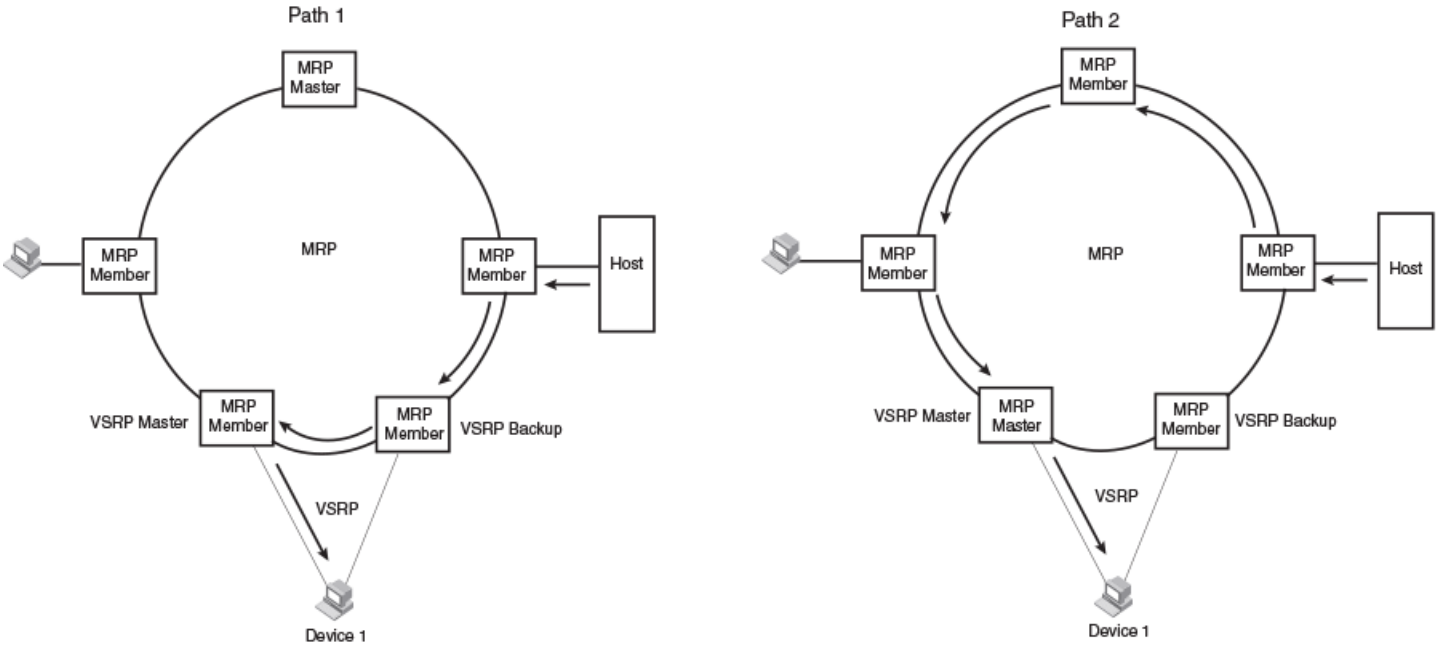
4. Verify the configuration using the **show vsrp** command.

```
device# show vsrp
VLAN 10
 Auth-type no authentication
 VRID 1
 ========
  State           Administrative-status Advertise-backup Preempt-mode Link-Red
  Backup          Enabled               Enabled          True         Enabled
  Parameter        Configured Current    Unit/Formula
  Priority         100        100                (100-0)*(3.0/3.0)
  Hello-interval   1          1                   sec/1
  Dead-interval    3          3                   sec/1
  Hold-interval    3          3               sec/1
  Initial-ttl      2          2               hops
  Backup-Hello     60                                    60                  sec/1
  Master router 10.243.150.0 or MAC xxxx.dbf3.9600 expires in 00:00:03
  Member ports:      ethe 2/14 ethe 2/18 to 2/19
  Operational ports: ethe 2/14 ethe 2/18 to 2/19
   Forwarding ports:  ethe 2/14
Link-Redundancy-port:
  port 2/14 status FORWARD
  port 2/18 status BLOCK
  port 2/19 status BLOCK
```

VSRP and MRP signaling

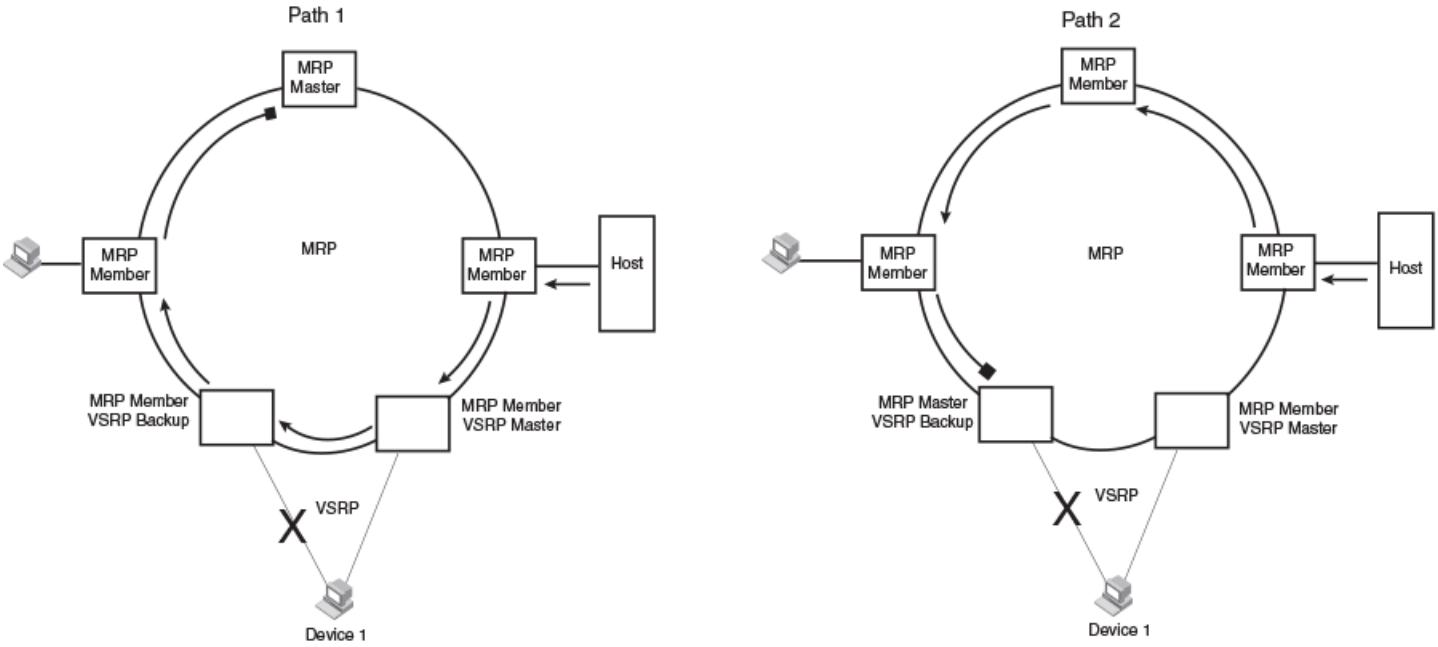
A device may connect to an MRP ring through VSRP to provide a redundant path between the device and the MRP ring. VSRP and MRP signaling ensures rapid failover by flushing MAC addresses appropriately. The host on the MRP ring learns the MAC addresses of all devices on the MRP ring and VSRP link. From these MAC addresses, the host creates a MAC database (table), which is used to establish a data path from the host to a VSRP-linked device. The following figure below shows two possible data paths from the host to Device 1.

FIGURE 171 Two data paths from host on an MRP ring to a VSRP-linked device



If a VSRP failover from master to backup occurs, VSRP needs to inform MRP of the topology change; otherwise, data from the host continues along the obsolete learned path and never reach the VSRP-linked device, as shown in the following figure.

FIGURE 172 VSRP on MRP rings that failed over

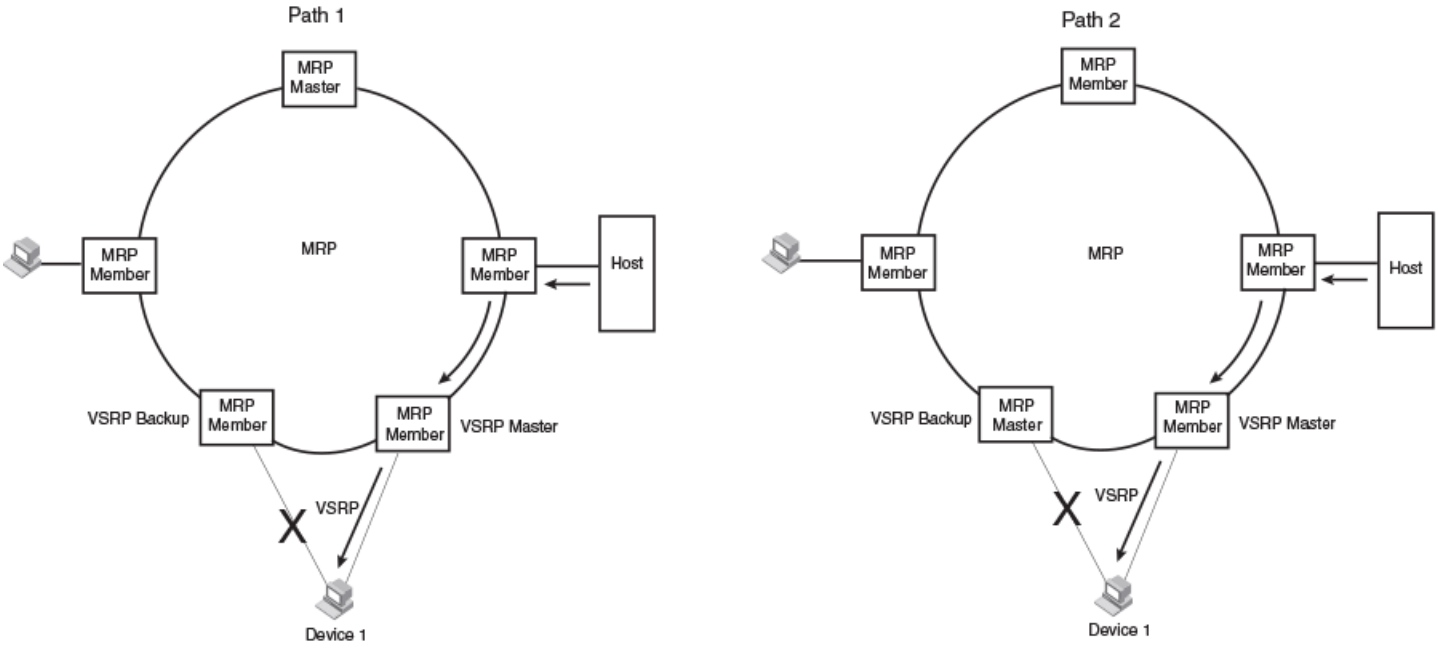


A signaling process for the interaction between VSRP and MRP ensures that MRP is informed of the topology change and achieves convergence rapidly. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all MRP instances impacted by the failover. Then each MRP instance does the following:

- The MRP node sends out an MRP PDU with the mac-flush flag set three times on the MRP ring.
- The MRP node that receives this MRP PDU empties all the MAC entries from its interfaces that participate on the MRP ring.
- The MRP node then forwards the MRP PDU with the mac-flush flag set to the next MRP node that is in forwarding state.

The process continues until the Master MRP node secondary (blocking) interface blocks the packet. Once the MAC address entries have been flushed, the MAC table can be rebuilt for the new path from the host to the VSRP-linked device as shown in the following figure.

FIGURE 173 New path established



There are no CLI commands used to configure this process.