Extreme

CONFIGURATION GUIDE

Extreme NetIron
MPLS Configuration Guide, 06.1.00

Supporting NetIron OS 06.1.00

Contents

Preface

Document conventions

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

Notes, cautions, and warnings

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

hazards.

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

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

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

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

Text formatting conventions

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

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

Format	Description
	Identifies command syntax examples.

Command syntax conventions

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

Convention	Description
bold text	Identifies command names, keywords, and command options.
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.

Extreme resources

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

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

Document feedback

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

You can provide feedback in two ways:

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

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

Contacting Extreme Technical Support

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

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

- GTAC (Global Technical Assistance Center) for immediate support
  - Phone: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact.
  - Email: support@extremenetworks.com. To expedite your message, enter the product name or model number in the subject line.
- GTAC Knowledge – Get on-demand and tested resolutions from the GTAC Knowledgebase, or create a help case if you need more guidance.
- The Hub – A forum for Extreme customers to connect with one another, get questions answered, share ideas and feedback, and get problems solved. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
- Support Portal – Manage cases, downloads, service contracts, product licensing, and training and certifications.

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

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

About This Document

Audience

This document is designed for system administrators with a working knowledge of Layer2 and Layer3 switching and routing.

If you are using an Extreme device, 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

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

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.

What's new in this document

This is a rebranding release.

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.

MPLS Traffic Engineering

MPLS Traffic Engineering overview

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) is supported on Extreme devices. MPLS can be used to direct packets through a network over a pre-determined path of routers. Forwarding decisions in MPLS are based on the contents of a label applied to the packet.

Traffic engineering is the ability to direct packets through a network efficiently, using information gathered about network resources. When used as an application of MPLS, traffic engineering involves creating paths that make the best use of available network resources, avoiding points of congestion and making efficient use of high bandwidth interfaces. Packets traveling over these paths are forwarded using MPLS.

IETF RFC and Internet draft support for MPLS TE

The implementation of MPLS Traffic Engineering (TE) supports the following IETF RFCs and Internet Drafts.

RFC	Description
RFC 3031	Multiprotocol Label Switching Architecture
RFC 3032	MPLS Label Stack Encoding
RFC 3036	LDP Specification
RFC 2205	Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification
RFC 2209	Resource ReSerVation Protocol (RSVP) Version 1 Message Processing Rule
RFC 3209	RSVP-TE
RFC 4090	Fast Reroute

MPLS

RFC 3031 - Multiprotocol Label Switching Architecture

RFC 3032 - MPLS Label Stack Encoding

RFC 3036 - LDP Specification

RFC 2205 - *Resource ReSerVation Protocol (RSVP)* -- Version 1 Functional Specification

RFC 2209 - *Resource ReSerVation Protocol (RSVP)* -- Version 1 Message Processing Rule

RFC 3209 - RSVP-TE

RFC 3270 - MPLS Support of Differentiated Services

RFC 4090 - Fast Reroute

OSPF

RFC 3630 TE Extensions to OSPF v2

IS-IS

RFC 3784 Intermediate System to Intermediate System (IS-IS) Extensions for *Traffic Engineering (TE)*

How MPLS works

MPLS uses a *label switching* forwarding method to direct packets through a network. In label switching, a packet is assigned a label and passes along a predetermined path of routers. Forwarding decisions are based on the contents of the label, rather than information in the packet's IP header.

The following sections describe these basic MPLS concepts:

- How packets are forwarded through an MPLS domain
- The kinds of *Label Switched Paths (LSPs)* that can be configured on a device
- The components of an MPLS label header

How packets are forwarded through an MPLS domain

An *MPLS domain* consists of a group of MPLS-enabled routers, called *Label Switching Routers (LSRs)*. In an MPLS domain, packets are forwarded from one MPLS-enabled router to another along a predetermined path, called an LSP. LSPs are one-way paths between MPLS-enabled routers on a network. To provide two-way traffic, the user configures LSPs in each direction.

The LSRs at the headend and tailend of an LSP are known as *Label Edge Routers (LERs)*. The LER at the headend, where packets enter the LSP, is known as the *ingress LER* . The LER at the tailend, where packets exit the LSP, is known as the *egress LER* . Each LSP has one ingress LER and one egress LER. Packets in an LSP flow in one direction: from the ingress LER towards the egress LER. In between the ingress and egress LERs there may be zero or more *transit LSRs* . A device enabled for MPLS can perform the role of ingress LER, transit LSR, or egress LER in an LSP. Further, a device can serve simultaneously as an ingress LER for one LSP, transit LSR for another LSP, and egress LER for some other LSP.

Label switching in an MPLS domain depicts an MPLS domain with a single LSP consisting of three LSRs: an ingress LER, a transit LSR, and an egress LER.

Label switching in an MPLS domain
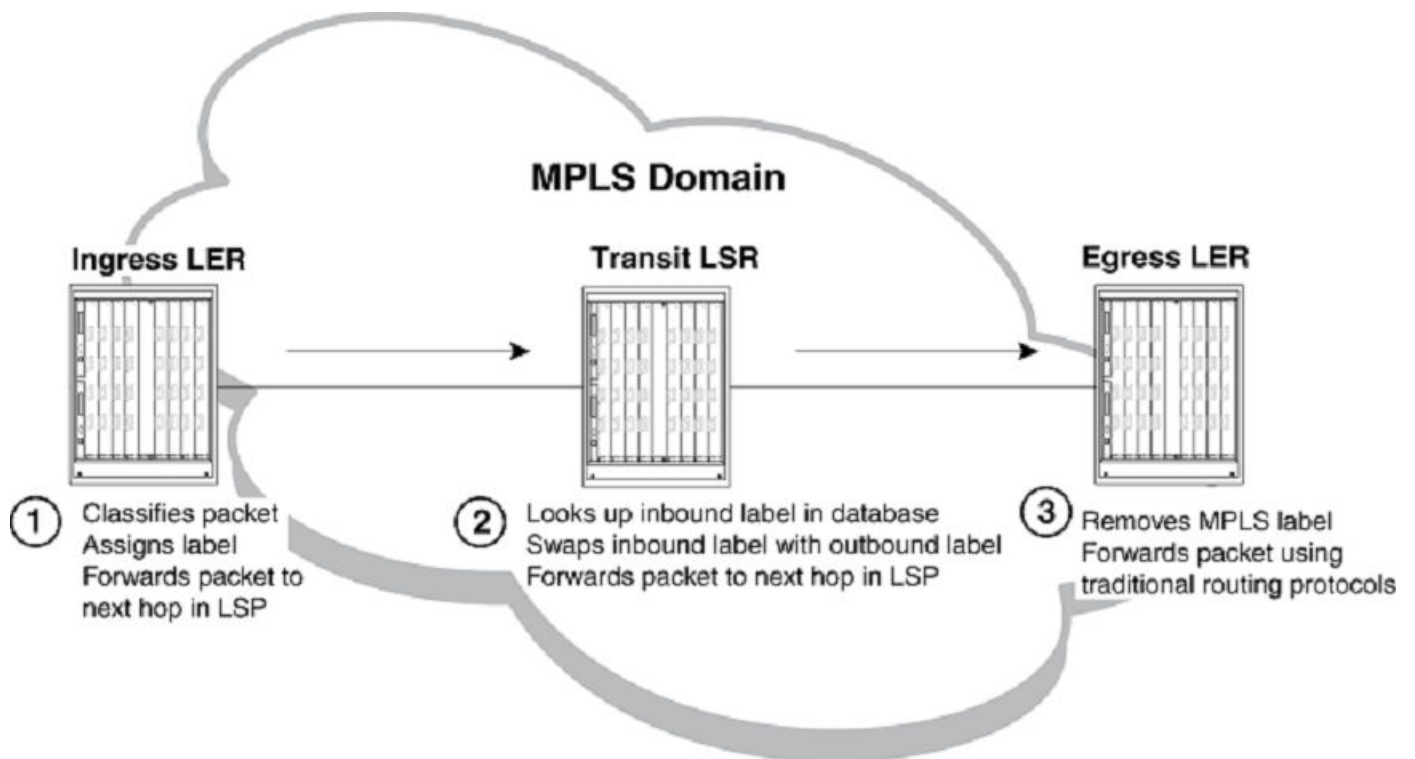
Label switching in an MPLS domain works as described below.

1. The Ingress LER receives a packet and pushes a label onto it.

   When a packet arrives on an MPLS-enabled interface, the device determines to which LSP (if any) the packet are assigned. Specifically, the device determines to which *Forwarding Equivalence Class (FEC)* the packet belongs. An FEC is simply a group of packets that are all forwarded in the same way. For example, a FEC could be defined as all packets from a given *Virtual Leased Line (VLL)*. FECs are mapped to LSPs. When a packet belongs to a FEC, and an LSP is mapped to that FEC, the packet is assigned to the LSP.
   When a packet is assigned to an LSP, the device, acting as an ingress LER, applies (pushes) a tunnel label onto the packet. A label is a 32-bit, fixed-length identifier that is significant only to MPLS. Refer to MPLS label header encoding on page 30 for specific information about the contents of a label. From this point until the packet reaches the egress LER at the end of the path, the packet is forwarded using information in its label, not information in its IP header. The packet's IP header is not examined again as long as the packet traverses the LSP. The ingress LER may also apply a VC label onto the packet based on the VPN application.

   On the ingress LER, the label is associated with an outbound interface. After receiving a label, the packet is forwarded over the outbound interface to the next router in the LSP.

2. A transit LSR receives the labeled packet, swaps the label, and forwards the packet to the next LSR.

   In an LSP, zero or more transit LSRs can exist between the ingress and egress LERs. A transit LSR swaps labels on an MPLS packet and forwards the packet to the next router in the LSP.

   When a transit LSR receives an MPLS packet, it looks up the label in its *MPLS forwarding table* . This table maps the label and inbound interface to a new label and outbound interface. The transit LSR replaces the old label with the new label and sends the packet out the outbound interface specified in the table. This process repeats at each transit LSR until the packet reaches the next-to-last LSR in the LSP (for signaled LSPs).

   Figure 1 illustrates an example of the label swapping process on a transit LSR.

   FIGURE 1 Label swapping on a transit LSR
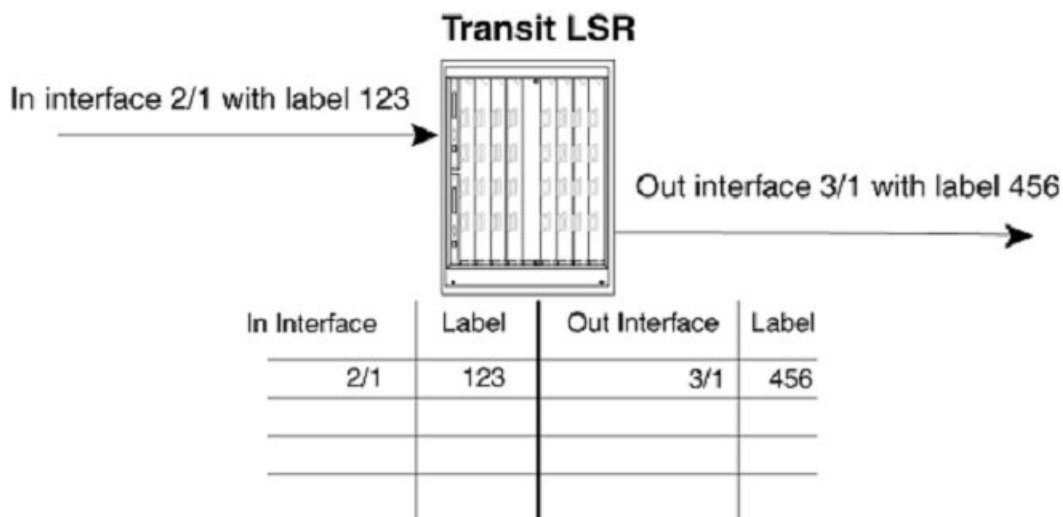
   In this example, a packet comes into interface 2/1 with label 123. The transit LSR then looks up this interface-label pair in its MPLS forwarding table. The inbound interface-label pair maps to an outbound-interface-label pair - in this example, interface 3/1 with label 456. The LSR swaps label 123 with label 456 and forwards the packet out interface 3/1.

3. The egress LER receives labeled packet, pops label, and forwards IP packet.

   When the packet reaches the egress LER, the MPLS label is removed (called *popping* the label), and the packet can then be forwarded to its destination using standard hop-by-hop routing protocols. On signaled LSPs, the label is popped at the penultimate (next to last) LSR, rather than the egress LER. Refer to Penultimate hop popping on page 30 for more information.

Types of LSPs

Signaled LSPs

Signaled LSPs are configured at the ingress LER only. When the LSP is enabled, RSVP signaling messages travel to each LSR in the LSP, reserving resources and causing labels to be dynamically associated with interfaces. When a packet is assigned to a signaled LSP, it follows a pre-established path from the LSPs ingress LER to its egress LER. This path can be one of the following:

- A path that traverses an explicitly specified set of MPLS routers
- The IGP shortest path across the MPLS domain, determined from local routing tables

- A traffic-engineered path calculated by the device using constraints such as bandwidth reservations, administrative groups, and network topology information

Penultimate hop popping

On signaled LSPs, the MPLS label is popped at the next-to-last LSR in the LSP, instead of at the egress LER. This action is called *penultimate hop popping*. Penultimate hop popping improves forwarding efficiency by allowing the egress LER to avoid performing both a MPLS forwarding table lookup and an IP forwarding table lookup for each packet exiting the LSP. Instead, the MPLS label is popped at the penultimate (next-to-last) LSR, and the packet is forwarded to the egress LER with no MPLS encoding. The egress LER, in fact, does not recognize the packet as emerging from an LSP.

Figure 2 illustrates the operation that takes place at the penultimate LSR in an LSP.

FIGURE 2 Penultimate hop popping
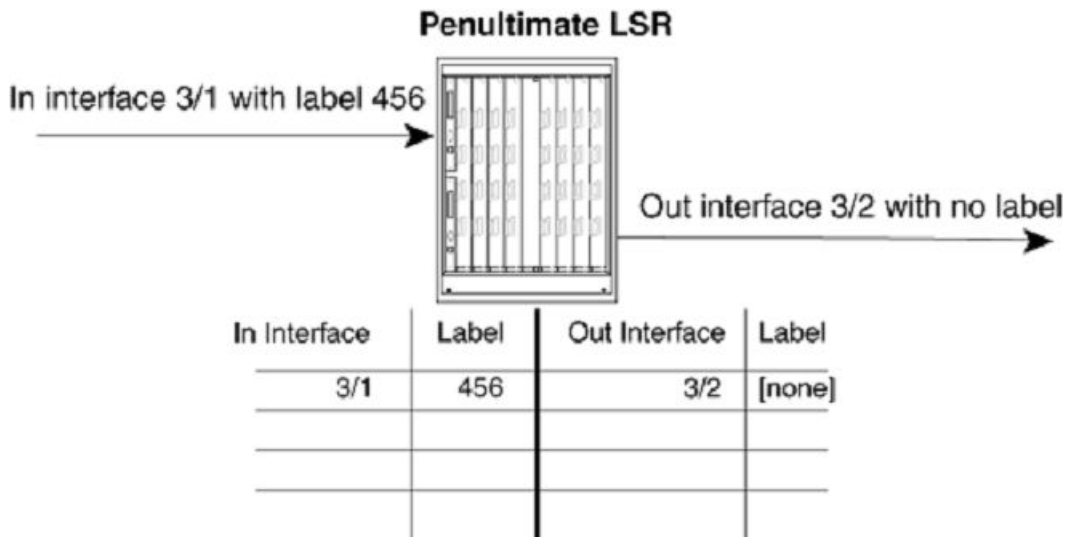
When an LSR receives an MPLS packet, it looks up the label in its MPLS forwarding table. Normally, this table maps the label and inbound interface to a new label and outbound interface. However, when this is the penultimate LSR in an LSP, the label and inbound interface map only to an outbound interface. The penultimate LSR pops the label and forwards the packet - now a regular IP packet - out the outbound interface. When the packet reaches the egress LER, there is no indication that it had been forwarded over an LSP. The packet is forwarded using standard hop-by-hop routing protocols.

MPLS label header encoding

The following diagram illustrates the structure of the 32-bit MPLS label header. When a packet enters an LSP, the ingress LER pushes a label onto the packet.

FIGURE 3 Structure of an MPLS Label Header

An MPLS label header is composed of the following parts:

Label value (20 bits)

The label value is an integer in the range 16 - 1048575. (Labels 0 - 15 are reserved by the IETF for special usage.) For signaled LSPs, the device dynamically assigns labels in the range 1024 - 499999.

EXP field (3 bits)

The EXP field is designated for experimental usage. By default, a device uses the EXP field to define a Class of Service (CoS) value for prioritizing packets traveling through an LSP. Please refer to MPLS Traffic Engineering on page 25, for more information. Note that software forwarded VPLS packets do not use the EXP encode table.

S (Bottom of Stack) field (one bit)

An MPLS packet can be assigned multiple labels. When an MPLS packet has multiple labels, they are logically organized in a last-in, first-out *label stack*. An LSR performs a pop or swap operation on the topmost label; that is, the most recently applied label in the stack. The Bottom of Stack field indicates whether this label is the last (oldest) label in the stack. When the label is the last one in the stack, the Bottom of Stack field is set to one. If not, the Bottom of Stack field is set to zero.

A device acting as an LSR can perform one push, swap, or pop operation on an incoming MPLS packet. The device can accept MPLS packets that contain multiple labels, but only the topmost label is acted upon.

TTL field (eight bits)

The TTL field indicates the *Time To Live (TTL)* value for the MPLS packet. At the ingress LER, an IP packet's TTL value is copied to its MPLS TTL field. At each transit LSR hop, the MPLS TTL value is decremented by one. When the MPLS TTL value reaches zero, the packet is discarded. Optionally, the user can configure the LSRs not to decrement the MPLS TTL value at each hop.

OSPF-TE Link State Advertisements for MPLS interfaces

MPLS-enabled devices running OSPF can be configured to send out LSAs that have special extensions for traffic engineering. These LSAs, called *OSPF-TE LSAs*, contain information about interfaces configured for MPLS. The OSPF-TE LSAs are flooded throughout the OSPF area. LSRs that receive the OSPF-TE LSAs place the traffic engineering information into a TED, which maintains topology data about the nodes and links in the MPLS domain.

Traffic engineering information is carried in OSPF traffic engineering (OSPF-TE) LSAs. OSPF-TE LSAs are Type 10 Opaque LSAs, as defined in *RFC 2370*. Type 10 Opaque LSAs have area flooding scope.

OSPF-TE LSAs have special extensions that contain information related to traffic engineering; these extensions are described in *RFC 3630*. The extensions consist of *Type/Length/Value triplets (TLVs)* containing the following information:

- Type of link (either point-to-point or multi-access network)
- ID of the link (for point-to-point links, this is the Router ID of the LSR at the other end of the link; for multiaccess links, this is the address of the network's designated router)
- IP address of the local interface for the link
- IP address of the remote interface for the link (this could be zero for multicast links)
- Traffic engineering metric for the link (by default, this is equal to the OSPF link cost)
- Maximum bandwidth on the interface
- Maximum reservable bandwidth on the interface
- Unreserved bandwidth on the interface
- Administrative groups to which the interface belongs

When configured to do so, the device sends out OSPF-TE LSAs for each of its MPLS-enabled interfaces. The user can optionally specify the maximum amount of bandwidth that can be reserved on an interface, as well as assign interfaces to administrative groups. Refer to Setting traffic engineering parameters for MPLS interfaces on page 99 for more information.

The following events trigger the device to send out OSPF-TE LSAs:

- Change in the interface's administrative group membership
- Change in the interface's maximum available bandwidth or maximum reservable bandwidth
- Significant change in unreserved bandwidth per priority level:
  - If for any priority level, the difference between the previously advertised unreserved bandwidth and the current unreserved bandwidth exceeds five percent of the maximum reservable bandwidth
  - Any changes while the total reserved bandwidth exceeds 95 percent of the maximum reservable bandwidth

In addition, OSPF-TE LSAs can be triggered by OSPF; for example, when an interface's link state is changed. When an interface is no longer enabled for MPLS, the device stops sending out OSPF-TE LSAs for the interface.

Using MPLS in traffic engineering

Traffic engineering is the task of routing network traffic to avoid points of congestion and make efficient use of high bandwidth interfaces. When used as an application of MPLS, traffic engineering involves creating LSPs that make the best use of available network resources; that is, *traffic-engineered LSPs*. This section explains the process of creating traffic-engineered LSPs.

Creating traffic-engineered LSPs involves the following tasks:

- Gathering information about the network
- Using the gathered information to select optimal paths through the network
- Setting up and maintaining the paths

For traffic-engineered signaled LSPs, devices can perform these tasks dynamically. Figure 4 illustrates the process that takes place to configure, establish, and activate traffic-engineered signaled LSPs.

> **NOTE**
> Adaptive LSPs can have primary and secondary sessions up at the same time. Extreme devices support a maximum of 5000 LSPs, and a maximum of 10000 sessions.

FIGURE 4 How traffic-engineered LSPs are configured, established, and activated
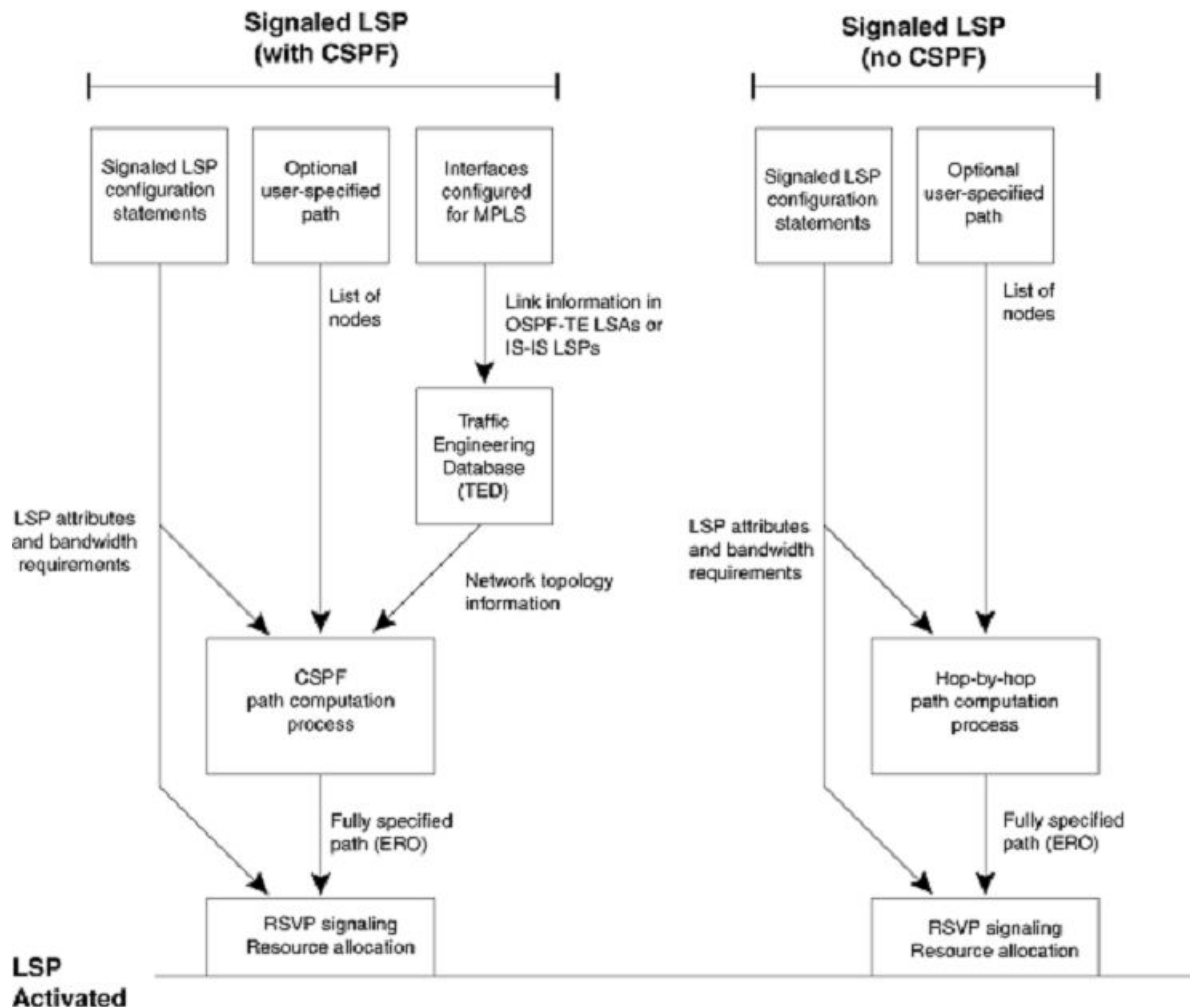
CSPF calculates a traffic-engineered path

When the user configures a signaled Label Switched Path, the user specifies the address of the egress LER, as well as optional attributes, such as the LSPs priority and bandwidth requirements. The user can optionally specify a path of LSRs that the LSP must pass through on the way to the egress LER. When the user enables the signaled LSP, the *Constrained Shortest Path First (CSPF)* process on the ingress LER uses this information to calculate a *traffic-engineered path* between the ingress and egress LERs.

CSPF is an advanced form of the *Shortest Path First (SPF)* process used by IGP routing protocols. The CSPF process on the ingress LER uses the configured attributes of the LSP, user-specified path (when there is one), and the information in the *Traffic Engineering Database (TED)* to calculate the traffic-engineered path. This process consists of a sequential list of the physical interfaces that packets assigned to this LSP pass through to travel from the ingress LER to the egress LER. The traffic-engineered path takes into account the network topology, available resources, and user-specified constraints. The traffic-engineered path calculated by CSPF may or may not be the same as the shortest path that would normally be calculated by standard IGP routing protocols.

CSPF is enabled by default for signaled LSPs, but can be disabled. When signaled LSPs are configured without CSPF, the shortest path from the ingress LER to the egress LER is calculated using standard hop-by-hop routing methods. When the LSP also is configured to

use a user-specified path, the device calculates the shortest path between each LSR in the path. As with CSPF, the output of this process is a fully specified path of physical interfaces on LSRs.

The advantage of configuring signaled LSPs without CSPF is that it can span multiple IS-IS levels. Since IS-IS LSPs with TE extensions have an area and level flooding scope, the information in an LSRs TED is relevant only to their area or level. Consequently, signaled LSPs that use CSPF can span only an IS-IS level. Signaled LSPs that do not use CSPF, because they do not rely on information in the TED, do not have this restriction.

Once the path for the LSP has been calculated, RSVP signaling then causes resources to be reserved and labels to be allocated on each LSR specified in the path. This may cause already existing, lower priority LSPs to be preempted. Once resources are reserved on all the LSRs in the path, the signaled LSP is considered to be activated; that is, packets can be forwarded over it.

The following sections provide additional information about the individual components of the process for activating traffic-engineered signaled LSPs, illustrated in Using MPLS in traffic engineering on page 32.

IS-IS Link State Protocol data units with TE extensions for MPLS interfaces

An MPLS-enabled device running IS-IS can be configured to send out *Link State Protocol (LSP)* data units that contain special extensions to support *Traffic Engineering (TE)*. (In this section -- and nowhere else in this chapter -- LSP is the acronym for Link State Protocol. In other sections, LSP means Label Switched Path.) These LSPs are composed of a fixed header and a number of tuples known as *Type/Length/Value triplets (TLVs)*. LSPs that are used for traffic engineering contain a new object called a sub-TLV. Sub-TLVs are similar to regular TLVs except that, where regular TLVs exist inside IS-IS packets, sub-TLVs reside within regular TLVs. Each sub-TLV consists of three fields: a one-octet Type field, a one-octet Length field, and zero or more octets of Value.

These LSPs are flooded throughout the IS-IS domain. LSRs that receive the IS-IS LSPs with TE extensions place the traffic engineering information into a *Traffic Engineering Database (TED)*, which maintains topology data about the nodes and links in the MPLS domain.

IS-IS LSPs have special extensions that contain information related to traffic engineering and are described in *RFC 3784*. The extensions consist of Type/Length/Value triplets (sub-TLVs) containing the following information:

- IP address of the local interface for the link
- IP address of the remote interface for the link (for point-to-point adjacencies)
- Traffic engineering metric for the link (by default, this is equal to the IS-IS link cost)
- Maximum bandwidth on the interface
- Maximum reservable bandwidth on the interface
- Unreserved bandwidth on the interface
- Administrative groups to which the interface belongs

When configured to do so, the device sends out IS-IS LSPs with TE extensions for each of its MPLS-enabled interfaces. The user can optionally specify the maximum amount of bandwidth that can be reserved on an interface, as well as assign interfaces to administrative groups. Refer to Setting traffic engineering parameters for MPLS interfaces on page 99 for more information.

Any of the following events trigger the device to send out IS-IS LSPs with a TE extension:

- Change in the interface's administrative group membership.
- Change in the interface's maximum available bandwidth or maximum reservable bandwidth.
- Significant change in unreserved bandwidth per priority level, which can be either of the following:
  - For any priority level, the difference between the previously advertised, unreserved bandwidth and the current, unreserved bandwidth exceeds five percent of the maximum reservable bandwidth.

    – Any change when the total reserved bandwidth exceeds 95 percent of the maximum reservable bandwidth.

In addition, IS-IS LSPs with TE extensions can be triggered by IS-IS (for example, when an interface's link state changes). Furthermore, when an interface is no longer enabled for MPLS, the device stops sending out IS-IS LSPs with TE extensions for that interface.

Traffic engineering database

An LSR TED stores topology information about the MPLS domain. This topology information comes from the IS-IS LSPs with TE extensions that are flooded throughout the IS-IS domain. When an LSR receives IS-IS LSPs with TE extensions from neighboring LSRs, it places the traffic engineering information into its TED.

LSP attributes and requirements used for traffic engineering

In addition to the topology information in the TED, the device considers attributes and requirements specified in configuration statements for the LSP. The following user-specified parameters are considered when the device calculates a traffic-engineered path for a signaled LSP:

- Destination address of the egress LER
- Explicit path to be used by the LSP
- Bandwidth required by the LSP
- Setup priority for the LSP
- Metric for the LSP
- Whether the LSP includes or excludes links belonging to specified administrative groups

Refer to Configuring signaled LSP parameters on page 119 for more information on how to set these parameters.

How CSPF calculates a traffic-engineered path

Using information in the TED in addition to the attributes and requirements of the LSP, CSPF calculates a traffic-engineered path for the LSP by performing the tasks listed below.

1. When more than one LSP needs to be enabled, CSPF selects the LSP for path calculation based on the LSPs setup priority and bandwidth requirement.

   When multiple LSPs are enabled simultaneously, such as when the device is booted, CSPF calculates the paths one at a time. CSPF starts with the LSP that has the highest configured setup priority. When more than one LSP has the same setup priority, CSPF calculates the path first for the LSP with the highest configured bandwidth requirement.

2. Eliminate unsuitable links from consideration.

   The device examines the topology information in its TED and uses this information to eliminate links from consideration for the traffic-engineered path. A link is eliminated when any of the following are true:

   - The link is half duplex
   - The link does not have enough reservable bandwidth to fulfill the LSPs configured requirements
   - The LSP has an **include** statement, and the link does not belong to an administrative group in the statement
   - The LSP has an **exclude** statement, and either the link belongs to an administrative group specified in the exclude statement or the link does not belong to any administrative group at all

3. Using the remaining links, calculate the shortest path through the MPLS domain.

    Using the links that were not eliminated in the previous step, the device calculates the shortest path between the ingress and egress LERs. When the LSP is configured to use an explicit path, the device individually calculates the shortest path between each node in the path. Refer to Setting up paths on page 118 for more information on explicit paths.

    By default, the path calculated by CSPF can consist of no more than 255 hops, including the ingress and egress LERs. The user can optionally change this maximum to a lower number. Refer to Limiting the number of hops the LSP can traverse on page 128.

4. When multiple paths have the same cost, select one of them.

    The shortest path calculation performed in the previous step may result in multiple, equal-cost paths to the egress LER. In this case, the device chooses the path whose final node is the physical address of the destination interface.

    When more than one path fits this description, by default, the device chooses the path with the fewest hops. When multiple paths have this number of hops, the device chooses one of these paths at random. The user can optionally configure the device to choose the path that has either the highest available bandwidth or the lowest available bandwidth. Refer to Specifying a tie-breaker for selecting CSPF equal-cost paths on page 128.

    The output of the CSPF process is a traffic-engineered path, a sequential list of the physical interfaces that packets assigned to this LSP pass through to reach the egress LER. Once the traffic-engineered path has been determined, RSVP signaling attempts to establish the LSP on each LSR in the path. Refer to the following section, How RSVP establishes a signaled LSP on page 36, for a description of how this works.

How RSVP establishes a signaled LSP

The traffic-engineered path calculated by CSPF consists of a sequential list of physical interface addresses, corresponding to a path from the ingress LER to the egress LER. Using this traffic-engineered path, RSVP establishes the forwarding state and resource reservations on each LSR in the path.

Special extensions for traffic engineering are defined for RSVP. These extensions include the EXPLICIT_ROUTE, LABEL_REQUEST, LABEL, and RECORD_ROUTE objects in addition to the *Fixed Filter (FF)* reservation style. These extensions are described in *RFC 3209* .

The following diagram illustrates how RSVP establishes a signaled LSP.

FIGURE 5 How RSVP establishes a signaled LSP
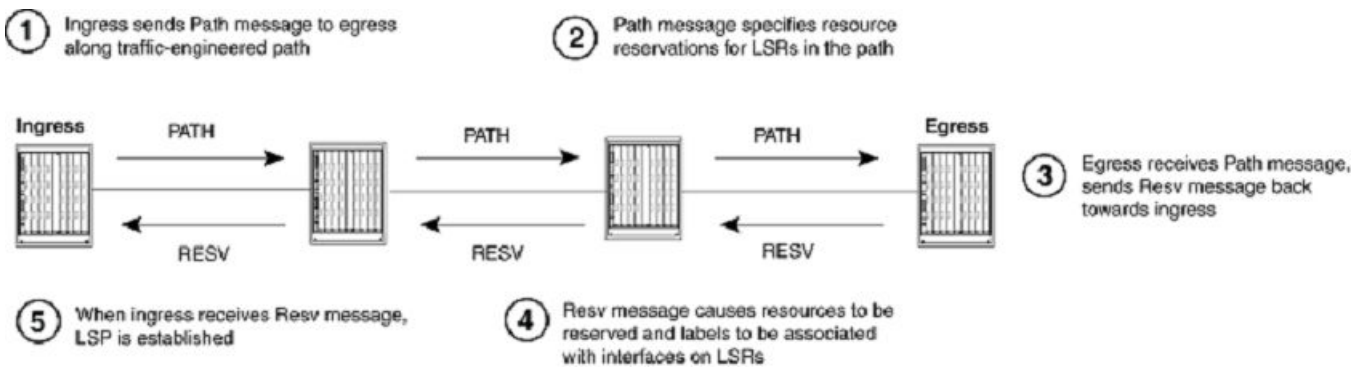
RSVP signaling for LSPs works as described below.

1. The ingress LER sends an RSVP Path message towards the egress LER.

   The Path message contains the traffic engineered path calculated by the CSPF process, specified as an *EXPLICIT_ROUTE object (ERO)* . The Path message travels to the egress LER along the route specified in the ERO.

   The Path message also describes the traffic for which resources are being requested and specifies the bandwidth that needs to be reserved to accommodate this traffic. In addition, the Path message includes a LABEL_REQUEST object, which requests that labels be allocated on LSRs and tells the egress LER to place a LABEL object in the Resv message that it sends back to the ingress LER.

   Before sending the Path message, the ingress LSR performs admission control on the outbound interface, ensuring that enough bandwidth can be reserved on the interface to meet the LSPs requirements. Admission control examines the LSPs configured setup priority and mean-rate settings. For the LSP to pass admission control, the outbound interface must have reservable bandwidth at the LSPs setup priority level that is greater than the amount of bandwidth specified by the LSPs mean-rate setting. Refer to Admission control, bandwidth allocation, and LSP preemption on page 39, for more information and examples of this process.

2. The Path message requests resource reservations on the LSRs along the path specified in the ERO.

   When the LSP passes admission control, the ingress LER sends a Path message to the address at the top of the ERO list. This is the address of a physical interface on the next LSR in the path. As the ingress LER did, this LSR performs admission control to make sure the outbound interface has enough reservable bandwidth to accommodate the LSP.

   When the LSP passes admission control, the LSR then removes its address from the top of the ERO list and sends the Path message to the address now at the top of the ERO list. This process repeats until the Path message reaches the last node in the ERO list, which is the egress LER.

3. The egress LER receives the Path message and sends a Resv message towards the ingress LER.

   Resv messages flow upstream from the receiver of the Path message to the sender (that is, from the egress LER to the ingress LER), taking the exact reverse of the path specified in the ERO. In response to the LABEL_REQUEST object in the Path message, the Resv message from the egress LER includes a LABEL object. The LABEL object is used to associate labels with interfaces on the LSRs that make up the LSP.

4. As the Resv messages travel upstream, resources are reserved on each LSR.

   When an LSR receives a Resv message, it again performs admission control on the interface where the Resv message was received (that is, the interface that is the outbound interface for packets traveling through the LSP). When the LSP still passes admission control, bandwidth is allocated to the LSP. The LSR allocates the amount of bandwidth specified by the LSPs mean-rate setting, using bandwidth available to its hold priority level. This may cause lower priority LSPs active on the device to be preempted.

   Once bandwidth has been allocated to the LSP, the LABEL object in the Resv message is used to associate labels with interfaces in the LSRs MPLS forwarding table. Figure 6 shows an example of how this works.

   FIGURE 6 How the RSVP LABEL object associates a label with an interface in the MPLS forwarding table
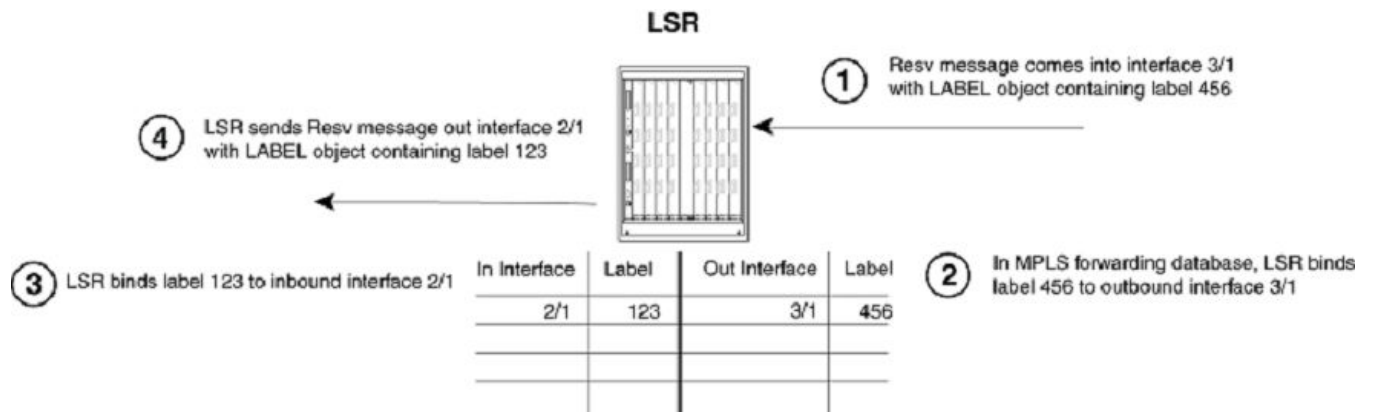
   In the example above, the LSR receives a Resv message on interface 3/1 from the downstream LSR in the ERO. The Resv message has a LABEL object containing label 456. After performing admission control and bandwidth allocation, the LSR adds an entry to its MPLS forwarding table for this LSP, associating label 456 with outbound interface 3/1.

   The LSR then takes a label from its range of available labels (for example, 123) and places it in the LABEL object in the Resv message that it sends to the upstream LSR. In this example, the LSR sends the Resv message out interface 2/1 to the upstream LSR in the ERO. In its MPLS forwarding table for this LSP, the LSR associates label 123 with inbound interface 2/1.

   This process repeats at each LSR until the Resv message reaches the ingress LER.

   > NOTE
   > To enable penultimate hop popping for the LSP, the LABEL object sent by the egress LER to the penultimate LSR contains a value of three (3) (Implicit Null Label). This is an IETF-reserved label value that indicates to the penultimate LSR that it must pop the label of MPLS-encoded packets that belong to this LSP.

5. Once the Resv message reaches the ingress LER, and the process described in Step 4 takes place, the LSP is activated. At this point each LSR in the LSP has reserved resources, allocated labels, and associated labels with interfaces. The LSP is activated, and the ingress LER can assign packets to the LSP.

Refresh messages

Once a signaled LSP is enabled at the ingress LER, the router persistently attempts to establish the LSP through periodic retries until the LSP is successfully established. To maintain the forwarding states and resource reservations on the routers in an LSP, Path and Resv messages are exchanged between neighboring LSRs at regular intervals. When these refresh messages are not received on the routers in the LSP, the RSVP forwarding states and resource reservations are removed. The user can control how often the Path and Resv messages are sent, as well as how long the device waits before removing forwarding states and resource reservations. The user can also use reliable messaging and refresh reduction to reduce RSVP message bandwidth and improve the dependability of RSVP paths and reservations states.

Admission control, bandwidth allocation, and LSP preemption

When a Resv message is received on an LSR, admission control determines whether the LSP can be established, based on its configured priority. When an LSP passes admission control, bandwidth is allocated to the new LSP, possibly preempting existing LSPs that have lower priority.

An LSPs priority consists of a setup priority and a hold priority. The setup priority is the priority for taking resources; the hold priority is the priority for holding resources. An LSPs setup priority is considered during admission control, and its hold priority is considered when bandwidth is allocated to the LSP. The setup and hold priorities are expressed as numbers between zero (0) (highest priority level) and seven (7)(lowest priority level). An LSPs setup priority must be lower than or equal to its hold priority. The user can configure either of these values for an LSP; by default, an LSPs setup priority is seven and its hold priority is zero.

On an MPLS-enabled interface, a certain amount of bandwidth is allocated for usage by LSPs; this amount can be either the maximum available bandwidth on the interface (the default) or a user-specified portion. The amount of bandwidth an individual LSP can reserve from this pool of allocated bandwidth depends on two user-configured attributes of the LSP: the LSPs priority and the LSPs mean-rate (the average rate of packets that can go through the LSP). The following conditions also apply:

- For an LSP to pass admission control, the bandwidth available to its setup priority level must be greater than the value specified by its mean-rate.
- When an LSP passes admission control, the bandwidth specified by its mean-rate is allocated to the LSP, using bandwidth available to its hold priority level.
- For the allocation of bandwidth to the new LSP, the system might preempt existing, lower-priority LSPs.

When setting up an LSP, the device actually performs admission control twice: when the Path message is received and when the Resv message is received. when the LSP passes admission control after the Resv message is received, bandwidth allocation and LSP preemption take place.

The sections that follow include examples of how admission control, bandwidth allocation, and preemption work.

Admission control

Admission control examines the LSPs setup priority and mean-rate settings to determine whether the LSP can be activated. To pass admission control, the reservable bandwidth available at the LSPs setup priority level must be greater than the value specified by its mean-rate.

For example, when the maximum reservable bandwidth on an interface is 10,000 Kbps and no LSPs are currently active, the amount of reservable bandwidth on the interface for each priority level would be as follows:

Priority	Unreserved Bandwidth
0	10,000
1	10,000
2	10,000

3	10,000
4	10,000
5	10,000
6	10,000
7	10,000
Active LSPs : None	

The LSR receives a Resv message for an LSP that has a configured setup priority of six and a hold priority of three. The mean-rate specified for this LSP is 1,000 Kbps. For priority level 6, up to 10,000 Kbps can be reserved. Because the configured mean-rate for this LSP is only 1,000 Kbps, the new LSP passes admission control.

Bandwidth allocation

Once the LSP passes admission control, bandwidth is allocated to it. The bandwidth allocation procedure examines the LSPs hold priority and mean-rate settings. The amount of bandwidth specified by the mean-rate is allocated to the LSP, using reservable bandwidth available at the LSPs hold priority level.

In this example, the LSPs hold priority is three and mean-rate is 1,000 Kbps. On this interface, for priority level three, up to 10,000 Kbps can be reserved. The amount of bandwidth specified by the mean-rate (1,000 Kbps) is allocated to the LSP.

After bandwidth is allocated to this LSP, the amount of unreserved bandwidth on the interface is reduced accordingly. In the example, the reservable bandwidth array for the interface now looks like this:

Priority	Unreserved Bandwidth
0	10,000
1	10,000
2	10,000
3	9,000
4	9,000
5	9,000
6	9,000
7	9,000
Active : LSP with setup 6, hold 3, mean-rate 1,000	

Given the bandwidth allocation above, when an LSP is established with a setup priority of three and a mean-rate of 9,500 Kbps, it would not pass admission control because only 9,000 Kbps is available at priority 3.

LSP preemption

When there is not enough unallocated bandwidth on an interface to fulfill the requirements of a new LSP that has passed admission control, existing LSPs that have a lower priority may be preempted. When preemption occurs, bandwidth allocated to lower-priority LSPs is reallocated to the higher-priority LSP. LSP preemption depends on the bandwidth requirements and priority of the new LSP, compared to the bandwidth allocation and priority of already existing LSPs.

When LSP preemption is necessary, the device uses the following rules:

NOTE
LSP preemption rules have changed to improve the scalability and performance for *Fast Reroute (FRR)* enabled LSPs. See bullets three and four below for changes to LSP preemption for FRR enabled LSPs.

- Preempt existing LSPs that have lower priority than the new LSP
- When several existing LSPs have lower priority than the new LSP, preempt the LSP that has the lowest priority
- When two LSPs have equal priority and one LSP must be preempted, preempt the LSP which is currently FRR enabled irrespective of its bandwidth requirement
- Preempt as many FRR enabled LSPs as necessary before preempting unprotected LSPs of the same priority. For example, when both FRR enabled LSPs and non-FRR enabled LSPs are configured, the system attempts its best to preempt FRR enabled LSPs first before preempting non-FRR enabled LSPs until the bandwidth requirement is met for a new high priority LSP

In the example above, bandwidth has been allocated to an LSP that has a hold priority of three and a mean-rate of 1,000 Kbps. When a new LSP with a setup priority of two, hold priority of one, and mean-rate of 10,000 Kbps is established, admission control, bandwidth allocation, and LSP preemption work as described below.

1. **Admission control:** On the interface, there is 10,000 Kbps available to priority two. The mean-rate for the new LSP is 10,000, so the LSP passes admission control; bandwidth can be allocated to it.
2. **Bandwidth allocation:** The hold priority for the new LSP is one. On the interface, 10,000 Kbps is available to priority one. This entire amount is allocated to the LSP.
3. **LSP preemption:** The first LSP had been using 1,000 Kbps of this amount, but its hold priority is only three. Consequently, the first LSP is preempted, and its bandwidth allocation removed in order to make room for the new LSP.

   Once this happens, the reservable bandwidth array for the interface looks like this:

Priority	Unreserved Bandwidth
0	10,000
1	0
2	0
3	0
4	0
5	0
6	0
7	0
Active : LSP with setup 2, hold 1, mean-rate 10,000	
Preempted: LSP with setup 6, hold 3, mean-rate 1,000	

On this interface, the only LSP that could preempt the active LSP would be have a setup and hold priority of zero.

When multiple LSPs are candidates for preemption, the device normally preempts the LSP with the lowest priority. However, when preempting a higher priority LSP with a high bandwidth requirement would allow lower priority LSPs with lower bandwidth requirements to avoid preemption, the higher-priority LSP is preempted.

For example, consider an interface with 10,000 Kbps of reservable bandwidth, allocated to two active LSPs: one with a setup priority of three, hold priority of two, and mean-rate of 5,000 Kbps; and another with a setup priority of four, hold priority of three, and mean-rate of 2,500 Kbps. When an LSP with a setup priority of one, hold priority of zero, and mean-rate of 7,500 Kbps is established, the following take place.

Calculating a path based on an interface address

Under normal conditions, router IDs are used to configure hops within an MPLS path. In situations where the user wants to exercise more control over the path, the user can specify actual interface addresses in the MPLS path to make sure that the path traverses the interface specified. In previous versions, the CSPF calculation would always resolve a specified interface address to the router ID. Consequently, although a particular interface on a router is specified, the CSPF calculation can end up connecting the path through a different interface on the router where the interface has been specified.

In the network described in the diagram below, the source node is "A" and the destination node is "E". In this configuration, incoming and outgoing interfaces are defined in the figure by their relationship to where the arrowhead on the connecting line points. The arrowheads point to the incoming interface from the outgoing interface. For instance "A1", "A2" and "A3" are the outgoing interfaces of node A and "C1" and "C2" are the incoming interfaces of node C. The following example describes how the router might calculate a path between "A" and "B" under the default operating condition.

In this example, an MPLS path has been configured with a source "A" and a destination "E1". Under default operation, the interface "E1" destination is resolved to the routerID for "E". This means that the path can be calculated to arrive at the "E" node on any of the following interfaces: "E1", "E2" or "E3". While a path that travels from node "A" to node "B" to node "E" is the only path that actually satisfies the intent of the configuration, any of the following paths could be created by CSPF under the default operation condition: "A" to "C" to "E", "A" to "D" to "C" to "E" or "A" to "D" to "F" to "E".

FIGURE 7 Calculating a path based on an interface
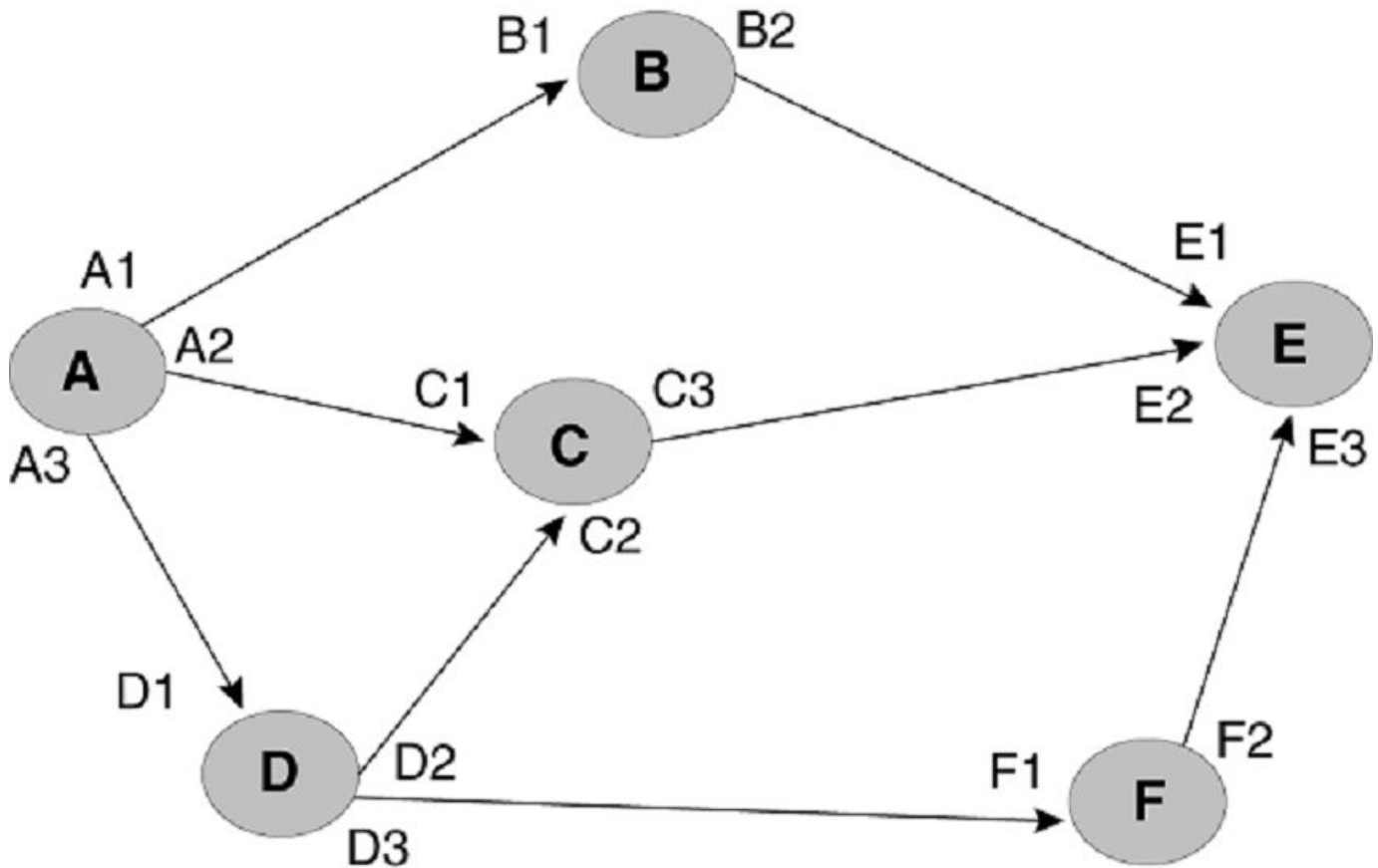
The global **cspf-interface-constraint** command directs the router to include the interface address as a constraint when it determines the shortest path. When invoked, this command ensures that a specified interface must be included in an LSP. This constraint can be turned

on and off dynamically and does not affect established primary or secondary LSPs. CSPF interface constraint is significant for the ingress node only, where CSPF calculation takes place for an LSP.

When configuring CSPF interface constraint, the user must be aware that the imposition of this additional constraint can increase the possibility of no path being found where otherwise there could be a path. One case where this can occur is where the path required to conform to the interface constraint fails the configured bandwidth constraint. Additionally, no path may be found where a configured path contains an inherently contradictory condition. For example, when a path is configured "B1 (strict) to E2 (loose) as shown in the diagram below, no path is found. This is because CSPF always appends B1 into the final CSPF path. This has the effect of making "B" the source node of the next hop and therefore excludes "E1 as a traversed interface in subsequent paths to the destination node "E". Consequently, in this example the LSP is down. However, when the **cspf-interface-constraint** command is not active, a CSPF path is found and the LSP goes up.

FIGURE 8 Example of where no path is found
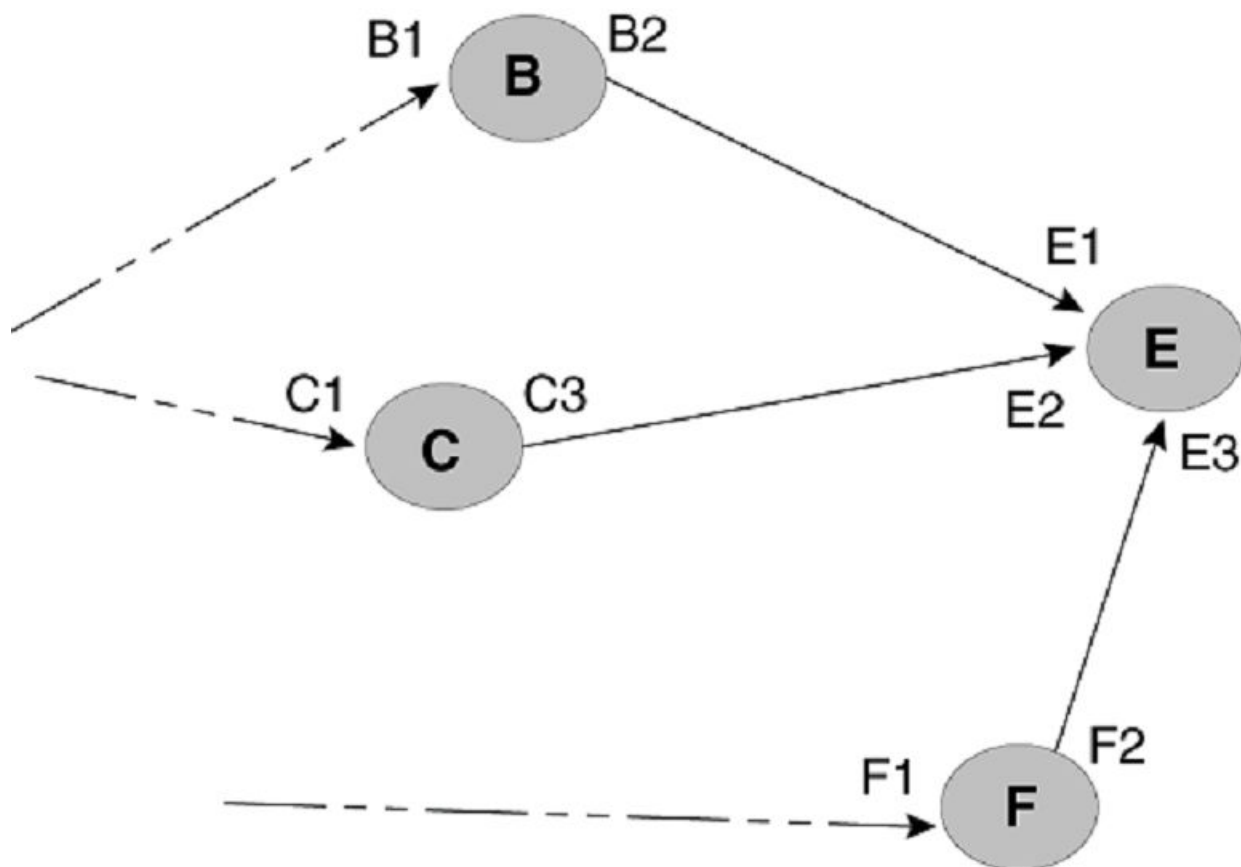
Displaying the Traffic Engineering database

An LSRs *Traffic Engineering Database (TED)* contains topology information about nodes in an MPLS domain and the links that connect them. This topology information is obtained from the IS-IS LSPs with traffic engineering extensions. IS-IS LSPs with TE extensions have special extensions that contain information about an MPLS-enabled interface's traffic engineering metric, bandwidth reservations, and administrative group memberships.

An LSR, when configured to do so, floods IS-IS LSPs with TE extensions for its MPLS-enabled interfaces to its neighboring routers in the IS-IS area. Other LSRs store the information from the IS-IS LSPs with TE extensions in their own *Traffic Engineering Databases*

(TED), allowing each LSR in the area to maintain an identical TED describing the MPLS topology. The topology information in the TED is used by the CSPF process when it calculates traffic-engineered paths for signaled LSPs.

Displaying a traffic engineering path to a destination

The user can display a traffic engineering path to a IPv4 destination address using a specified set of resource parameters. This allows the user to gain insight into a traffic engineering path in a network, before setting it up using RSVP. This helps the user to avoid a RSVP path setup failure due to unavailable requested resources along the path to the destination host.

T o display the traffic engineering path to a IPv4 destination address, enter the following command.

```
device# show mpls te path 10.4.4.4
```

The following example displays an output from the show mpls te path command.

```
device# show mpls te path 10.12.12.12 hop-limit 2
Path to 10.12.12.12 found! Time taken to compute: 0 msec
Hop-count: 2 Cost: 2000 ISIS Level-1
     Hop 1: 10.1.0.1, Rtr 10.13.13.13
     Hop 2: 10.1.0.2, Rtr 10.12.12.12
```

Displaying signaled LSP status information

The user can display status information about signaled LSPs for which the device is the ingress LER as shown in the example below.

```
device# show mpls lsp
*: The LSP is taking a Secondary Path
Admin Oper    Tunnel    Up/Dn Retry Active
Name    To          State State Intf    Times No.    Path
t1      10.3.3.3    UP        UP*    tnl1    1        5      v2
```

> **NOTE**
> The **show mpls lsp brief** command displays the same information as the **show mpls lsp** command.

The **show mpls lsp detail** command displays detailed information about a specific LSP. To display detailed information about the status of the LSPs for which the device is the ingress LER, enter the **show mpls lsp detail** command as shown in the example below.

```
device# show mpls lsp detail
LSP t1, to 10.3.3.3
 Path selected: pathsecondaryb, mode: manual revert-timer:
 Path selected is up for 3 seconds for the latest instance, traffic will be switched to it in 7 seconds.
 From: 10.2.3.4, admin: UP, status: UP, tunnel interface: tnl1
 Times primary LSP goes up since enabled: 1
 Metric: 1, number of installed aliases: 0
 Maximum retries: 0, no. of retries: 3
 Pri. path: dir, active: no
  Setup priority: 7, hold priority: 0, ipmtu 1400
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
 Sec. path: v2, active: active
  Hot-standby: no, status: up
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes hop limit: 0
 Active Path attributes:
  Tunnel interface: tnl1, outbound interface: e1/1
  Tunnel index: 5, outbound label: 1966
  Path calculated using constraint-based routing: no
  Explicit path hop count: 1
   10.10.10.2 (S) -> 10.20.20.2 (S) Recorded routes:
  Protection codes: P: Local    N: Node    B: Bandwidth    I: InUse
   10.10.10.2 -> 10.20.20.2
```

Displaying path information

A path is a list of router hops that specifies a route across an MPLS domain. The user can create a path and then configure the LSPs that see the path. When the LSP is enabled, the ingress LER attempts to signal the other LSRs in the path, so that resources can be allocated to the LSP.

The user can display information about the paths configured on the device as shown in the following example.

```
device# show    mpls    path
Path Name        Address          Strict/loose    Usage Count
to110_120        10.110.110.2     Strict          1
                 10.120.120.3     Strict
to2_pri          10.10.10.2       Strict          0
to2_sec          10.110.110.2     Strict          0
to3              10.110.110.2     Loose           1
                 10.120.120.3     Loose
to3_pri          10.10.10.2       Strict          1
                 10.20.20.3       Strict
to3_sec          10.110.110.2     Strict          0
                 10.120.120.3     Strict
to4              10.110.110.2     Loose           1
                 10.120.120.3     Loose
                 10.130.130.4     Loose
to_23            10.110.110.2     Strict          1
                 10.20.20.3       Strict
```

Displaying the MPLS routing table

The MPLS routing table is used to store routes to egress LERs.

To display the contents of the MPLS routing table, enter the **show mpls route** command. The port field displays whether an interface/port is either Ethernet or POS.

Displaying the MPLS forwarding information

The **show mpls forwarding** command displays MPLS forwarding information. The 'out-intf' field in the output of the **show mpls forwarding** command displays whether an interface/port is either an Ethernet port or a POS port.

Displaying the P2MP hardware forwarding information

The **show mpls lsp_p2mp_xc** command displays information about the forwarding information of hardware that is allocated for the point-to-multipoint (P2MP) cross-connect.

For additional information, see the **show mpls lsp_p2mp_xc** command in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying RSVP information

The user can display RSVP version information, the status of RSVP interfaces, RSVP session information, and RSVP statistics.

Displaying the RSVP version

To display the RSVP version number, as well as the refresh interval and refresh multiple, use the **show mpls rsvp** command.

Displaying the status of RSVP interfaces

Use the **show mpls rsvp interface** command to display the status of RSVP on devices where it is enabled.

For additional information, see the **show mpls rsvp interface** CLI command in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying RSVP session information

To display RSVP session information, use the **show mpls rsvp session** command. For additional information regarding this command, see *NetIron Command Line Interface (CLI) Reference Guide*.

To display the entire LSP name on one line, use the **show mpls rsvp session wide** command. For additional information on this command, see *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying RSVP P2MP session information

The **show mpls rsvp session p2mp** command filters the RSVP sessions based on the P2MP LSP type.

For additional information on the **show mpls rsvp session p2mp** command, go to the **show mpls rsvp session p2mp** command in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying RSVP statistics

The device constantly gathers RSVP statistics. RSVP statistics are collected from the time RSVP is enabled, as well as from the last time the RSVP statistics counters were cleared.

For additional information, see the **show mpls rsvp statistics** command in *NetIron Command Line Interface (CLI) Reference Guide*.

To clear the RSVP statics counters, use the following command:

```
device# clear mpls rsvp statistics
```

Syntax: clear mpls rsvp statistics

This command resets the counters listed under "since last clear" for the **show mpls rsvp interface detail** and **show mpls rsvp statistics** commands.

Displaying RSVP session destination information

To display information about the RSVP session destination, use the **show mpls rsvp session destination** command.

For additional information, go to the **show mpls rsvp session destination** command in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying information about OSPF-TE LSAs

To display information about OSPF-TE LSAs.

```
device# show ip ospf database link-state opaque-area
Area ID Type    LS ID        Adv Rtr     Seq(Hex)    Age    Cksum
0       OpAr    10.0.0.0     10.3.3.3    80000006    1337   0x1a19
  Area-opaque TE LSA
  1 - router address (len 4): 10.3.3.3
Area ID Type    LS ID        Adv Rtr     Seq(Hex)    Age    Cksum
0       OpAr    10.0.0.2     10.2.2.2    80000007    1333   0x88f1
  Area-opaque TE LSA
  2 - link (len 100):
    1 - link type (len 1): point-to-point(1)
    2 - link ID (len 4): 10.1.1.1
    3 - local i/f ip addr (len 4): 10.1.1.2
    4 - remote i/f ip addr (len 4): 10.1.1.1
    5 - TE metric (len 4):
    6 - max BW (len 4): 2372 Mbits/sec
    7 - max reservable BW (len 4): 2372 Mbits/sec
    8 - unreserved BW (len 32):
      Priority 0: 2372 Mbits/sec
      Priority 1: 2372 Mbits/sec
```

```
     Priority 2: 2372 Mbits/sec
     Priority 3: 2372 Mbits/sec
     Priority 4: 2372 Mbits/sec
     Priority 5: 2372 Mbits/sec
     Priority 6: 2372 Mbits/sec
     Priority 7: 2372 Mbits/sec
  9 - color (len 4): 0
```

Syntax: show ip ospf database link-state opaque-area

Displaying information about IS-IS LSPs with TE extensions

To display information about IS-IS LSPs with TE extensions.

```
device# show isis database level2 detail
IS-IS Level-2 Link State Database
LSPID                  LSP Seq Num  LSP Checksum  LSP Holdtime     ATT/P/OL
SLX-OS3.00-00            0x00000644  0x78e3          843              1/0/0
   Area Address:  49.0002
   NLPID:  CC(IP)
   Hostname:  SLX-OS3
   Auth: Len 17 MD5 Digest "c33db90a87b93c80111980dbd59a19ed"
   TE  Router ID: 15.15.15.15
   Metric:  10       IP-Extended 15.15.15.15/32     Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 132.0.0.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 121.0.0.0/24       Up: 0 Subtlv: 0
   Metric:  10       IS-Extended PE4.06
     Admin Group: 0x00000000
     Interface IP Address: 121.0.0.2
     Link BW: 1000000 kbits/sec
     Reservable BW: 1000000 kbits/sec
     Unreserved BW:
       [0]    1000000 kbits/sec  [1]    1000000 kbits/sec
       [2]    1000000 kbits/sec  [3]    1000000 kbits/sec
       [4]    1000000 kbits/sec  [5]    1000000 kbits/sec
       [6]    1000000 kbits/sec  [7]    1000000 kbits/sec
   Metric:  10       IS-Extended SLX-OS4.00
     Admin Group: 0x00000000
     Interface IP Address: 132.0.0.2
     Neighbor IP Address: 132.0.0.1
     Link BW: 10000000 kbits/sec
     Reservable BW: 10000000 kbits/sec
     Unreserved BW:
       [0]   10000000 kbits/sec  [1]   10000000 kbits/sec
       [2]   10000000 kbits/sec  [3]   10000000 kbits/sec
       [4]   10000000 kbits/sec  [5]   10000000 kbits/sec
       [6]   10000000 kbits/sec  [7]   10000000 kbits/sec
```

Displaying MPLS Fast Reroute information

The following sections describe how to get information about MPLS Fast Reroute:

The commands used to display MPLS and RSVP information are described elsewhere in this document. This section describes the display of information about MPLS Fast Reroute

Displaying MPLS Fast Reroute LSP information

To display MPLS Fast Reroute LSP information for a protected LSP that uses one-to-one (detour) backup.

```
device# show mpls lsp frr_tunnel
LSP frr_tunnel, to 10.4.4.4
  From: 10.1.1.1, admin: UP, status: UP, tunnel interface: tnl4
```

```
        Times primary LSP goes up since enabled: 1
        Metric: 0, number of installed aliases: 0
        Maximum retries: 0, no. of retries: 0
        Pri. path: p1, active: yes
        Path specific attributes:
         Tunnel interface: tnl4, outbound interface: e1/1
         Setup priority: 7, hold priority: 0
         Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
         Constraint-based routing enabled: yes
         Tie breaking: random, hop limit: 0
         Explicit path hop counts: 3
          11.1.1.2 (S) -> 13.1.1.2 (S) -> 15.1.1.2 (S)
         Recorded routes:
          Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
          11.1.1.2 (PNB) -> 13.1.1.2 (PNB) -> 15.1.1.2
        Fast Reroute: one-to-one backup desired
         Bandwidth: 1024 kbps
         Detour LSP: UP, out-label: 1028, outbound interface: e1/3
```

Output that is shown in bold is unique to the **show mpls lsp** command when the LSP is configured for Fast Reroute by way of detour backup. The output is described in Table 2. Fields that are common to the output from the **show mpls lsp** command when an LSP is not configured for Fast Reroute are described in Displaying signaled LSP status information on page 44.

TABLE 2 Output from the **show mpls lsp** command

This field...	Displays...
Fast Reroute	The method of Fast Reroute configured for this LSP. Currently only one-to-one backup is available.
Bandwidth	The bandwidth in Kilobits per second for the bypass route. A value of 0 means that the detour route uses a best effort value for bandwidth.
Detour LSP	Indicates when the detour route is Up or Down.
out-label	The outbound label used when sending traffic over a detour LSP.
outbound interface	The physical interface on the router that is used for the detour route.

Displaying adaptive bypass LSP information

Enter the **show mpls bypass-lsp name** command to display information about a bypass LSP.

```
device# show mpls bypass-lsp name t100
LSP t100, to 10.1.1.1
  From: 10.2.2.2, admin: UP, status: UP
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0   Adaptive
  Maximum retries: NONE, no. of retries: 0
  Pri. path: NONE, up: no, active: no
  Setup priority: 7, hold priority: 0 ReoptimizeTimer: 300
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
   Path calculated using constraint-based routing: no
   Path calculated using interface constraint: no
  Tie breaking: random, hop limit: 0
  Active Path attributes:
```

For additional information regarding this command, see the **show mpls bypass-lsp** command in *NetIron Command Line Interface (CLI) Reference Guide.*

Displaying RSVP fast reroute session information

Displays RSVP fast reroute session information for a protected LSP.

NOTE
This section provides a brief example of the display from the **show mpls rsvp session name frr_tunnel** command.

```
device# show mpls rsvp session name frr_tunnel
 Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
        DE:Egress Detour  RP:Repaired Session

 To          From         St   Style Lbl_in Lbl_out LSPname
 10.4.4.4    10.1.1.1(DI) Up   SE    -      1028    frr_tunnel
   Time left in seconds (PATH refresh: 1, ttd: 4294621
                         RESV refresh: 20, ttd: 156)
   Tspec: peak 0 kbps rate 1024 kbps size 0 bytes m 20 M 65535
   Explicit path hop count: 3
    12.1.1.2 (S) -> 18.1.1.2 (S) -> 15.1.1.2 (S)
   Received RRO count: 3
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    12.1.1.2 -> 18.1.1.2 -> 15.1.1.2
   Detour Sent: Number of PLR and Avoid Node ID pair(s): 1
    [1]: PLR: 11.1.1.1  Avoid Node: 11.1.1.2
   PATH sentto:  12.1.1.2        (e1/3          ) (MD5 OFF)
   RESV rcvfrom: 12.1.1.2        (e1/3          ) (MD5 OFF)

 To          From         St   Style Lbl_in Lbl_out LSPname
 10.4.4.4    10.1.1.1     Up   SE    -      1029    frr_tunnel
   Time left in seconds (PATH refresh: 6, ttd: 146
                         RESV refresh: 20, ttd: 140)
   Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
   Fast Reroute: one-to-one backup desired
    Setup priority: 7, hold priority: 0
    Bandwidth: 1024 kbps, hop limit: 255
    Detour LSP: UP.  Nexthop (node) protection available.
     Up/Down times: 1, num retries: 0
   Explicit path hop count: 3
    11.1.1.2 (S) -> 13.1.1.2 (S) -> 15.1.1.2 (S)
   Received RRO count: 3
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    11.1.1.2 (PNB) -> 13.1.1.2 (PNB) -> 15.1.1.2
   PATH sentto:  11.1.1.2        (e1/1          ) (MD5 OFF)
   RESV rcvfrom: 11.1.1.2        (e1/1          ) (MD5 OFF)
```

For additional information, see the **show mpls rsvp session** command in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying MPLS configuration information

The **show runing router mpls** command displays all of the user-configured MPLS parameters. Using the **show runing router mpls** command, the user can display all of the following global parameters configured on an Extreme device:

- brief
- cspf-group
- interface
- lsp
- path

Displaying in the brief mode

In this mode, the information under router mpls policy, RSVP, LDP, MPLS OAM configuration (BFD), dynamic bypass global configuration, and SNMP traps are displayed as shown in the following:

```
device# show mpls config brief
policy
  admin-group m2 2
  traffic-eng isis level-1
  no propagate-ttl
```

```
    retry-limit 22
rsvp
  refresh-interval 40
  refresh-multiple 25
ldp
hello-interval 20 hello-timeout 50
hello-interval target 20
ka-interval 18
advertise-labels for 5
session 10.30.30.6 key 1 $!dZ@
targeted-peer 10.44.44.20
tunnel-metric 5000
graceful-restart
graceful-restart reconnect-time 200
graceful-restart recovery-time 100
graceful-restart max-neighbor-reconnect-time 100
graceful-restart max-neighbor-recovery-time 75
bfd
  min-tx 100 min-rx 200 multiplier 20
dynamic-bypass
  enable
  enable-all-interfaces
  max-bypasses 25
  max-bypasses-per-mp 5
  reoptimize-timer 20000
lsp-xc-traps enable
end of MPLS configuration
```

For additional information, see *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying in the detail mode

The user can display all of the MPLS global information and all of the MPLS configuration information using the **show run router mpls** command. The **show run router mpls** command displays all of the detailed information.

```
device# show run router mpls
router mpls
 policy
  admin-group m2 2
  traffic-eng isis level-1
  retry-limit 22 rsvp
  refresh-interval 40
 rsvp
  refresh-interval 40

 ldp
  hello-timeout 12 ka-interval 18
  advertise-fec  list1
  session 10.30.30.6 key 1 $!dZ@

 mpls interfaces
  mpls-interface e1/1
  ldp-enable

 mpls-interface e1/2
 ldp-enable
 reservable-bandwidth percentage 80
 admin-group 2

mpls paths
path mu1_to_mu3
 strict 10.1.1.1
 strict 10.1.1.2
 strict 10.3.3.1
 strict 10.3.3.2
path mu1_to_mu2_2
 strict 10.5.1.1
 strict 10.5.1.2
path mu1_to_mu2_1
```

```
 strict 10.1.1.1
 strict 10.1.1.2
lsp frr1
 to 10.4.2.1
 cos 6
ipmtu 1028
traffic-eng max-rate 180 mean-rate 125
metric 5
enable

lsp lsp13d
 to 10.3.3.2
 primary mu1_to_mu3
 cos 7
 traffic-eng max-rate 250 mean-rate 120 no cspf
 enable

lsp lsp12d
 to 10.1.1.2
 cos 7
 traffic-eng max-rate 100 mean-rate 50
 enable

end of MPLS configuration
```

Displaying MPLS configuration information for a VE interface

The **show running router mpls mpls-interface ve** *num* command to display specific MPLS interface configuration information. When MPLS is configured on a VE interface, the VE interface ID is displayed in the output.

The command allows the user to display configuration information for an MPLS-enabled interface. The user can specify a VE interface on the CLI. The following example displays CLI commands executed for the interface ve 20.

```
device# show running router mpls mpls-interface ve 20
  mpls-interface ve 20
  ldp-enable
```

Displaying filtered MPLS configuration information

An individual MPLS interface, LSP, VLL, or VPLS can be specified in the **show run router mpls** command to display configuration of the specified object only. The following example displays the MPLS configuration information for the LSP named "frr1".

```
device# show run router mpls lsp frr1
 lsp frr1
  to 10.4.2.1
  cos 6
  ipmtu 1028
  traffic-eng max-rate 180 mean-rate 125
  metric 5
  frr
    bandwidth 80
    hop-limit 55
  enable
```

When an option is used without a variable specified, the configuration parameters for the option are shown for all elements that match the option are displayed. For instance, in the following example the **lsp name** option is used without a specified *lsp-name* variable. Consequently, the display contains the configuration information for all three LSPs configured on the router.

```
device# show run router mpls lsp
lsp frr1
  to 10.4.2.1
  cos 6
  ipmtu 1028
  traffic-eng max-rate 180 mean-rate 125
  metric 5
  enable
```

```
lsp lsp13d
 to 10.3.3.2
 primary mu1_to_mu3
 cos 7
 traffic-eng max-rate 250 mean-rate 120
 no cspf
 enable
```

MPLS Point-to-Multipoint Traffic Engineering

The MPLS *Point-to-Multipoint (P2MP)* feature enables forwarding of information from a single source to multiple destinations along an optimized MPLS path. P2MP feature is ideal for transporting multicast data traffic, leveraging MPLS, and using optimal bandwidth utilization of the network's links.

The P2MP feature supports the following RFCs:

- *RFC 4875*: Extensions to *Resource Reservation Protocol - Traffic Engineering (RSVP-TE)* for Point-to-Multipoint TE *Label Switched Paths (LSPs)*
- *RFC 4461*: Signaling Requirements for Point- to-Multipoint Traffic-Engineered MPLS *Label Switched Paths (LSPs)*
- *RFC 2961*: RSVP Refresh Overhead Reduction Extensions
- *RFC 3209*: RSVP-TE: Extensions to RSVP for LSP Tunnels
- *RFC 2205*: *Resource ReSerVation Protocol (RSVP)* -- Version 1 Functional Specification

Figure 9 shows the network topology of a P2MP network.

FIGURE 9 MPLS P2MP network topology
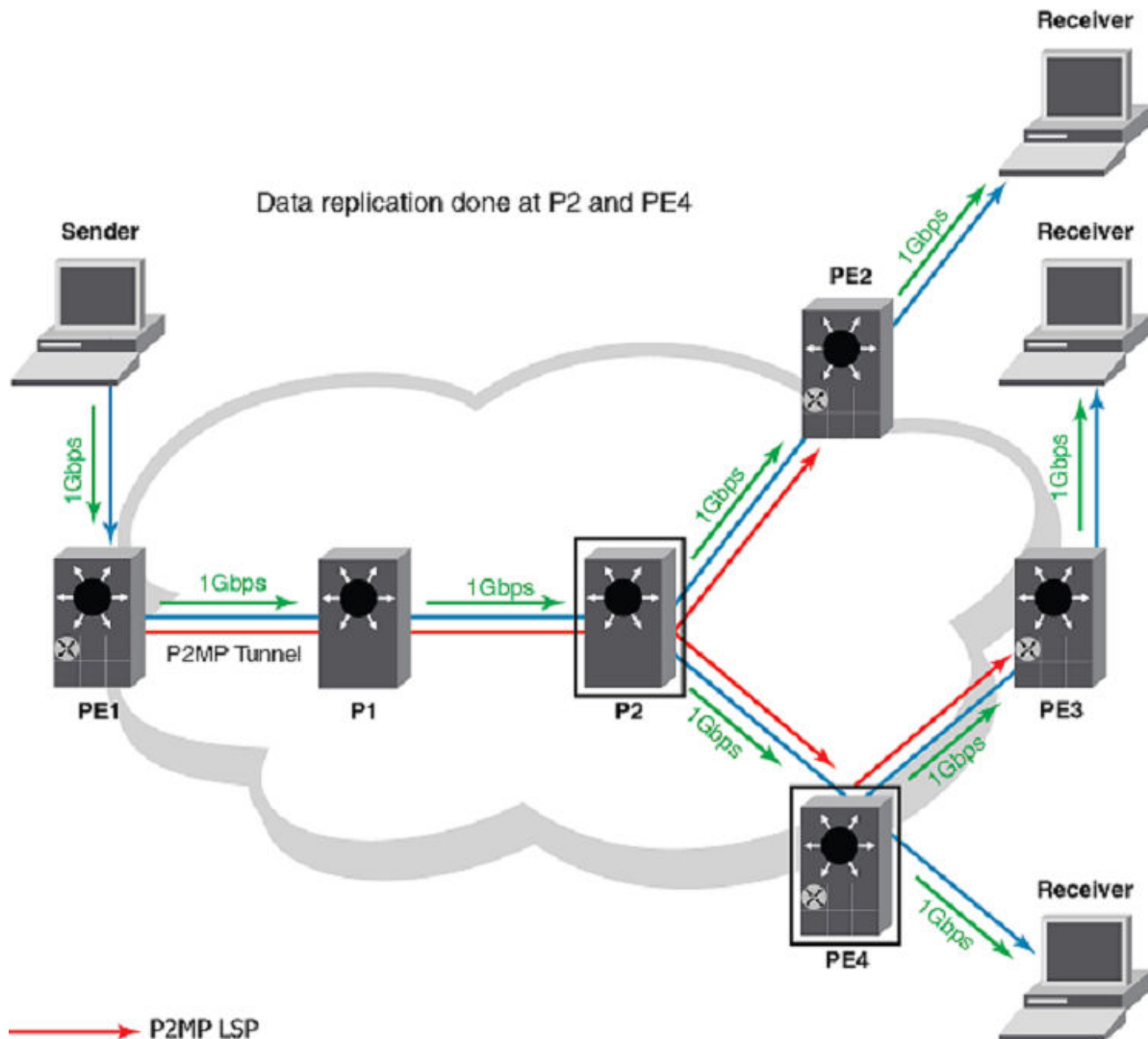
Figure 9 displays a P2MP LSP originating at PE1 and ending at the three destinations PE2, PE3, and PE4. As depicted in the topology diagram, the bandwidth utilization across the network is 1 Gbps.

The P2MP network consists of the following key elements in the topology.

- PE1: The source or the root or the ingress Label Switch Router (LSR)
- P1 and P2: The transit routers
- PE2, PE3, and PE4: The destination or the leaves or the egress LSRs
- P2 is known as the branch node since it is has more than one directly connected downstream LSRs
- PE4 is known as the bud node since it has directly connected local receivers. It also acts like a branch router
- The path of a P2MP LSP from its ingress LSR to all egress LSRs is known as the P2MP tree. In Figure 9, the P2MP tree is rooted at PE1 with leaves at PE2, PE3 and PE4

P2MP LSP mechanism

The P2MP LSP mechanism of forwarding MPLS traffic from a single source to multiple destinations is explained using MPLS Point-to-Multipoint Traffic Engineering on page 52.

1. PE1 sends out multicast data of 1 Gbps to the destination LSRs. Let us assume that PE1 learns that PE2, PE3, and PE4 form a P2MP LSP tree.
2. PE1 sends multicast data to reach P2. The path taken by the LSP is PE1-->P1-->P2.
3. At P2 and PE4, data gets replicated and the multicast data reaches the egress routers.
4. The egress routers route the multicast data to the specified receivers.

Prerequisites and limitations

- P2MP feature supports transit and branch functionality only.
- FPGA-based second generation line cards support P2MP LSP feature. This feature is not supported by FPGA-based first generation line cards and 24x10g module line cards. In a network deployment with a router having both first generation and second generation line cards, the feature provides the following responses to specific scenarios:
  - Router receives packets on a first generation port -- The control packet is dropped and an error message (if applicable) is returned to the ingress so that the user can find an alternative path for the P2MP LSP
  - Router transmits packets on a first generation port -- The control packet is dropped and an error message (if applicable) is returned to the ingress so that the user can find an alternative path for the P2MP LSP
  - LAG and VE ports with either generation line cards -- If a P2MP control message is received on a LAG or VE, or sent by a LAG or VE, that has at least one member port of FPGA-based first generation line card, the control message is dropped and/ or an error message is returned
  - Modifying LAG or VE on the fly -- While adding a first generation member port to an existing LAG or VE port, if the LAG or VE is an incoming or outgoing port for at least one P2MP LSP, then all associated P2MP LSPs are torn down
- Each P2MP LSP requires one *Mapped VLAN ID (MVID)* at the transit router. When the system runs out of MVID resources, a new incoming Path message for P2MP session is rejected and a Path error message is sent back indicating that the system is out of resource
- Any P2MP LSPs transiting through the router no longer comes up if there is an image downgrade.
- The following P2MP features are not supported:
  - Ingress features
  - Egress features
  - Bud node features
  - Data traffic load balancing over LAG
  - *Fast ReRoute (FRR)*
  - LSP stitching and LSP hierarchy
  - Soft preemption

Scalability limitations

- Maximum number of P2MP sessions -- The number of P2MP sessions supported at a transit or branch router is limited by the replication entry resource. There are 2000 replication entries available on a MLX Series device. Therefore, the maximum number of P2MP sessions supported is 2000.
- Maximum number of total branches -- Each FPGA XPP image supports a maximum of 8000 replications. That is, one incoming packet can be replicated into 8000 copies by one XPP. Depending on the number of XPPs on a line card, the total

number of branches supported will vary. For example, on a 8x10 module with two network processors, there are a maximum of 16000 branches supported.

- Maximum number of branches per P2MP LSP -- There is no limit on number of branches per P2MP LSP. However, considering the linear behavior as detailed in *RFC 4461* about number of egress points and branch LSRs, the standard recommendation is 64 branches per P2MP.

- Maximum number of LSPs and XCs -- On the CER 2000 Series device, the maximum number of LSPs are 512 with 4000 S2Ls. On the CES 2000 Series device, the maximum number of LSPs are 128 with 1000 S2Ls.

Benefits of P2MP

The benefits of P2MP feature are:

- Efficiently uses the bandwidth in the network
- Improves the overall capacity of the network
- Efficiently transports broadcast, multicast, and unknown unicast packets

P2MP LSP Traffic-Engineering (TE) Constraints

P2MP LSPs can be configured with various TE constraints to control the path that it should take. The following TE constraints are applicable to P2MP LSP:

- Bandwidth
- Priority
- Least-fill/most-fill/random tie-breaking when there are ECMP path for the LSP available
- Administrative groups
- Explicit path (user-configured) with strict/loose hops, and so on

Use case scenario: Transit LSR application for one-to-many applications

MLX Series routers can be deployed as core transit routers for customers with multiple vendor devices that act as PE devices. In such a deployment scenario, when P2MP LSP applications are deployed, the MLX Series routers behave as transit and branch routers for the P2MP LSP network. Applications such as content distribution, financial services, video distribution, IP multicast distribution, IPTV, and others, use P2MP topologies for efficient data traffic routing. Customers who deploy these services can expect ExtremeRouting MLX Series transit routers to support P2MP branch functionality which efficiently replicates the incoming data to egress nodes.

The following deployment scenario indicates MLX Series routers in the core of the network with other vendor devices at the edge (PE) acting as ingress, egress, and bud nodes. In Figure 10, PE1 is the ingress node for the P2MP LSP while PE2, PE3, and PE4 are egress nodes.

FIGURE 10 MLX Series router in transit LSR application
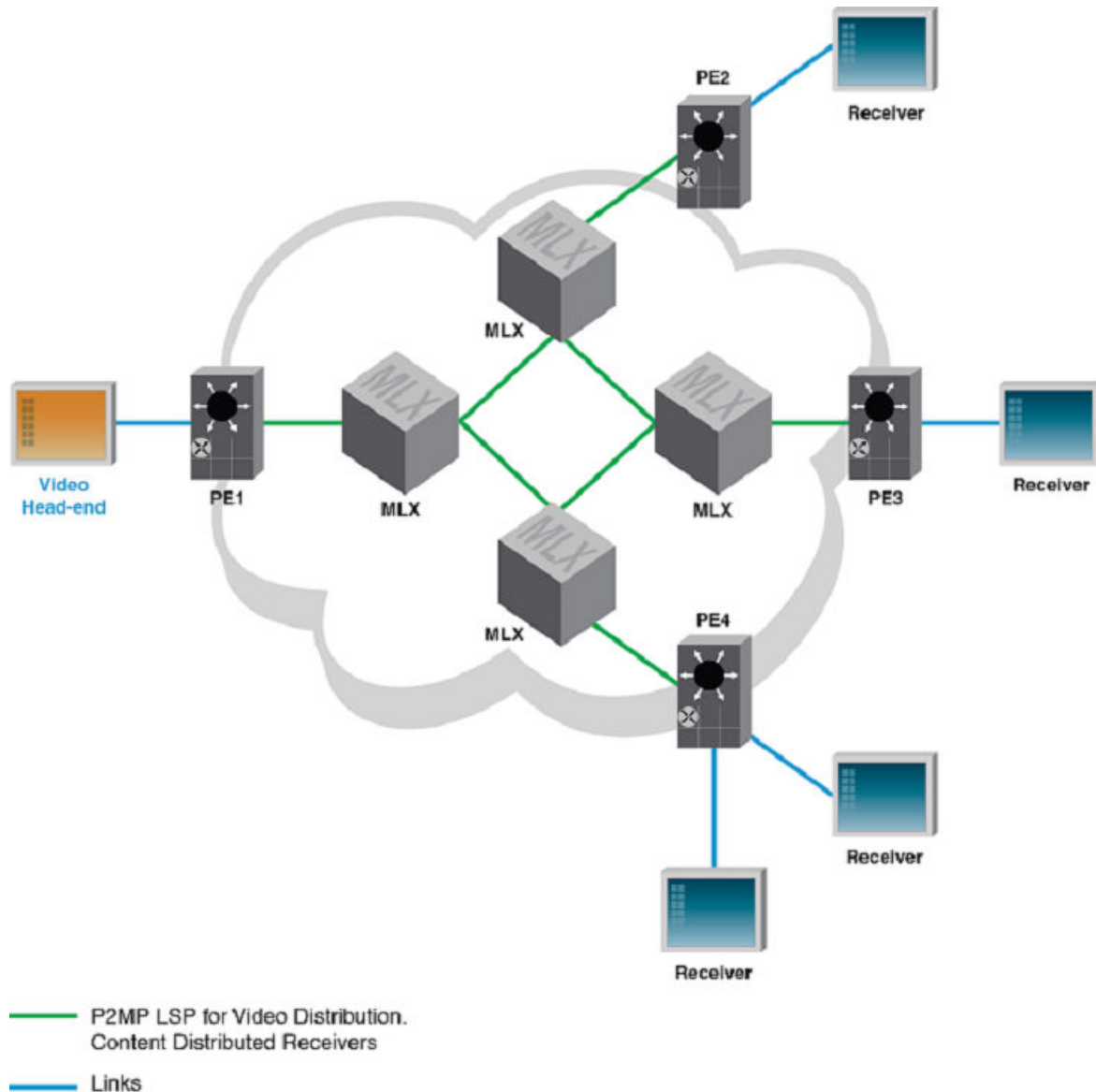
Source-to-leaf sub-LSP
======================

The P2MP LSP topology as observed in MPLS Point-to-Multipoint Traffic Engineering on page 52 consists of multiple point-to-point LSPs that originate from the source device and terminate at a destination device or a leaf. These LSPs are known as *source-to-leaf (S2L)* sub-LSPs. These S2L sub-LSPs can either be signaled in individual Path and Resv messages, or can be combined into a small number of messages.

> NOTE
> Extreme devices support only one S2L signaling per Path message and reject Path messages if there are more than one S2L LSPs from other vendor routers.

Figure 11 indicates three S2L sub-LSPs -- S2L1, S2L2, and S2L3, in a P2MP tunnel.

FIGURE 11 P2MP S2L sub-LSPs
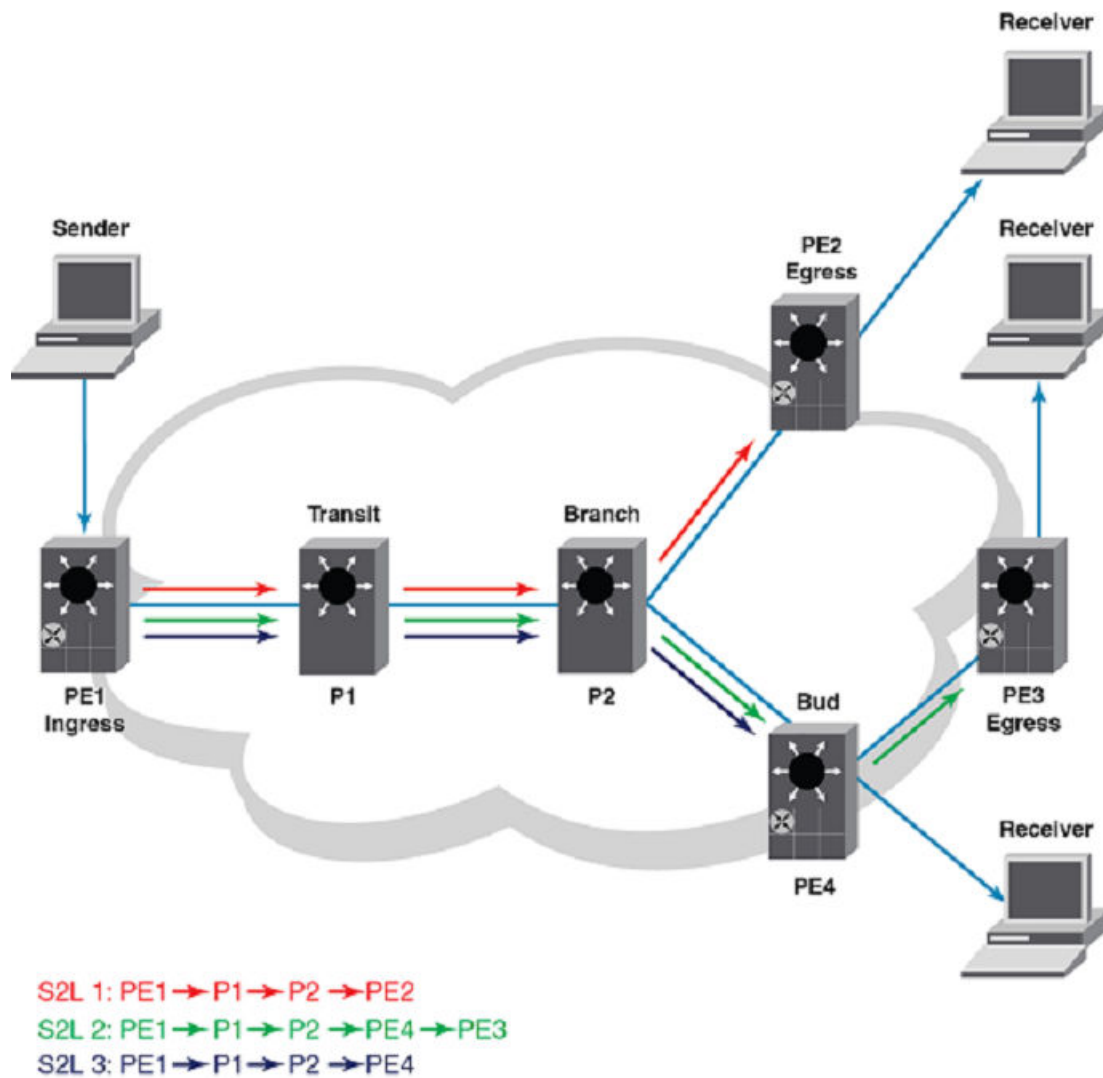
Though S2L sub-LSPs are signaled in separate Path and Resv messages, they are always part of the same P2MP LSP. LSRs such as P2 in Figure 11, handle multiple incoming S2L sub-LSPs on the same interface, allocates a single label and advertises it to LSR P1. It avoids unnecessary duplication of traffic in the data plane.

S2L sub-LSP groups

S2L sub-LSPs can further be grouped into sub-groups using a node that is part of the P2MP LSP. Such sub-groups should be signaled in a separate Path message or Resv message. In Source-to-leaf sub-LSP on page 56, P2 groups S2L2 and S2L3 into a sub-group. Each sub-group is identified by the sub-group originator ID (LSR P2) and the sub-group ID (assigned by LSR P2). Extreme devices support only one S2L sub-LSP per Path or Resv message.

Grafting

The process of adding new egress LSRs to an existing P2MP LSP is known as grafting. Grafting can be achieved using any one of the following methods:

- Implicit -- Adding new S2L sub-LSPs to an existing Path message and refreshing the entire Path message
- Explicit -- Adding egress LSRs by signaling only the impacted or the new S2L sub-LSPs in a new Path message

    NOTE
    Extreme devices support the explicit method of adding LSRs only and do not support the implicit method.

Pruning

The process of removing egress LSRs from an existing P2MP LSP is known as pruning. It allows removal of egress nodes from a P2MP LSP at different points in time. Pruning can be achieved using any one of the following signaling methods:

- Implicit S2L Sub-LSP Teardown -- Sending a modified Path message that includes all S2L sub-LSPs except the one that is being pruned.
- Explicit S2L Sub-LSP Teardown -- Sending a Path Tear message for the corresponding Path message. Path Tear message contains P2MP session-object and sender template object to uniquely identify any S2L Sub-LSP that is being pruned.

    NOTE
    Extreme devices support the explicit method of pruning LSRs only and do not support the implicit method.

RSVP refresh reduction support to P2MP

RSVP refresh reduction feature support is extended to P2MP LSPs. The following refresh reduction extensions are applicable to P2MP.

- Reliable messaging
- Guaranteed message delivery (ACK, Retransmission)
- Message bundling
- Combined multiple RSVP messages into one
- Summary refresh

RSVP soft preemption

RSVP soft preemption implements a suite of protocol modifications extending the concept of preemption with the goal of reducing or eliminating traffic disruption of TE LSPs. It is achieved by using additional signaling and maintenance mechanisms to alert the ingress LER of the preemption that is pending and allows for temporary control-plane under-provisioning while the preempted tunnel is rerouted in a non-disruptive fashion (make before-break) by the ingress LER. During the period that the tunnel is being rerouted, link capacity is under-provisioned on the midpoint where preemption was initiated and potentially one or more links upstream along the path where other soft preemptions may have occurred. Soft preemption is a property of the LSP and is disabled by default.

The default preemption in an MPLS-TE network is hard preemption. This is helpful in cases where actual resource contention happens in the network. Soft Preemption provides flexibility for operators to select the type of preemption based on network conditions.

MPLS soft preemption is useful for network maintenance. For example, all LSPs can be moved away from a particular interface, and then the interface can be taken down for maintenance without interrupting traffic. MPLS soft preemption is also useful in dynamic networks where preemption often occurs.

Only adaptive and non-FRR LSPs could be enabled for soft preemption. LSPs which are adaptive and without FRR configuration have the facility to enable or disable the soft preemption feature without disabling the LSP. When the soft preemption configuration is changed, RSVP is notified for this change and a new Path message is triggered with the soft preemption desired flag bit (0x40) set in session attribute for signaling.

Frequently used terms

Point of preemption – The midpoint (transit) or ingress LSR which, due to RSVP provisioning levels, is forced to either hard preempt or under-provision and signal soft preemption.

Hard preemption – The (typically default) preemption process in which lower priority TE LSPs are intrusively displaced at the point of preemption by high priority TE LSP. In hard preemption, the TE LSP is torn down before reestablishment.

Soft preemption – Soft preemption attempts to establish a new path for a preempted LSP before tearing down the original LSP.

Under-provisioned bandwidth (BW) – Amount of BW which is released as a result of preemption but not yet actuated from the data plane point of view. It is actuated when the preempted LSP is torn down as a result of the new MBB setup or hard preemption is triggered by point of preemption.

Soft preemption wait timer value – The soft preemption wait timer value is the time in seconds for which the point of preemption router waits to receive the Path tear message for the preempted session. In case the Path Tear is not received from the ingress and the wait timer expires, point of preemption router again initiates the path error message intimating hard preemption of this session with error code two, error value zero.

Configuration considerations

- Only adaptive LSPs could be configured with soft preemption capability.
- No FRR enabled LSP can be enabled for soft preemption. Vice versa is also true.
- No guarantee of preempting soft preempt-able LSP first. The protected FRRLSP is preempted first and then the unprotected LSPs. The behavior remains the same except, in unprotected LSPs which requests soft preemption would be soft preempted.
- In the case of an LSP passing through a series of routers running mixed releases with soft preemption enabled, the user may see delayed propagation of the soft preemption bit downstream from the first router.
- When bit 0x40 is not propagated to all downstream routers, soft preemption behavior does not occur. The same is true when de-configuring soft preemption. Soft preemption may occur when the 0x40 bit reset signaling has not propagated to the point of preemption on a downstream router.

Upgrade and downgrade considerations

- Downgrading Ingress router to lower release, soft preemption configuration is lost.
- Downgrading transit-router to lower releases where MPLS soft preemption functionality is not supported, the user will see hard preemption behavior for all LSPs, irrespective of soft preemption desired request, where this router is acting as point of preemption.

Configuring RSVP soft preemption

Soft preemption capability on unprotected adaptive LSPs (which is disabled by default) can be configured irrespective of its state (enable or disable).

Non-adaptive and/or FRR enabled LSPs cannot be configured with soft preemption capability. In this scenario, the LSP must be disabled first to configure soft preemption based on the policies, other changes also may be required, such as removing FRR.

All secondary paths configured on the LSP would be allowed to have soft preemption configured independently.

The following steps must be followed to configure soft preemption.

1. Configure LSP
2. Configure LSP as adaptive
3. Configure primary path, when intended
4. When FRR is configured, remove the FRR configuration
5. Configure soft preemption
6. Configure secondary paths
7. For each secondary path, where soft preemption is intended to be configured, mark them adaptive, when already not adaptive, configure SOFT preemption
8. Enable LSP

RSVP soft preemption configuration example

The **soft-preemption** command enables soft preemption functionality. This command *must* be used on both, the primary and secondary paths.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp test
device(config-router-mpls-lsp-test)# to 10.1.1.100
device(config-router-mpls-lsp-test)# traffic-eng mean-rate 100
device(config-router-mpls-lsp-test)# adaptive
device(config-router-mpls-lsp-test)# soft-preemption
device(config-router-mpls-lsp-test)# secondary
device(config-router-mpls-lsp-test)# secondary-path sec
device(config-router-mpls-lsp-test)# traffic-eng mean-rate 100
device(config-router-mpls-lsp-test-secpath-sec)# adaptive
device(config-router-mpls-lsp-test-secpath-sec)# soft-preemption
device(config-router-mpls-lsp-test-secpath-sec)# enable
Connecting signaled LSP test
```

Detailed command information

Soft-preemption

The **soft-preemption** command enables soft preemption functionality. This command must be used on both, the primary and secondary paths.

Primary path example

```
device# configure
device(config)# router-mpls
device(config-router-mpls)# lsp test
device(config-router-mpls-lsp-test)# soft-preemption
```

Syntax: [no] soft-preemption

The [**no**] function disables soft preemption for the path on which the command is executed.

Secondary path example

```
device# configure
device(config)# router-mpls
device(config-router-mpls)# lsp test
device(config-router-mpls-lsp-test)# secpath sec
device(config-router-mpls-lsp-test-secpath-sec)# soft-preemption
```

Soft-preemption cleanup-timer

Use the **soft-preemption cleanup-timer** command is used to set the amount of time that the point of preemption must wait to receive the Path tear notification from the ingress LSR, before sending a hard preemption path error.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# soft-preemption cleanup-timer 30
```

Syntax: [no] soft-preemption cleanup-timer *decimal-value*

The *decimal-value* is the time the point of preemption wait must to receive the Path tear notification from the ingress LSR, before sending a hard preemption path error. Values ranging from 1 - 29 are not valid values for this timer. The default setting is 30 seconds. The acceptable range for this timer is 30 - 300. Zero indicates soft preemption is disabled on the router.

The [**no**] function returns the timer value settings to the default setting (30 seconds).

Scalability

With the default value (30 seconds) configured for the soft preemption wait timer, the following scalability numbers have been collected under the following conditions:

- System was running MPLS with few interfaces enabled with MPLS and OSPF as traffic engineering
- LSPs were configured on the routers
- Soft preemption was triggered by decreasing the BW on transit router/high priority LSP
- All the routers were soft preemption enabled with default wait timer configuration
- Alternate paths were available for MBB of LSPs impacted

TABLE 3 Maximum soft preempted LSPs

Device Type	Maximum soft preempted LSPs
XMR Series (with MP++)	2000
MLX Series	1000
CES 2000 Series	128
CER 2000 Series	500

Behavioral impact on changing the wait timer value:

- Timer value zero (0) means no soft preemption.
- Timer value 30 is minimum which can be configured other than 0, for which above data is applicable.
- Timer value >30 configuration shows better performance in terms of number of LSPs which can get successfully soft preempted provided other criteria are met for MBB. It is expected to be almost directly proportional to the amount by which timer value is increased. For example, 60 seconds is double the number of LSPs/maximum configurable LSPs on device under test. The user must verify for accuracy.

RASLOG messages

The following notification RASLOG messages are logged under the conditions indicated. No additional traps are generated.

1. When first path error requesting soft preemption is received for an LSP, following message is printed to RASLOG.

    ```
    Dec  9 23:58:49 Brocade MPLS: LSP test soft preemption triggered. Preemption point 10.1.1.20
    ```

2. When MBB is successful for make before break setup of soft preemption requested LSP, following message is printed to RASLOG.

    ```
    Dec  9 23:58:49 Brocade MPLS: LSP test using path NULL soft preempted with make before break
    ```

Auto-bandwidth for RSVP LSPs

Automatic bandwidth allocation allows an MPLS tunnel to automatically adjust its bandwidth allocation based on the volume of traffic flowing through the tunnel. The new bandwidth is determined by inspecting the traffic flowing through the LSP.

The user can configure an LSP with minimal bandwidth. With this feature, the user can dynamically adjust the LSPs bandwidth allocation based on current traffic patterns. The bandwidth adjustments do not interrupt traffic flow through the tunnel.

At the end of the automatic bandwidth adjustment time interval, the current maximum average bandwidth usage is compared with the allocated bandwidth for the LSP. When the LSP needs more bandwidth, an attempt is made to set up a new path where bandwidth is equal to the current maximum average usage. When the attempt is successful, the LSPs traffic is routed through the new path and the old path is removed. When the attempt fails, the LSP continues to use its current path.

FIGURE 12 Basic auto-bandwidth functionality
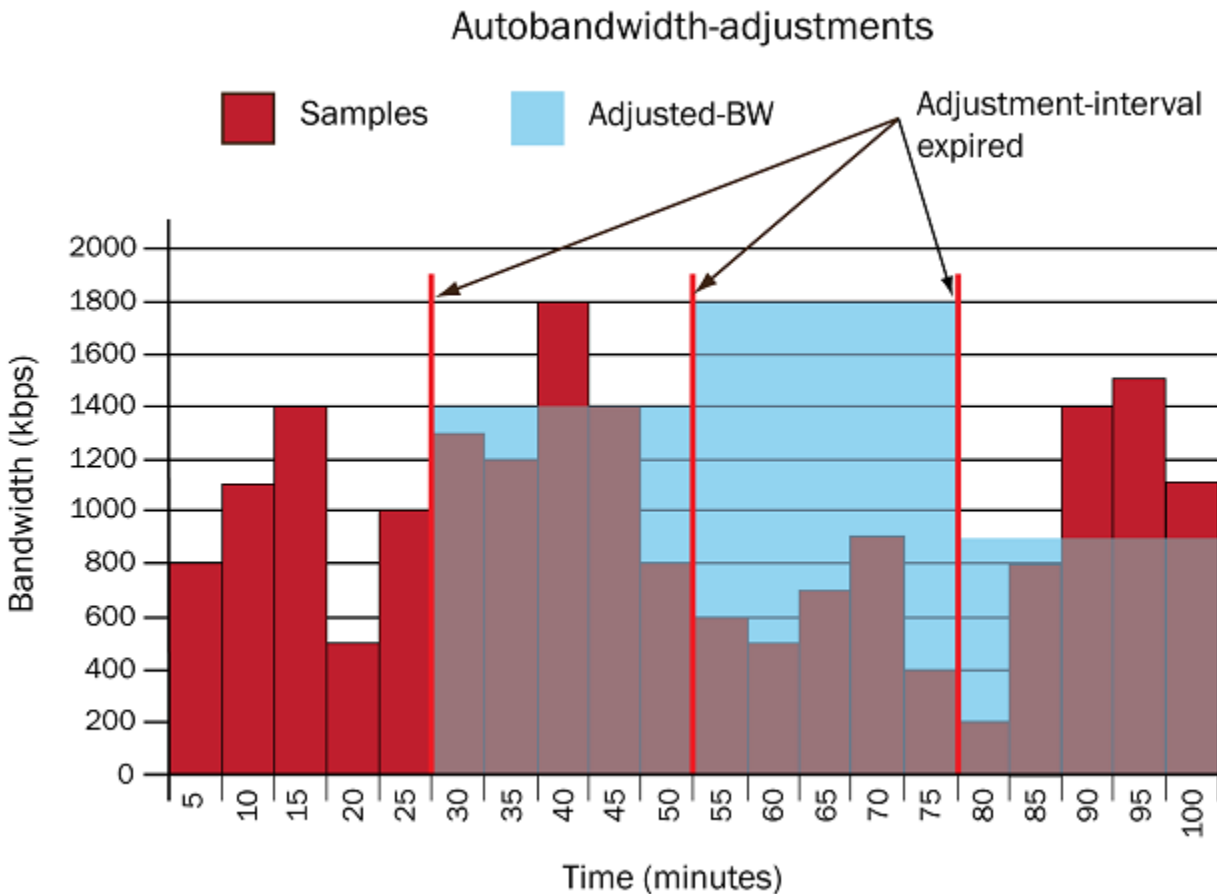
Configuration considerations

- This feature must not be used when a strict bandwidth guarantee is to be provided.
- This feature is only valid for adaptive LSPs.
- When auto-bandwidth is enabled on an LSP and it is not in monitor-only mode and a failover to secondary LSP occurs, the secondary LSP is set up with the configured mean-rate, and the auto-bandwidth process restarts for the secondary LSP when it is adaptive. Therefore, every time the active path changes, auto-bandwidth process starts afresh, with initial bandwidth as the traffic-engineering configured bandwidth on that path.
- A re-optimization event, when triggered, signals the LSP using the current bandwidth and not the highest-sampled bandwidth calculated so far with the auto-bandwidth feature.
- When auto-bandwidth is configured on an LSP, and a change in mean-rate is committed, the auto-bandwidth statistics and counters are cleared, and the entire procedure starts afresh.
- After a switchover, failover, or hitless-reload of MP, the auto-bandwidth counters are cleared.
- Auto-bandwidth is not supported for bypass LSPs.
- Auto-bandwidth continues to function despite FRR failover to the backup path. Irrespective of whether it is a detour or facility backup, or the repair occurred at ingress or transit, the auto-bandwidth process continues. In case of adjustment event occurring while the LSP is repaired, the retries for new instance of protected path is signaled with the new bandwidth. When the new-instance does not come up, the backup remains active.

- The system maximum for LSP accounting must be set for this feature to work. Only those LSPs for which rate counters are allocated on the LP have the auto-bandwidth functionality working.

  NOTE
  A system reload is needed after changing the **system-max lsp-out-acl-cam value** command.

Configuring auto-bandwidth feature at the global level

Auto-bandwidth is disabled by default. It is important that enabling auto-bandwidth globally is necessary to have the auto-bandwidth session running on the LSP. The auto-bandwidth parameters can be pre-configured on an LSP at the LSP level. For additional information and use of the **auto-bandwidth** command, see *NetIron Command Line Interface (CLI) Reference Guide*.

Configuring per-LSP adjustment interval

There are two mechanisms of configuring LSP level parameters. The direct configuration and the template-based configuration. With these mechanisms, there are inheritance and overriding behaviors for various cases.

To specify the bandwidth reallocation interval in seconds for a specific LSP, enter a command similar to the example below:

```
device(config-mpls-lsp-xyz)# auto-bandwidth
device(config-mpls-lsp-xyz-auto-bandwidth)# adjustment-interval 86400
```

The *value* parameter specifies the time interval after which the LSP bandwidth must be adjusted.

For additional information and use of the **auto-bandwidth** command, see *Extreme NetIron Command Reference*.

Configurable table-based absolute adjustment-threshold

The current percentage-based threshold configuration method has certain shortcoming when there are LSPs with a very wide range of traffic-rates. Consider all the links from the ingress to be one Gbps links and the adjustment threshold to be 10%. Now, assume there are two LSPs A and B with bandwidth 10 Mbps and 10 kbps respectively. The actual traffic-rate for LSP A is found to be 11 Mbps with an absolute difference of one Mbps (1000 kbps). The actual traffic rate for LSP B is found to be 11 kbps with an absolute difference of one kbps. Note that the percentage difference for both LSPs is 10% and with the current method an adjustment event is triggered for both LSPs.

It may not be required to initiate a bandwidth adjustment for LSP B as the absolute difference is very small as compared to the link bandwidth. You can have a higher threshold percentage for LSPs with smaller current bandwidth and lower threshold percentage for LSPs with higher current bandwidth.

One way to achieve such a threshold is to define a table which can give the absolute threshold based on the current traffic rate. Following is an example of how a typical threshold table looks like. An additional column is added for illustrating how the percentage threshold varies as the current bandwidth increases.

Range of actual traffic rate Threshold Percentage threshold range

0-1000 kbps 2000 kbps ?? - 200%

1000 kbps to 10 Mbps 3000 kbps 300% - 30%

10 Mbps to 100 Mbps 5000 kbps 50% - 5%

100 Mbps to 1Gbps 7000kbps 7% - 0.7%

The absolute threshold values increase with the actual-traffic rate but the percentage threshold decreases. This way you can make sure that for low traffic-rate LSPs, insignificant bandwidth changes are ignored. This saves costly make-before-break procedures and thus

provide an scalability benefit over the current uniform percentage based method. This is particularly beneficial in cases where a router is having LSPs with a wide range of actual-traffic rates.

Pros of the threshold-table based method:

- Useful when there are LSPs with wide range of actual traffic rate.

- Huge scalability benefit. Avoid insignificant bandwidth adjustments.

  NOTE
  It is important that the range of values and the corresponding thresholds are chosen very carefully.

The percentage-based threshold method and table-based threshold methods will co-exist. There is another option to configure when an LSP is using the percentage-based threshold or the table based threshold. There is a single global table only to be used system-wide by all LSPs. An LSP is allowed to chose from either the threshold-table or the LSP level percentage threshold configured. This flag behaves in the same way as the other LSP level auto-bandwidth parameters. This flag also is allowed to be configured on an auto-bandwidth template. It follows the same inheritance mechanism as other parameters. This threshold is valid for both overflow and underflow determination.

TABLE 4

Global-level auto-bandwidth commands	
Command	Description

Enabling auto-bandwidth globally	The auto-bandwidth parameters can be pre-configured on an LSP at the LSP level. To globally enable automatic bandwidth, use the following commands.
	
	`device(config)# router mpls`
	`device(config-mpls)# policy`
	`device(config-mpls-policy)# auto-bandwidth sample-`
	`interval 30`

Configuring sample intervals	Syntax: **no sample-interval** *value* The sample-interval parameter specifies the time after which the traffic rate is sampled. The *value* parameter specifies the time interval after which the LSP bandwidth must be adjusted. The range is from 60 through 604800 seconds (one week). The default value is 300 seconds.
Configurable parameters:	
Parameter	Description
Adjustment-interval	Displays the configured adjustment-timer value.
Adjustment-threshold	Displays the configured adjustment-threshold value.
Percentage-based	Syntax: **no adjustment-threshold** *value* The *value* parameter specifies the bandwidth is adjusted when the percentage difference in old and new bandwidth is greater than or equal to the specified adjust-threshold percentage, the LSPs. bandwidth is adjusted to the current bandwidth demand. The range is from zero through 100 percent. The default value is zero percent.
Absolute table based	Syntax: **no overflow limit** *value* The *value* parameter specifies the least number of times the sampled bandwidth must consecutively overflow the adjustment threshold to trigger premature adjustment. The range is from zero through 65535 (where zero means it never adjusts for limit overflow). The default value is zero.
Overflow-limit	Displays the configured overflow-limit value.

TABLE 4 (continued)

Underflow-limit	The number of sample which have to be below the threshold to trigger a premature adjustment.
Minimum bandwidth	The configured minimum bandwidth.
Maximum bandwidth	The configured maximum bandwidth.
Mode	Mode will tell if the LSP is in monitor-only or monitor-and-signal mode.
Executable commands:	
Command	**Description**
Clearing auto-bandwidth statistics	To clear statistics for auto-bandwidth-enabled LSPs, enter the following command. `device(config-mpls-lsp-xyz-auto-bandwidth)# clear mpls auto-bandwidth-statistics lsp xyz`
Clearing auto-bandwidth sample history	The auto-bandwidth history is deleted only in the cases when LSP is itself deleted or when user clears or deletes the samples manually. Clearing of auto bandwidth samples by user is recorded in the LSP history.

Manual auto-bandwidth adjustment	To specify the bandwidth reallocation interval in seconds for a specific LSP, enter the following commands.
	
	`device(config-mpls-lsp-xyz)# auto-bandwidth`
	`device(config-mpls-lsp-xyz-auto-bandwidth)# mpls adjust-bandwidth lsp xyz`

Direct configuration versus template-based configuration	There are two mechanisms of configuring LSP level parameters. The direct configuration and template-based configuration. With these mechanisms, there are inheritance and overriding behaviors for various cases.
	
	To specify the bandwidth reallocation interval in seconds for a specific LSP, enter the following commands.
	
	`device(config-mpls-lsp-xyz)# auto-bandwidth`
	`device(config-mpls-lsp-xyz-auto-bandwidth)# adjustment-interval 86400`

Overriding and Inheritance behavior	When the user applies the template to an LSP, the LSP inherits the values from the template. Because direct configuration is local to the LSP, the user can override an inherited value using direct configuration.
Show commands	
Command	**Description**
show mpls lsp detail	The **show mpls lsp detail** command displays detailed information about a specific LSP. Syntax: **show mpls lsp detail**
show mpls lsp extensive	The **show mpls lsp extensive** command shows the adjustment event with the previous rate and the maximum sampled rate. Syntax: **show mpls lsp extensive**
show mpls lsp autobw-samples	The samples obtained in an adjustment-interval can be displayed whenever needed with the show command. Syntax: **show mpls lsp autobw-samples**
show mpls autobw-threshold-table	This command displays the global-threshold table with the range of current-bandwidth and the corresponding absolute adjustment-threshold. Syntax: **show mpls autobw-threshold-table**
show mpls autobw-template	With this method, the user can create a template of auto-bandwidth values and apply it on the primary and secondary paths. Syntax: **show mpls autobw-threshold-template**

Template-based configuration

Template-based configuration provides a different way to configure the parameters. With this method, the user can create a template of Auto-bandwidth values and apply it on the primary and secondary paths.

To create a template, use the following commands:

```
device(config-mpls)# autobw-template abw1
device(config-mpls-autobw-template-abwl)# adjustment-interval 600
device(config-mpls-autobw-template-abwl)# overflow-limit 5
device(config-mpls-autobw-template-abwl)# mode monitor-and-signal
```

To apply the template on a path, go into the **auto-bandwidth** mode and use the following command:

```
device(config-mpls-lsp-tl-autobw)# template abwl
```

Overriding using direct configuration

Template-based configuration is useful when a large number of LSPs need to have the same parameter values. When the user applies the template to an LSP, the LSP inherits the values from the template. Because direct configuration is local to the LSP, the user can override an inherited value using direct configuration.

To override, use the following commands.

```
device(config-mpls-lps-tl-autobw)# template abwl
device(config-mpls-lsp-tl-autobw)# adjustment-interval 300
```

The effective value for the adjustment-interval is 300, irrespective of what value is configured for the template abw1.

Configuring per-LSP range of bandwidth values

Use the following commands to maintain the LSPs bandwidth between minimum and maximum limits by specifying the values.

Setting the maximum bandwidth

The following conditions must be met for setting the maximum bandwidth configuration: Min-bandwidth <= (traffic-eng mean-rate) <= Max-bandwidth. To configure the maximum bandwidth parameter on a per-LSP level, enter the following commands.

```
device(config-mpls-lsp-xyz)# auto-bandwidth
device(config-mpls-lsp-xyz-auto-bandwidth)# max-bandwidth 60000
```

Syntax: [no] **max-bandwidth** *value*

The *value* parameter specifies that the LSP bandwidth can never be greater than this value. When **traffic-eng max-rate** is configured, **max-bandwidth** cannot be configured to be greater than **traffic-eng max-rate**. The range is from 0 through 2147483647 kbps. The default value is 2147483647 kbps.

The [**no**] option sets the corresponding parameter value to the default.

Setting the minimum bandwidth

To configure the minimum bandwidth parameter on a per-LSP level, enter the following commands.

```
device(config-mpls-lsp-xyz)# auto-bandwidth
device(config-mpls-lsp-xyz-auto-bandwidth)# min-bandwidth 600
```

Syntax: [no] **min-bandwidth** *value*

The *value* parameter specifies that the LSP bandwidth can never be lower than this value. The range is from 0 through 2147483647 kbps. The default value is 0 kbps.

The [**no**] option sets the corresponding parameter value to the default.

Underflow-limit

Current provision for premature bandwidth adjustment is available only for the case when the actual traffic rate is found to be consistently greater that the current bandwidth. The opposite case where the actual traffic rate is much less than the reserved bandwidth is not handled. This may result in very long periods of over-provisioned bandwidth. In order to avoid this, a new parameter called the underflow-limit is provided. This parameter can be configured along the same lines as overflow-limit. This parameter is configurable on an auto-bandwidth template as well as an LSP. The same inheritance rules as other parameters applies.

Suppose the underflow-limit is set to 10. When 10 consecutive samples of the LSP traffic rate are found to be lesser than the LSP bandwidth by an amount more than the threshold, a premature adjustment is triggered setting the LSP bandwidth to the maximum of these 10 consecutive samples. The sampled-rate chosen as the new bandwidth for the LSP is the maximum of those 10 samples that triggered the underflow. If adjustment-threshold is configured to use the autobw-threshold-table, the threshold from the auto-bandwidth table is used. Unlike overflow-limit, the number of underflow counts is not reset after adjustment-interval expiry. This means that out of the 10 samples that triggered the adjustment, six may be the beginning samples of current adjustment period while the remaining four may be the last samples of the previous adjustment period.

Configuring overflow limit to enable premature adjustment

The automatic bandwidth adjustment timer is a periodic timer that is triggered at every adjustment interval to determine whether any bandwidth adjustments are required on the LSPs. This interval is configured as a long period of time, usually a number of hours. If, at the end of an adjustment interval, the change in bandwidth is above an certain adjustment threshold, the LSP is re-signaled with the new bandwidth.

During every adjustment interval, the device samples the average bandwidth utilization of an LSP and, when this has exceeded the current maximum average bandwidth utilization, the maximum average bandwidth utilization is updated.

Overflow-limit allows for adjusting the bandwidth without waiting for the adjustment-timer to expire. When the sampled traffic rate is found to be greater than the current bandwidth and exceeding the threshold, consecutively, for a number of times configured as overflow-limit, bandwidth adjustment is triggered using the maximum sampled bandwidth in the current adjustment interval.

To specify a limit on the number of bandwidth overflow samples before triggering an automatic bandwidth allocation adjustment, enter the following commands.

```
device(config-mpls-lsp-xyz)# auto-bandwidth
device(config-mpls-lsp-xyz-auto-bandwidth)# overflow limit 5
```

Syntax: [no] overflow limit *value*

The *value* parameter specifies the least number of times the sampled bandwidth must consecutively overflow the adjustment threshold to trigger premature adjustment. The range is from zero through 65535 (where zero means it never adjusts for limit overflow). The default value is zero (0).

The [**no**] option sets the corresponding parameter value to the default.

Configuring the monitoring mode

The **mode** command can be either **monitor-and-signal** or **monitor-only**. When the mode is set to **monitor-only**, LSPs bandwidth adjustments are disabled, but the maximum average bandwidth is continuously monitored. The auto-bandwidth session only gathers and displays the relevant information. The default mode is **monitor-and-signal**.

```
device(config-mpls-lsp-xyz)# auto-bandwidth
device(config-mpls-lsp-xyz-auto-bandwidth)# mode monitor-only
```

Syntax: **[no] mode monitor-only | monitor-and-signal**

The [**no**] option sets the corresponding parameter value to the default.

Manually triggered bandwidth allocation adjustments

For manually triggered bandwidth allocation adjustments, the minimum value for the adjustment interval is five (5) minutes (300 seconds). It might be necessary to trigger a bandwidth allocation adjustment manually.

A manually triggered bandwidth adjustment operates only on the active LSP. When the user enables periodic automatic bandwidth adjustments, the periodic automatic bandwidth adjustment parameters are not reset after a manual adjustment.

Once the user executes the **mpls adjust-bandwidth** command, the automatic bandwidth adjustment validation process is triggered. The conditions to be met for successful manual adjustment are that at least one sample must have been collected and the maximum of samples collected must cross the threshold. This must be executed in the **monitor-and-signal** mode. Once all the criteria for adjustment are met, the LSPs active path bandwidth is adjusted to the maximum of the rate samples collected so far in the current adjustment-interval.

```
device(config-mpls-lsp-xyz)# auto-bandwidth
device(config-mpls-lsp-xyz-auto-bandwidth)# mpls adjust-bandwidth lsp xyz
```

Syntax: **mpls adjust-bandwidth lsp** *lsp_name*

The **lsp** *lsp_name* parameter adjusts the bandwidth for the named LSP.

Clearing auto-bandwidth counters

To clear statistics for auto-bandwidth-enabled LSPs, enter the following command.

```
device(config-mpls-lsp-xyz-auto-bandwidth)# clear mpls auto-bandwidth-statistics lsp xyz
```

Syntax: **clear mpls auto-bandwidth-statistics lsp** *lsp_name*

The **lsp** *lsp_name* parameter clears the statistics for the named LSP.

Sample-history

With the current implementation of the feature, only the previous sampled rate is display as part of the **show mpls lsp detail** command. The LSP history obtained with the **show mpls lsp extensive** command also only shows the adjustment event with the previous rate and the maximum sampled rate. With this feature, you are given the option to record all the events related to auto-bandwidth of an LSP using the CLI command **sample-recording enable** or **disable**.

The samples obtained in an adjustment-interval can be displayed whenever needed with the show command **show mpls lsp autobw-samples**. The history contains relevant auto-bandwidth events.

You are given the freedom to clear or delete the auto-bandwidth samples at any point of time using the CLI command **clear mpls auto-bandwidth-samples**. The auto-bandwidth history is deleted only in the cases when LSP is itself deleted or when you clear or delete the samples manually. Clearing of auto bandwidth samples by you is recorded in the LSP history.

Displaying auto-bandwidth configurations

The auto-bandwidth fields under the primary path and secondary path sections display the Auto-bandwidth parameter values configured on that path. Under the Active Path attributes sections, the effective parameter values and the auto-bandwidth session information is displayed. The parameters values are suffixed with "(T)", when the parameter value is inherited from the template.

> NOTE
> The running configuration information for **auto-bandwidth** is not displayed when there is no active path or the active path is not adaptive. Auto-bandwidth does not function under these conditions. Instead, a message is displayed indicating the cause.

MPLS fast reroute using one-to-one backup

The Multi-Service IronWare software supports MPLS Fast Reroute to provide the ability for an LSP to route traffic around a failed node by using a detour route as described in *RFC 4090*. By using the one-to-one backup method, each LSR except the egress router is identified as a *Point of Local Repair (PLR)*. Each PLR tries to initiate a *detour LSP* to provide a backup route for the protected path. This detour LSP is used to reroute traffic locally on the detour path in the event of a failure on the protected path.

Finding a detour at a PLR

Figure 13 illustrates how the algorithm works to determine the detour at each PLR.

> NOTE
> Although the example illustrates this method from only the Ingress router point-of-view, the same functionality operates on each PLR in the protected path.

FIGURE 13 Fast reroute using one-to-one backup
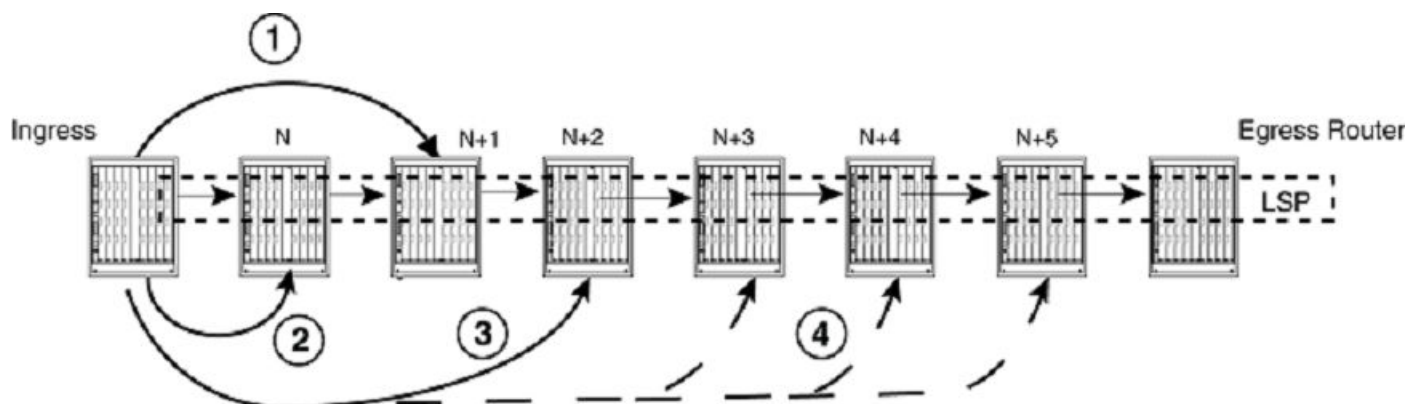
As shown In Figure 13, MPLS Fast Reroute operates according to the steps in the following list in a situation where the path from the ingress router to router N becomes inoperable.

1. The router first tries to find a detour path from the ingress router to the N + 1 node that excludes the failed link that the protected path traverses out of the ingress route and Node N.

2. When unable to find a detour path to node N + 1, in step 1, the router attempts to find a detour path from the ingress router to node N that excludes the link that the protected path traverses out of the ingress router.

3. When it is unable to find a detour path in steps 1 and 2, it attempts to find a detour path to any downstream node (until it reaches the egress LSR) immediately following the node N+1 in strict order. The exclusion criteria includes the downstream links (in the direction of the protected LSP) used in the protected path at each PLR.

Failover sequence

The following steps describe what happens when the ingress LER learns that a downstream break along an LSP has caused the LSP to take a detour.

1. At the PLR, the LSPs traffic has switched over to a detour within 50 milliseconds. Signaling has informed the router at the ingress LER of the tunnel that this event has occurred.

2. When the secondary path configured is a standby and it is in an operationally UP state, the ingress LER waits up to two minutes before switching the traffic to the LSPs secondary path. When the secondary path configured is a non-standby, the ingress LER attempts to bring the secondary path UP. Once the non-standby secondary path comes up, the ingress LER switches the traffic to the secondary path immediately.

3. The ingress LER tears down the LSPs primary path and builds a new primary path.

   After the new primary path is up for the duration of the user-configured LSP revert timer, the LSP switches over to the primary LSP path.

MPLS Fast Reroute using facility backup over a bypass LSP

A bypass LSP is an MPLS LSP that serves as a tunnel to support facility backup of multiple, Fast Reroute LSPs, as specified in *RFC 4090*. Although the underlying mechanism of this feature is facility backup, the execution of facility backup is implemented through a user-defined bypass LSP, so this section focuses on bypass LSP.

The advantage to using bypass LSP is an improvement in the scalability of protection. It provides a nearly hitless backup and, as a result, improves network resiliency. A bypass LSP consists of a predefined tunnel with a list of LSPs for which it is always ready to reroute traffic and is, therefore, a many-to-one backup. (With a detour backup, as described in MPLS fast reroute using one-to-one backup on page 70, the network calculates an end-to-end detour for each disrupted LSP.)

The following definitions are important for understanding and configuring bypass LSPs:

- *Protected LSP*: An LSP whose traffic is carried over the bypass LSP when a link or router fails along the path of the protected LSP. When an MPLS LSP is configured to have Fast Reroute backup, that LSP can also be configured to request either facility backup or one-to-one backup.

- *Facility backup*: The standards-based mechanism for many-to-one backup.

- *Bypass LSP*: A tunnel that carries traffic when any number of its protected LSPs fail.

- *Point of local repair (PLR)*: A router where the protected LSP and the bypass LSP first intersect and the LSPs bypass protection begins. Put another way, the PLR is the ingress of the bypass LSP. The PLR can be the ingress of the protected LSP or a transit node of the protected LSP. (refer to PLR/R2 in Figure 14 and the description in Configuring a bypass LSP on page 73.)

- *Merge point (MP)*: The egress router of the bypass LSP, where it merges the traffic back into the protected LSPs. (R4 in Figure 14 is the MP.) At the MP, the protected LSPs continue to carry traffic towards their own egress routers. Just as the PLR is common to all the LSPs protected by a specific bypass LSP, the MP must also be common to the protected LSPs.

- *Exclude interface* : An MPLS interface that is either a physical interface or a LAG and has the following traits:
    - It is an interface on the path of the protected LSP. (The notion that an excluded interface is protected by a bypass LSP is described in Configuring a bypass LSP on page 73.)
    - It is an interface that cannot be part of the bypass LSP itself.
    - Exclude interfaces can consist of individual interfaces, ranges of interfaces, groups, or a LAG.

FIGURE 14 Facility backup applied to multiple routers over a bypass LSP
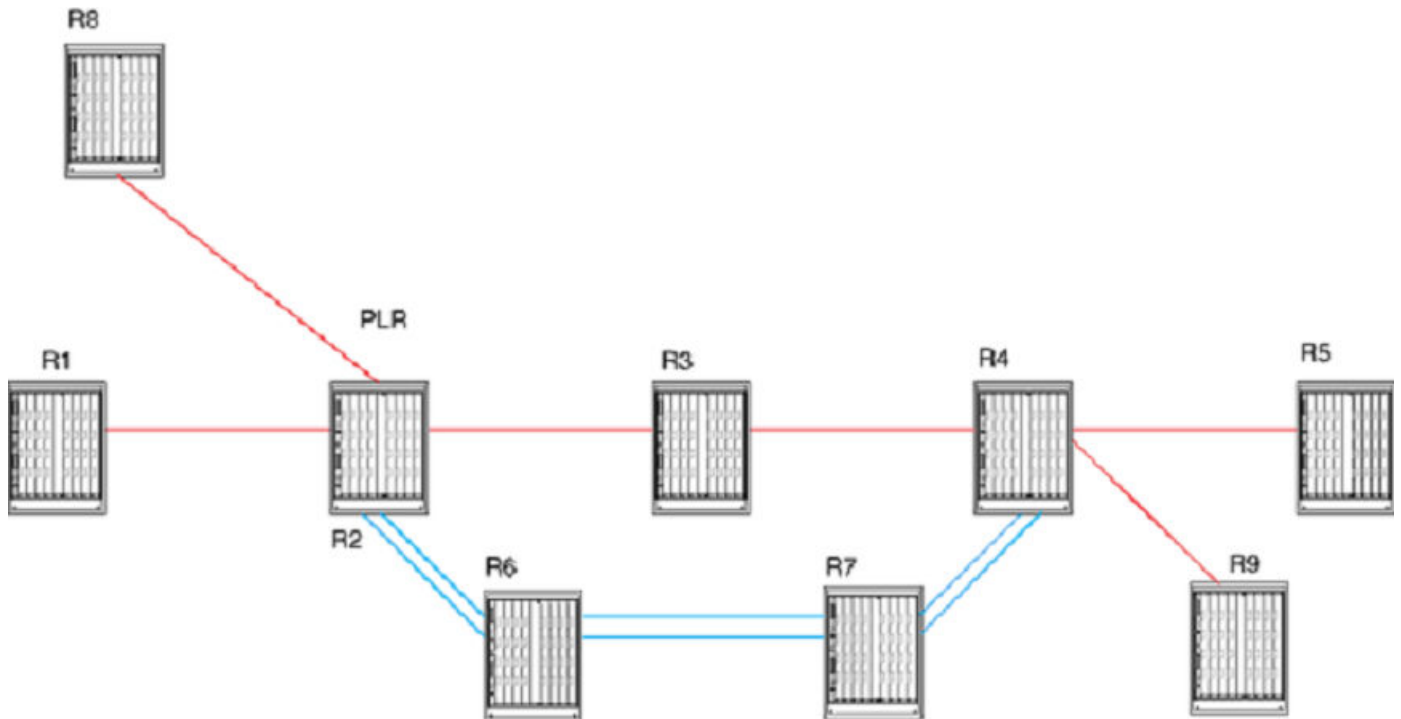
Configuring a protected LSP

To acquire the protection of one or more bypass LSPs along its route, an LSP that is requesting facility backup checks the interfaces that it traverses for the availability of a bypass LSPs that meet its requirements. (A Fast Reroute LSP that needs facility backup must request it. Refer to Protecting MPLS LSPs through a bypass LSP on page 147 for the configuration steps.) The requesting LSP checks all of the bypass LSPs on the outbound interface of each router and selects a candidate bypass LSP that best meets its criteria so that, when the protected LSP fails, its traffic immediately switches to the bypass LSP that is upstream from the point of failure.

When an LSP is enabled for Fast Reroute, the CLI enters the configuration level for Fast Reroute, which has the option for requesting facility backup. Entering the keyword **facility-backup** in the Fast Reroute level configures the LSP to request facility backup as provided by a bypass LSP. Subsequently, for the LSP to acquire the protection of a bypass LSP, that bypass LSP must have the bandwidth, the constraints, the route (for the merge point), and other criteria that the LSP requires. Furthermore, the configuration of the bypass LSP itself must list the interface on the router where the candidate LSP and the bypass LSP first intersect. The description of linking a Fast Reroute LSP to a bypass LSP is in Configuring a bypass LSP on page 73.

Configuring a bypass LSP

The crucial topics to understand for configuring a bypass LSP are the PLR, the merge point, and the excluded interfaces. This section provides a detailed definition of these items and describes how they relate to each other. Refer to MPLS Fast Reroute using facility backup over a bypass LSP on page 71 and Figure 15 for the description of these topics.

The PLR is the ingress of a bypass LSP. When a protected link breaks downstream from the PLR, the bypass LSP carries the traffic of the LSPs it protects around the break. As shown in MPLS Fast Reroute using facility backup over a bypass LSP on page 71, the LSPs from R1 and R8 enter R2 (the PLR). The double line that originates at R2 and then traverses R6 and R7 to terminate at R4 is the bypass LSP.

In MPLS Fast Reroute using facility backup over a bypass LSP on page 71, the egress router for the protected LSPs is R5. Upstream from R5 is the point (R4) where the bypass LSP terminates and merges the traffic back into the protected LSPs. This router is the merge point.

In facility backup, the interfaces that go into this arrangement can belong to either:

- The bypass LSP
- The protected LSP

The specification of a bypass LSP includes manual entry of a list of interfaces at the PLR that cannot make up the bypass LSPs own route. These *excluded* interfaces are the interfaces that the protected LSPs traverse. Therefore, from the standpoint of the bypass LSP, the protected interfaces on the PLR are called *excluded* interfaces. (When the protected interfaces were included in the backup path rather than excluded from the backup path, then the interfaces would be protecting themselves -- a logical contradiction.)

In facility backup, the linkage of the protected LSP to the bypass LSP is established by the following events:

- The request from an MPLS LSP for facility backup: At the ingress node (R1 in MPLS Fast Reroute using facility backup over a bypass LSP on page 71, for example), LSP 1 is configured to request facility backup.
- The intersection of an MPLS LSP and a bypass LSP: When an LSP requesting facility backup traverses an interface on a router (R2 MPLS Fast Reroute using facility backup over a bypass LSP on page 71) with a bypass LSP that has LSP 1's outbound interface in its user-specified list of excluded interfaces, then LSP 1 can become protected at that point, and R2 is a PLR.

The bypass LSP identifies the interfaces to protect in the command that creates the bypass LSP. In Figure 15, LSP 1 and LSP 2 enter R2. The outbound interfaces for these LSPs are e 1/1 and e1/2. To provide protection to LSP 1 and LSP 2, the interfaces e 1/1 and e 1/2 are listed as exclude interfaces in the configuration of the bypass LSP.

In complex topologies, an interface can have multiple bypass LSPs protecting it. For example, the LSPs that traverse an interface might have destinations that make a single merge point impossible, so multiple bypass LSPs would be needed in this case to support different LSPs. Therefore, more than one bypass LSP can have the same interface in its list of exclude interfaces.

> **NOTE**
> The bypass LSP must have the bandwidth capacity to carry the traffic of all of its assigned LSPs. Before a candidate LSP chooses a bypass LSP on a given interface, software determines whether the bypass LSP can reserve sufficient bandwidth for the candidate LSP.

> **NOTE**
> In the current release, BFD for facility backup FRR LSP is not supported. The system returns an error when the user tries either to enable BFD for facility backup for an LSP or to set facility-backup mode for an LSP with BFD enabled. Further, the BFD option is not available in the bypass LSP configuration context.

FIGURE 15 Excluded interfaces on a PLR
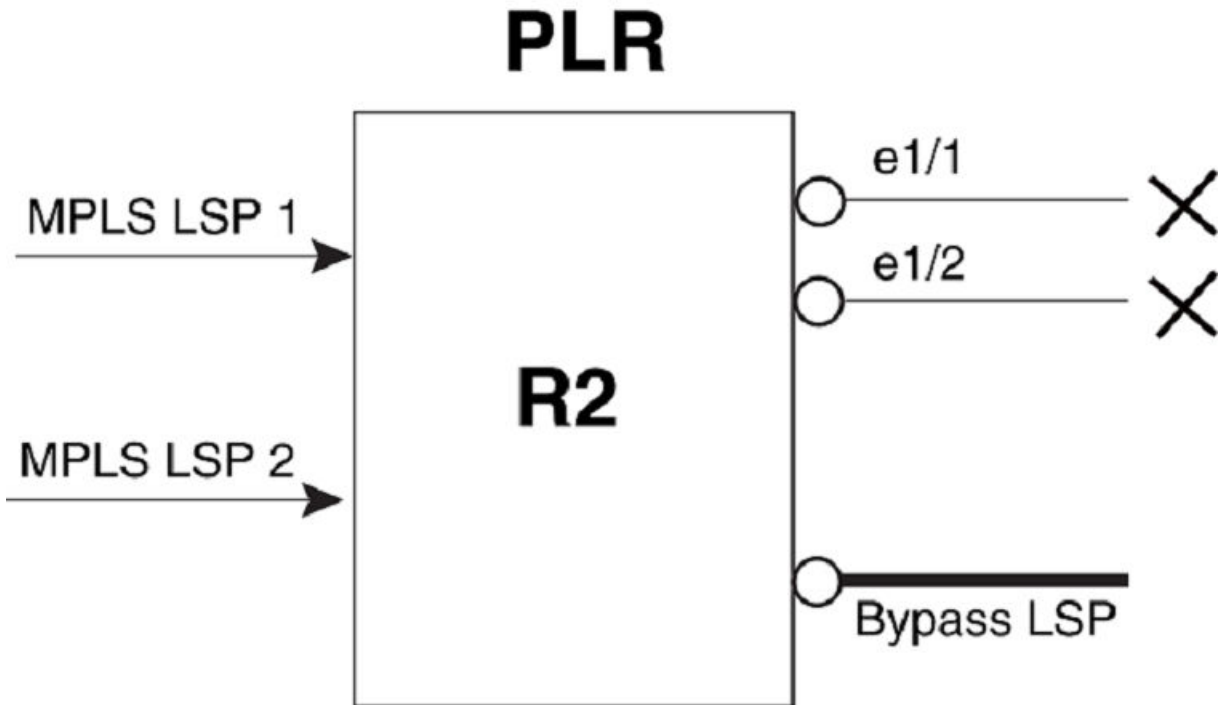
Bypass LSP, like one-to-one backup, fits within the scope of MPLS Traffic Engineering, so the configuration of bypass LSP includes elements of traffic engineering. For example, setting up a bypass LSP relies on RSVP and CSPF. In fact, CSPF is automatically enabled on a bypass LSP and, therefore, does not appear as a configurable option at the bypass LSP configuration level.

CLI differences between a protected LSP and a bypass LSP

In the Fast Reroute context of MPLS LSP configuration, the user can request facility backup. For example, to request facility backup for LSP xmr2-199, use the following command.

```
device(config-mpls-xmr2-199-frr)# facility-backup
```

For the configuration of a bypass LSP, certain parameters are either unsupported or unnecessary. These are:

- CSPF (because it is always enabled)
- BFD (not supported in this release)
- Commit
- Secondary path
- FRR
- Selected path
- IPMTU
- Metric
- Revert-timer
- Select-path
- Shortcuts

In contrast, the parameter that is unique to bypass LSP is the specification of excluded interfaces, which can be embodied as individual interfaces, ranges of interfaces, groups, or LAGs. With bypass LSP 123.

```
device(config-mpls)# bypass-lsp 123
device(config-mpls-bypasslsp-123)# exclude-interface ethernet 1/1 ethernet 1/3
device(config-mpls-bypasslsp-123)# exclude-interface ethernet 1/1 ethernet 1/3 to 1/4
```

Syntax: [no] exclude-interface ethernet | pos | ve *slot/port* [ethernet | pos | ve *slot/port* | to *slot/port*]

Facility backup over an adaptive bypass LSP

Adding the adaptive capability to a bypass LSP enables the following capabilities:

- An enabled bypass LSP can switch to a lower cost route, when one becomes available. This action is known as path re-optimization. Calculation of such a route can occur on demand, or periodically, based on the expiration of a configurable re-optimization timer.
- The user can modify LSP parameters on an enabled bypass LSP. Without the adaptive capability, the user must disable the bypass LSP before the user can modify its parameters.

Figure 16 shows an example of an adaptive bypass LSP for which a lower-cost route has been signaled. In this case, the original bypass LSP was configured to take the route beginning at the Point of Local Repair (PLR) at R2, through R6 and R7, and finally to the merge point at R4. A recalculation found a lower-cost route and re-routed the bypass LSP through R10 and R11, instead of R6 and R7. The Point of Local Repair does not change, nor does the merge point.

FIGURE 16 Adaptive bypass LSP
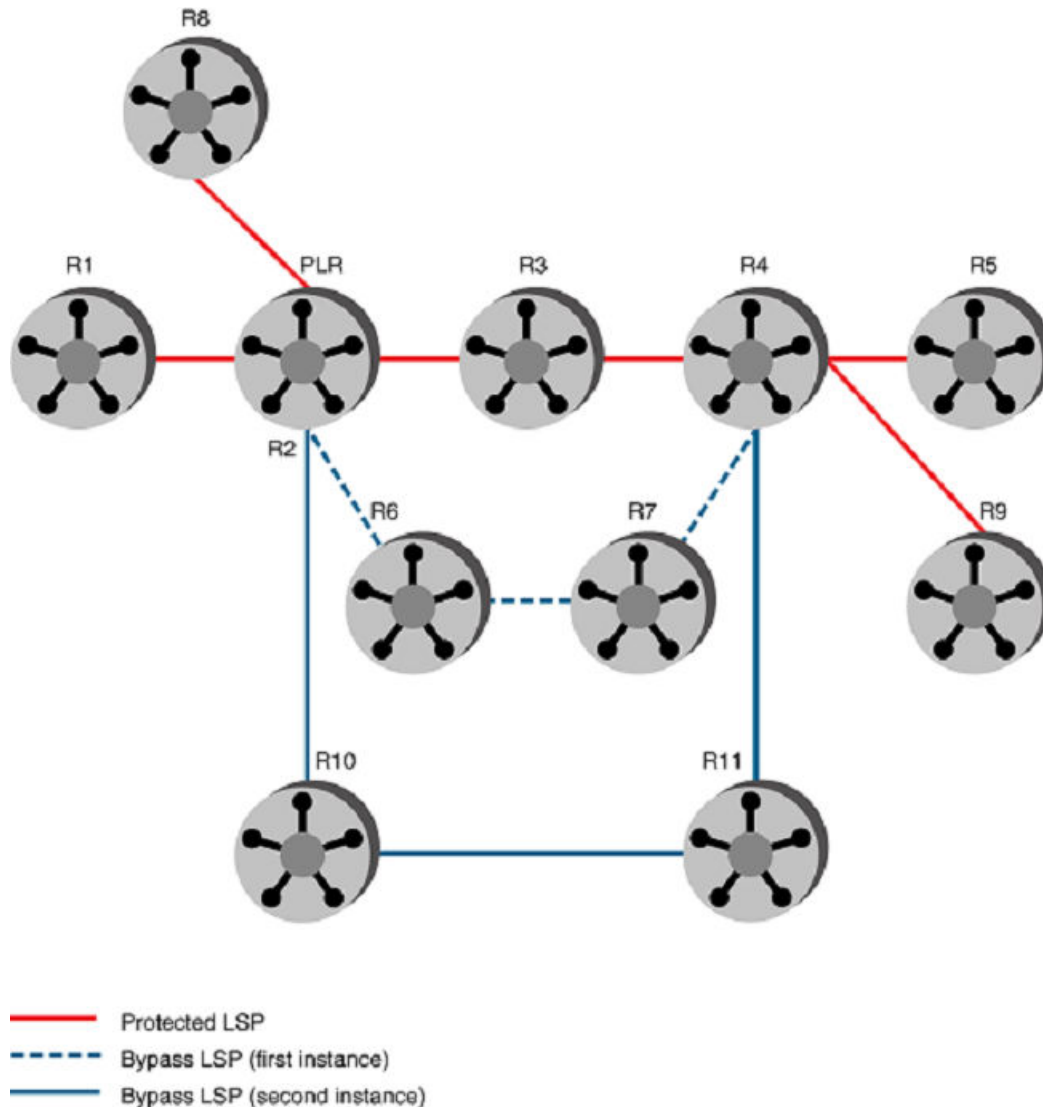
A new instance of the bypass LSP becomes active as soon as it is signaled. After the new instance becomes active, the old instance is released.

To minimize the effect on user traffic, signaling of a new instance for the bypass LSP not does occur when the current instance is carrying traffic. In this regard, adaptive bypass LSPs behave differently from adaptive regular LSPs. When a re-optimization calculation finds a better route while the bypass LSP is carrying traffic, the re-optimization is discarded, and another calculation is made the next time the re-optimization timer expires. An attempt to change an LSP parameter when the bypass LSP is carrying traffic is delayed until the bypass LSP is no longer carrying traffic. The configuration is accepted and it takes effect when the LSP is re-routed or the new instance is signaled.

For bandwidth-protected LSPs, it is possible that a new bypass LSP instance could have a lesser bandwidth than the cumulative bandwidth of the LSPs it is protecting. For example, the user could alter the bandwidth of the bypass LSP with the **traffic-eng mean-rate** command. In this case, backups are released using a priority-based scheme until the bandwidth of the new bypass instance is at least as big as the cumulative bandwidth of the LSPs it protects.

For instructions on how to configure an adaptive bypass LSP, refer to

Adaptive Fast Reroute (FRR) and Global Revertiveness

Adaptive capabilities support to *Fast Reroute (FRR)* and enabling global revertiveness enables the following capabilities:

- Once FRR is triggered, a make-before-break operation is performed to re-establish the primary path. When an established path attempts to reroute onto a new path, the ingress device maintains existing paths and allocated bandwidths, ensuring that the existing path is not prematurely torn down and allowing the current traffic to continue flowing while the new path is set up.
- Configuration of the secondary path to have the LSP re-trigger the primary path is no longer required.
- The LSP waits for the configured revertive hold time after FRR is triggered before trying to re-optimize.

Figure 17shows an example of a primary LSP between A-B-C and backup over a detour path A-D-C. The primary LSP is configured without a strict path. When the interface between A-B goes down, the global revertiveness feature triggers a new LSP on the path A-E-C. The traffic is shifted to the new instance and old instance is torn down.

When the primary LSP is triggered with strict path (A-B-C), after global revertiveness is triggered, a new instance tries the same path given in the strict path. In Figure 17, new instance also tries to come up in the path A-B-C.

FIGURE 17 Sample topology for global revertiveness

Router D
Loopback 10.4.4.4

Router A
Loopback 10.2.2.2

Router B
Loopback 10.1.1.1

Router C
Loopback 10.3.3.3

Router E
Loopback 10.1.1.1

Primary Path   Backup Path   New primary after A-B link failure
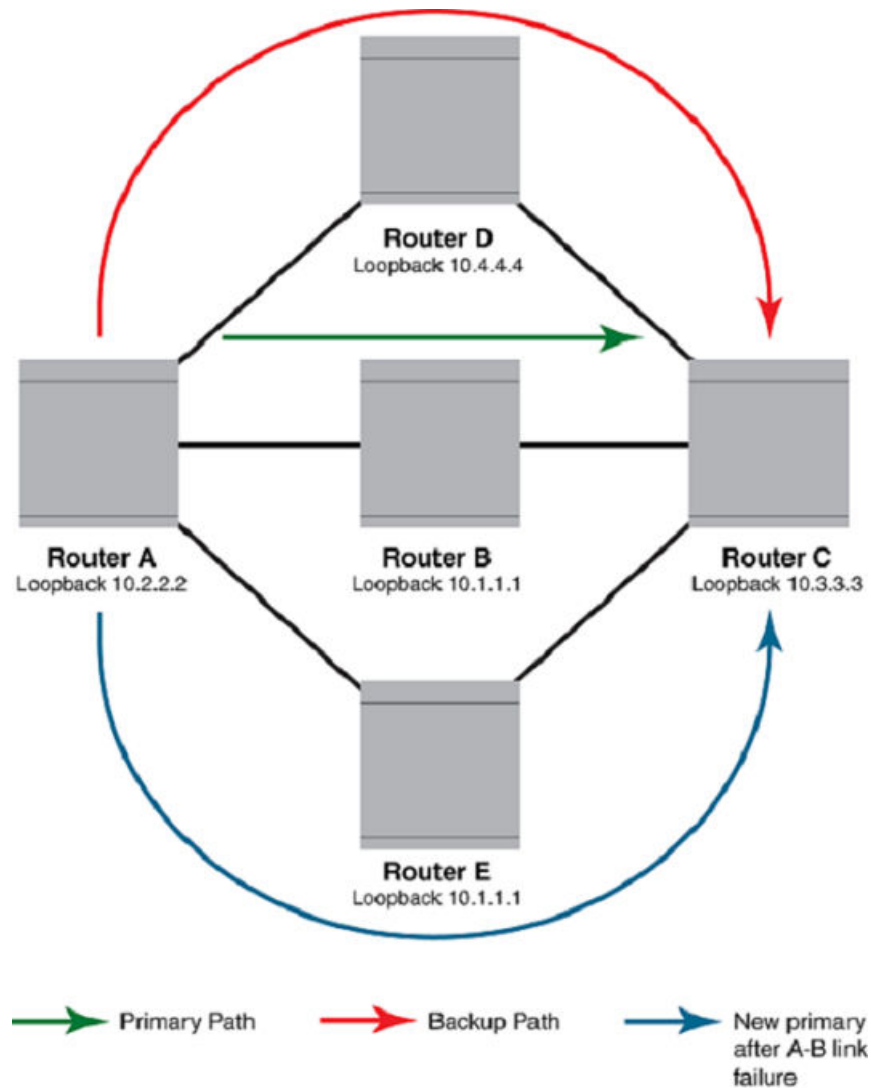
Configuring FRR on an LSP to be adaptive

When an FRR is enabled, the user can change the following parameters without disabling the LSP:

- bandwidth
- exclude-any
- hop-limit
- include-all
- include-any
- priority

For instructions on how to configure an adaptive FRR LSP, refer to Configuring MPLS Fast Reroute using one-to-one backup on page 145.

Global Revertiveness

> **NOTE**
> Local revertiveness is not supported in this release.

When failover happens, traffic continues to flow in backup. When global revertiveness for FRR is configured, a new LSP is created from the ingress after the ingress learns about the failover. The new LSP is protected with a backup LSP, if possible. When the primary LSP fails for the second time, it may still be protected when there is a backup path available.

When secondary path is configured along with global revertive configuration, then when new instance of global revertive is triggered, the secondary path is also triggered. After "n" number of retries configured by user for establishing new instance for global revertiveness, traffic switches to the secondary path. The retry limit is configured in **mpls policy** mode. When the retry limit is not configured, then new instance establishment is tried infinite times.

Configuring global revertiveness

Global revertiveness is enabled by default for LSPs with **FRR** and **adaptive** enabled. The **revertive mode global** command can be executed only on LSPs with **FRR** and **adaptive** enabled. To enable global revertiveness, enter commands such as the following.

> **NOTE**
> When **adaptive** is disabled, then global revertiveness is also disabled.

```
device(config-router-mpls)# lsp t1
device(config-router-mpls-lsp-t1)# adaptive
device(config-router-mpls-lsp-t1)# frr
device(config-router-mpls-lsp-t1-frr)# revertive global enable
```

Syntax: [no] **revertive global enable**

The [**no**] option disables global revertiveness on an LSP.

Setting the revertive hold time

Use the **revertive hold-time** command to specify the time the LSP holds before attempting a new path on the FRR LSP.

```
device(config-router-mpls)# lsp t1
device(config-router-mpls-lsp-t1)# adaptive
device(config-router-mpls-lsp-t1)# frr
device(config-router-mpls-lsp-t1-frr)# revertive global enable
device(config-router-mpls-lsp-t1-frr)# revertive holdtime 20
```

Syntax: [no] **revertive hold-time** *hold-time-value*

The *hold-time-value* parameter specifies the hold time value in seconds. The hold-time is the time between the primary LSP failure and the trigger of new instance of LSP by global revertiveness. Possible range is one through 60 seconds.

The default is five seconds.

The [**no**] option sets it the timer to the default.

Global Revertiveness configurations

Global revertiveness is enabled by default in FRR mode for an adaptive LSP.

Adaptive LSP configuration

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp t1
device(config-router-mpls-lsp-t1)# to 10.3.3.3
device(config-router-mpls-lsp-t1)# from 10.2.2.2
device(config-router-mpls-lsp-t1)# traffic-eng mean-rate 1000
device(config-router-mpls-lsp-t1)# adaptive
device(config-router-mpls-lsp-t1)# frr
device(config-router-mpls-lsp-t1-frr)# facility-backup
device(config-router-mpls-lsp-t1-frr)# exit
device(config-router-mpls-lsp-t1)# enable
device(config-router-mpls)#
```

Changing FRR bandwidth for an adaptive LSP

```
device(config)#
device(config)# router mpls
device(config-router-mpls)# lsp t1
device(config-router-mpls-lsp-t1)# frr
device(config-router-mpls-lsp-t1-frr)# bandwidth 1000
device(config-router-mpls-lsp-t1-frr)# exit
device(config-router-mpls-lsp-t1)# commit
device(config-router-mpls-lsp-t1)#
```

Global Revertiveness configuration

Global revertiveness is enabled by default in FRR mode for an adaptive LSP.

```
device# configure terminal
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# retry-limit 20
device(config-router-mpls-policy)# exit
device(config-router-mpls)# lsp t1
device(config-router-mpls-lsp-t1)# adaptive
device(config-router-mpls-lsp-t1)# frr
device(config-router-mpls-lsp-t1-frr)# revertive mode global
device(config-router-mpls-lsp-t1-frr)# revertive holdtime 20
device(config-router-mpls-lsp-t1-frr)# exit
device(config-router-mpls-lsp-t1)# commit
device(config-router-mpls-lsp-t1)#
```

Displaying global revertiveness information

Use the **show mpls lsp name** *lsp_name* command to display revertive mode information. The **show mpls lsp name***lsp_name* command displays detailed information about a specific LSP name.

```
device# show mpls lsp name tunnel1
LSP tunnel1, to 10.3.3.3
  From: 10.2.2.2, admin: UP, status: UP, tunnel interface(primary path): tnl0
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0  Adaptive
  Maximum retries: NONE, no. of retries: 0
  Pri. path: p1, up: yes (backup), active: yes
   Setup priority: 7, hold priority: 0
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
...
Active Path attributes:
    Tunnel interface: tnl0, outbound interface: e1/3
...
Backup LSP: UP, out-label: 3, outbound interface: e1/3
  Path cspf-group computation-mode: disabled
```

```
Global revertiveness enabled with hold time 20 secs
Revertive timer expires in 17 seconds
FRR Forwarding State: Pri(down), Backup(active)
```

The output from the **show mpls lsp name** *lsp_name* command is enhanced to display the global revertiveness configuration. In this example, the global revertiveness is enabled with a hold time 20 seconds. The revertive timer is set to expire in 17 seconds. The secondary switchover timer is set to expire in 31 seconds which triggers the secondary path establishment.

MPLS CSPF fate-sharing group

A MPLS CSPF fate-sharing group or a *Shared Risk Link Group (SRLG)* is a method used to group *Traffic Engineering (TE)* links and nodes in a network that share the same risk of failure. The user can influence the path computation for a CSPF-enabled LSP by configuring a CSPF fate-sharing group so that both the protected path and the backup path avoid sharing the same TE links traversed. The path computation for a CSPF-enabled LSP uses the information from the TE database to compute the best path for an LSP satisfying all constraints (bandwidth reservations, network topology information, available resources), yet has the shortest distance to its destination. The CSPF computation for an LSP only uses the information from the TE database at the time of computation. Any future updates to the TE database do not cause the CSPF-enabled LSP to recompute. Each CSPF fate-sharing group has an associated penalty (or cost) assigned to it. The penalty associated with a CSPF fate-sharing group is used to direct the path computation for a CSPF-enabled LSP away from TE links that share the same risk used by the set of TE links that the protected path is using. The greater the penalty associated with a group, the less likely the secondary shares TE links used by the protected path.

A CSPF fate-sharing group is identified by a group name, and uses the following four ways to identify elements in the TE database:

- Interface address – The interface address identifies all TE links by either the local address, or the remote address matching the configured interface address.
- Point-to-point link – A point-to-point link identifies TE links by the local address and the remote address on an interface. A point-to-point link specifies the *from* address and the *to* address. The order in which the address is configured is not significant.
- Node – The node address is used to identify the device. All TE links from this device are included.
- Subnet – The IP address with subnet mask identifies all TE links by either the local interface or the remote address belonging to the configured subnet.

A CSPF fate-sharing group can be used for setting up a secondary LSP when the associated primary LSP is in an UP state.

Refer to, Configuring an MPLS CSPF fate-sharing group on page 82 for more information on configuring the path computation for a CSPF-enabled LSP using CSPF fate-sharing group information.

Configuration considerations when using CSPF fate-sharing group information

Consider the following when using CSPF fate-sharing group information:

> NOTE
> This release only supports a single mode of CSPF computation for a CSPF group by adding penalties to each TE link's native IGP cost when it shares fate-sharing groups used by the protected path.

- CSPF computation using a CSPF group is only applicable when computing a secondary LSP path. It is not applicable to the primary or protected LSP path.
- CSPF calculates the least cost paths first and then applies the hop limit on the paths.
- CSPF computation using a CSPF group is used only for computing the secondary LSP path when the primary LSP is in an UP state. In this case, CSPF collects group information from all TE links used by the primary LSP. For each TE link, CSPF

computes the total adjusted distance. The total adjusted distance for each TE link is equal to the native IGP cost of the TE link plus the sum of all penalties of the CSPF groups that the TE link is associated with, and used by the primary LSP. For example, Q1, Q2, and Q3 is a collection of CSPF groups used by the primary LSP. TE link 1 is a member of CSPF groups Q1 and Q2. Q1 has a penalty of 10, and Q2 has a penalty 30. The total penalty of CSPF groups Q1 and Q2 is equal to 40. The total adjusted distance for TE link 1 is equal to the native IGP cost plus 40. The penalty is only applied once to each shared CSPF group that the TE link is associated with. The secondary LSP path is then computed from ingress to egress using the adjusted distance of each TE link.

Configuring an MPLS CSPF fate-sharing group

To configure a CSPF fate-sharing group, perform the following steps.

1. Enable the device for configuration.

    ```
    device# configure
    ```

2. Enable the MPLS router.

    ```
    device(config)# router mpls
    ```

3. Enable the policy mode.

    ```
    device(config-router-mpls)# policy
    ```

4. Specify the CSPF group computation mode for a fate-sharing group, and enable the **penalty** option by entering the following command in MPLS policy mode.

    ```
    device(config-router-mpls-policy)# cspf-group computation-mode add-penalty
    ```

5. Configure a CSPF fate-sharing group by assigning a name to the group. Enter the following command in router MPLS mode.

    ```
    device(config-router-mpls)# cspf-group group3
    ```

6. Set the penalty value for the CSPF fate-sharing group. Enter the following command.

    ```
    device(config-router-mpls-cspf-group-group3)# penalty 100
    ```

7. Configure the local address of the CSPF fate-sharing group. Enter the following command.

    ```
    device(config-router-mpls-cspf-group-group3)# from 10.1.1.1
    ```

8. Configure the local address and remote address on a point-to-point link of the CSPF fate-sharing group, enter the following command.

    ```
    device(config-router-mpls-cspf-group-group3)# link 10.1.1.1  10.1.1.2
    ```

9. Configure the local Subnet IP address with the subnet mask length. Enter the following command.

    ```
    device(config-router-mpls-cspf-group-group3)# subnet 10.1.2.0/24
    ```

10. To penalize all links from the node IP address, enter the following command.

    ```
    device(config-router-mpls-cspf-group-group3)# node 10.1.1.1
    ```

The following example configures a CSPF fate-sharing group.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# cspf-group computation-mode add-penalty
device(config-router-mpls)# cspf-group group3
device(config-router-mpls-cspf-group-group3)# penalty 100
device(config-router-mpls-cspf-group-group3)# from 10.1.1.1
device(config-router-mpls-cspf-group-group3)# link 10.1.1.1  10.1.1.2
device(config-router-mpls-cspf-group-group3)# subnet 10.1.2.0/24
device(config-router-mpls-cspf-group-group3)# node 10.1.1.1
```

Deleting all CSPF groups

This feature enables the user to delete all the CSPF fate-share groups using a single command. Users are required to confirm execution with a warning message. This command is only available at the router-mpls level and takes no arguments.

Deleting CSPF groups

In this example, group3 has already been set up as a fate-sharing CSPF group. To delete this CSPF fate-sharing group, enter the following command in router MPLS mode.

```
device(config-router-mpls)# no cspf-group group3
```

Syntax: **[no] cspf-group** *group-name*

The *group-name* variable specifies the name of the fate-sharing group and can be up to 128 characters. The objects that can be specified for a fate-sharing group are interface, point-to-point link, node, and subnet. The maximum number of CSPF fate-sharing groups that can be configured on a device is 1000. To delete each configuration group individually, enter the above command with the relevant value for the *group-name* argument.

This allows you to delete all configured groups at once.

Sample configuration

These are the commands for use with the feature.

```
device(config)# router mpls
device(config-router-mpls)# no cspf-group
This will delete all the CSPF groups
Do you want to continue? (enter 'y' or 'n'): y
device(config-router-mpls)#
```

All the CSPF groups are deleted at once at this point.

> **NOTE**
> If there are no cspf-groups to delete, the system generates an error message.

```
device(config-router-mpls)# no cspf-group
This will delete all the CSPF groups
Do you want to continue? (enter 'y' or 'n'): y
No CSPF-groups to delete
device(config-router-mpls)#
```

Displaying CSPF fate-sharing group configuration

To display CSPF fate-sharing group configuration for all groups configured on a device, use the **show run router mpls** command. To display CSPF fate-sharing group information for a specific CSPF group, use the **show run router mpls cspf-group** command.

```
device# show run router mpls cspf-group gold
cspf-group test8
penalty 65535
    node 10.7.7.3
    node 10.7.7.8
```

Syntax: **show run router mpls cspf-group** *name*

Fate-sharing group membership for any given TE link or node consists of its own membership to the group, and the TE node to which it belongs. The output from the **show mpls te database detail** command is enhanced to display the fate-sharing groups to which the TE links or nodes belong. In the following example output, node 10.20.20.20 displays fate-sharing group information for group1/100 and group2/10.

```
device# show mpls te database detail
This Router is 10.100.100.100
Global Link Gen 21
Area 0
  NodeID: 10.20.20.20, Type: Router
info from applied local policies:
  cspf-group member information (name/penalty):
    group1/100
    Type: P2P, To: 10.1.1.1, Local: 10.1.1.2, Remote: 10.1.1.1, Gen 16
      Admin Group: 0x00000000
      Metric: 1
      Link BW: 10000000 kbits/sec
      Reservable BW: 10000000 kbits/sec
      Unreserved BW:
        [0]   10000000 kbits/sec  [1]   10000000 kbits/sec
        [2]   10000000 kbits/sec  [3]   10000000 kbits/sec
        [4]   10000000 kbits/sec  [5]   10000000 kbits/sec
        [6]   10000000 kbits/sec  [7]   10000000 kbits/sec
  info from applied local policies:
    cspf-group member information (name/penalty):
      group2/10
    Type: P2P, To: 10.1.2.1, Local: 10.1.2.2, Remote: 10.1.2.1, Gen 13
      Admin Group: 0x00000000
      Metric: 1
      Link BW: 10000000 kbits/sec
      Reservable BW: 10000000 kbits/sec
      Unreserved BW:
        [0]   10000000 kbits/sec  [1]   10000000 kbits/sec
        [2]   10000000 kbits/sec  [3]   10000000 kbits/sec
        [4]   10000000 kbits/sec  [5]   10000000 kbits/sec
        [6]   10000000 kbits/sec  [7]   10000000 kbits/sec
```

The **show mpls lsp name** *lsp_name* command displays detailed information about a specific LSP name. The output from the **show mpls lsp name** *lsp_name* command is enhanced to display whether the fate-sharing group information is applied to the path computation for a specified LSP. When fate-sharing group information is applied, "yes" is displayed in the field, Fate-sharing group applied. When fate-sharing group information is not applied, "no" is displayed in the field. Fate-sharing group information can also be applied to the path computation for a secondary LSP. In the following example, fate-sharing group information is applied to LSP SJC_to_JFK.

```
device# show mpls lsp name SJC_to_JFK
LSP SJC_to_JFK, to 10.100.100.100
  From: 10.20.20.20, admin: UP, status: UP, tunnel interface(primary path): tnl3
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0  Adaptive
  Maximum retries: NONE, no. of retries: 0
  Pri. path: NONE, up: yes, active: yes
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
```

```
       Tie breaking: random, hop limit: 0
      LDP tunneling enabled: no
      Sec. path: path2, active: no
       Hot-standby: yes, status: up
       Setup priority: 7, hold priority: 0
       Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
       Constraint-based routing enabled: yes
       Fate-sharing group applied: yes
       hop limit: 0
      Active Path attributes:
       Tunnel interface: tnl3, outbound interface: e1/1
       Tunnel index: 2, Tunnel instance: 1 outbound label: 3
       Path calculated using constraint-based routing: yes
       Path calculated using interface constraint: no
       Recorded routes:
        Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
        10.1.1.1
```

Syntax: show mpls lsp name *lsp_name*

The **name** *lsp_name* variable specifies the name of the LSP for which the user wants to display information.

Path selection metric for CSPF computation

The IGP floods two metrics for every link when the MPLS traffic engineering (TE) is configured in a network. The two metrics are the IS-IS link metric and the TE link metric. To optimize the use and performance of the network, it is always better to identify specific tunnels to carry data traffic and voice traffic. This implementation allows you to specify tunnel path selection to the requirements of each type of traffic. For example, certain tunnels are to carry voice traffic (which requires low delay) and other tunnels are to carry data (where delay is acceptable).

The path calculation metric implementation allows you to specify the path calculation for a given tunnel based on either of the following requirements:

- IGP link metric for path calculation for data traffic
- TE link metric for path calculation for voice traffic

The decision of whether to use **te-metric** or **igp-metric** for CSPF computation by the LSPs is determined by CLI configurations at two levels:

- Global level: This configuration covers all RSVP LSPs (primary, secondary LSPs).
- Individual LSP level: This configuration covers all RSVP LSPs).

    NOTE
    The CLI configuration at the LSP level always overrides the configuration at the global level. That is, the decision to use **te-metric** or **igp-metric** for CSPF path calculation if configured at the LSP level, always overrides the configuration at the global level.

Path selection for CSPF computation

The selection of path for the LSP depends on whether you want to use IGP or TE metric for CSPF computation. Consider the network topology in the illustration. The ingress router is DUT 1 and the egress router is DUT 5. There are two cases depending upon whether IGP metric or TE metric is used for CSPF computation.

FIGURE 18 Path selection for CSPF computation
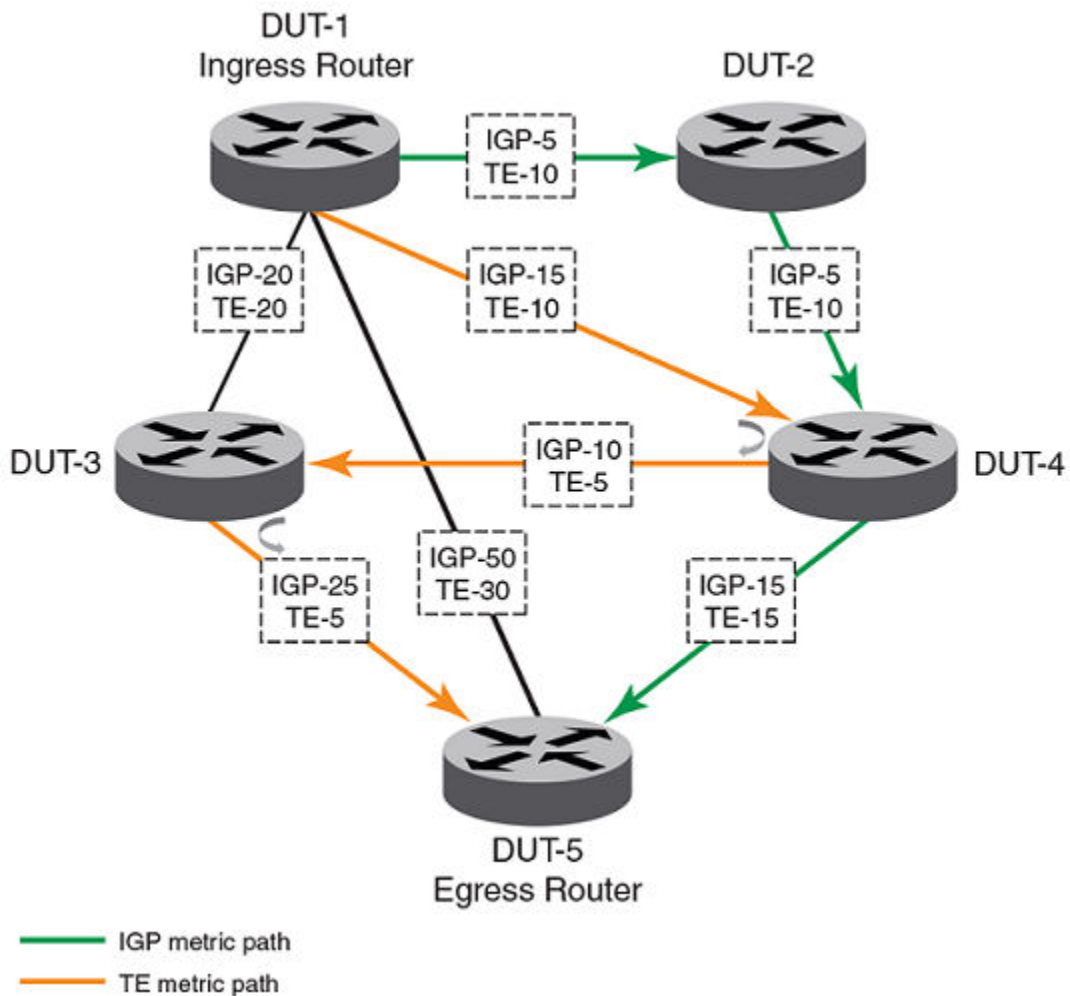
When IGP metric is selected

The LSP selects the following path:

- DUT1 --> DUT2 --> DUT4 --> DUT5

When TE metric is selected

The LSP selects the following path:

- DUT1 --> DUT4 --> DUT3 --> DUT5

Configuring TE-metric for MPLS interface

To configure TE-metric for a MPLS interface, you must perform the following steps.

1. Enable traffic engineering under router mpls policy to ISIS.

2. Configure the MPLS interface.

3. Set the te-metric value at the MPLS interface or leave it as a default value to use the igp-metric value of the te-links for CSPF computation (optional).

   NOTE
   By default, all LSPs use global
   configuration.

Configuring the CSPF computation mode

To configure the CSPF computation mode on a device, you must perform the following steps.

1. Set the cspf-computation mode under router mpls policy to use te-metric or igp-metric at the global level.

2. Enable or disable cspf-computation mode to use te-metric or igp-metric locally at the LSP level for primary and secondary LSPs.(optional)

   NOTE
   By default, all LSPs use the global
   configuration.

Configuring TE-metric for an interface

You can configure the TE metric value at a specified MPLS interface level. Note that traffic engineering needs to be enabled at the router policy level.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ethernet 1/1
device(config-router-mpls-if-eth-1/1)# te-metric 5
device(config-router-mpls-if-eth-1/1)#no te-metric 3
Error:TE-metric is configured to a value of 5
device(config-router-mpls-if-eth-1/1)#no te-metric 5
```

In the example, the TE-metric is set back to a default value of IGP-metric. Run the **show mpls interface ethernet** 1/1 command to view the configured value.

   NOTE
   If te-metric uses the default value or if the no form of the command is used, te-metric will be equal to igp-metric value in the MPLS TE database.

Configuring the CSPF computation mode value at global level

You can configure the cspf-computation mode at the global level under the router mpls policy.

```
device# configure
device(config)# router-mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)#cspf-computation-mode ?
device(config-router-mpls-policy)#cspf-computation-mode use-igp-metric
device(config-router-mpls-policy)#no cspf-computation-mode use-te-metric
Error:CSPF computation is configured to use igp-metric
device(config-router-mpls-policy)#no cspf-computation-mode use-igp-metric
```

In the example, the CSPF computation mode is set back to a default value of te-metric.

Run the **show mpls policy** command to view the configured value.

```
device# sh mpls pol
Current MPLS policy settings:
  CSPF interface constraint: disabled
  CSPF-Group computation-mode: disabled
  CSPF computation-mode:
    Use te-metric:    (default), Ignore overload bit: disabled
  TTL propagation for MPLS label: disabled, IPVPN: disabled, IP over MPLS: enabled
  Inter-AS route filtering: enabled, Intra-AS iBGP route filtering: disabled
  Ingress tunnel accounting: disabled
  Transit session accounting: disabled
  Polling interval for MPLS LSP traffic statistics: 300 seconds
  Advertise TE parameters via: none
  Handle IGP neighbor down event - ISIS: No OSPF: No
  LSP rapid retry: enabled, maximum number of retries: no limit
  LSP periodic retry time: 30 seconds
  FRR backup/detour retry time: 30 seconds
  Soft preemption cleanup-timer: 30 seconds
  MPLS TE Periodic Flooding Timer : 180 seconds
  MPLS TE flooding thresholds
    Global  UP   thresholds : None
    Global  DOWN thresholds : None
    Default UP   thresholds : 15 30 45 60 75 80 85 90 95 96 97 98 99 100
    Default DOWN thresholds : 99 98 97 96 95 90 85 80 75 60 45 30 15
```

Configuring the CSPF computation mode value for primary LSPs

You can configure the cspf-computation-mode value at the primary LSP level.

By default, the LSP uses the global configuration at the router mpls policy. If explicitly configured, the configuration at the LSP level always overrides the configuration at the global level.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp test
device(config-router-mpls-lsp-test)#cspf-computation-mode ?
device(config-router-mpls-lsp-test)# cspf-computation-mode use-igp-metric
device(config-router-mpls-policy)#no cspf-computation-mode use-te-metric
Error:CSPF computation is configured to use igp-metric
device(config-router-mpls-policy)#no cspf-computation-mode use-igp-metric
```

In the example, the CSPF computation mode is set back to a default value of te-metric. Run the **show mpls lsp detail**command to view the configured value.

> **NOTE**
> The configuration is not an adaptive parameter and another instance is not created when the configuration is changed on the fly for adaptive LSPs but on re-optimization it takes up the new configuration to perform cspf computation.

Configuring the CSPF computation mode value for secondary LSPs

You can configure the cspf-computation-mode value at the secondary LSP level.

By default, the LSP uses the global configuration at the router mpls policy. If explicitly configured, the configuration at the LSP level always overrides the configuration at the global level.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp test
device(config-router-mpls-lsp-test)#secondary-path 12
device(config-routr-mpls-lsp-secpath-12)#cspf-computation-mode ?
device(config-router-mpls-lsp-test-secpath-12)# cspf-computation-mode use-te-metric
```

In the example, the CSPF computation mode is set back to a default value of te-metric. Run the **show mpls lsp detail**command to view the configured value.

Configuring the CSPF computation mode value for static bypass LSPs

You can configure the cspf-computation-mode value at the static bypass LSP level.

By default, the LSP uses the global configuration at the router mpls policy. If explicitly configured, the configuration at the LSP level always overrides the configuration at the global level.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# bypass-lsp b1
device(config-router-mpls-bypasslsp-b1)# cspf-computation-mode use-te-metric
```

Run the **show mpls lsp detail**command to view the configured value.

Configuring the CSPF computation mode value for dynamic bypass LSPs

You can configure the cspf-computation-mode for dynamic bypass LSPs at the mpls-interface level.

By default, the LSP uses the global configuration at the router mpls policy. If explicitly configured, the configuration at the LSP level always overrides the configuration at the global level.

```
device(config)#router mpls
device(config-mpls)#mpls-interface eth 1/15
device(config-mpls-if-e1000-1/15)#dynamic-bypass
device(config-mpls-if-e1000-1/15-dynamic-bypass)#cspf-computation-mode ?
device(config-mpls-if-e1000-1/15-dynamic-bypass)#cspf-computation-mode use-te-metric
```

Run the **show mpls dynamic-bypass interface detail** command to view the configured value.

MPLS traffic engineering flooding reduction

Traffic engineering advertisements are triggered when a threshold value is reached or crossed. For all other bandwidth changes, a periodic flooding timer or *Connection Admission Check (CAC)* failure triggers the TE advertisements. When no thresholds are crossed, changes are flooded periodically unless periodic flooding was disabled. Configurations can be executed as a global configuration or interface specific configuration.

Interface specific configurations supersedes global configuration and default values. Global configuration supersedes default values. When there is no interface specific configuration and global configuration, then the default values are used.

Global configuration

Reserved bandwidth threshold configuration can be executed globally and is applied to all MPLS interfaces. Global configurations are done at the policy mode under router mpls.

To set RSVP TE flooding thresholds at the global configuration level, use the **rsvp-flooding-threshold** commands as shown in the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# rsvp-flooding-threshold up 10 20 30 40 50 55 60 65 70
85 90 92 93 94 95 96 97 98 99 100
```

```
device(config-router-mpls-policy)# rsvp-flooding-threshold down 99 98 97 96 95 94 93 92
91 90 85 80 75 70 65 60 55 50 40 30 20 10
```

The **rsvp-flooding-threshold** command can be executed multiple times in the policy mode, the threshold values are added to the existing set of global threshold values. The previously configured values are not overwritten.

The default values for the **rsvp-flooding-threshold** command are list below:

The default for DOWN is 99, 98, 97, 96, 95, 90, 85, 80, 75, 60, 45, 30, 15

The default for UP is 15, 30, 45, 60, 75, 80, 85, 90, 95, 96, 97, 98, 99, 100

In the example below, the UP threshold contains 10, 50, 55, 95, 96, 97, 98, 99 and 100. The DOWN threshold contains 50, 40, 30, 20 and 10.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# rsvp-flooding-threshold up 10 50 55 95 96
device(config-router-mpls-policy)# rsvp-flooding-threshold up 97 98 99 100
device(config-router-mpls-policy)# rsvp-flooding-threshold down 50 40 30
device(config-router-mpls-policy)# rsvp-flooding-threshold down 20 10
```

Interface specific configuration

The **rsvp-flooding-threshold** command can be executed multiple times for the same interface. The threshold values are added to the existing set of values for the interface. Previously configured values are not overwritten. The interface specific configuration overrides the global configuration. Using the no form of this command removes the sub-set of the configured threshold values.

Use the **rsvp-flooding-threshold** command at the MPLS interface level as shown below to set the reserved bandwidth threshold.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ethernet 1/1
device(config-router-mpls-if-eth-1/1)# rsvp-flooding-threshold up 10 20 30 40 50 55 60
65 70 85 90 92 93 94 95 96 97 98 99 100
device(config-routr-mpls-if-eth-1/1)# rsvp-flooding-threshold down 99 98 97 96 95 94
93 92 91 90 85 80 75 70 65 60 55 50 40 30 20 10
```

In the example below, the UP thresholds contain 10, 50, 55, 95, 96, 97, 98 and 100. The DOWN thresholds contain 50, 40, 30, 20 and 10.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ethernet 1/1
device(config-router-mpls-if-eth-1/1)# rsvp-flooding-threshold up 10 50 55 95
device(config-router-mpls-if-eth-1/1)# rsvp-flooding-threshold up 96 97 98 99 100
device(config-router-mpls-if-eth-1/1)# rsvp-flooding-threshold down 50 40
device(config-router-mpls-if-eth-1/1)# rsvp-flooding-threshold down 30 20 10
device(config-router-mpls-if-eth-1/1)#
```

Syntax: [no] rsvp-flooding-threshold [**up** | **down**] [**percentage**] *

Use the [**no**] form of this command to remove the RSVP TE flooding threshold configuration.

The **down** option sets the thresholds for decreased resource availability. Valid values are from 0 to 99. The default values for down is 100, 99, 98, 97, 96, 95, 90, 85, 80, 75, 60, 45, 30, 15.

The **up** option sets the thresholds for increased resource availability. Valid values are from 1 to 100. The default values for up is 15, 30, 45, 60, 75, 80, 85, 90, 95, 97, 98, 99, 100.

The **percentage** option sets the bandwidth threshold level.

The "*" represents multiple percent values can be given. A minimum one percentage value is required.

Percentage values can be given in any order and are internally sorted before storing.

Configuring the periodic flooding timer

All MPLS interfaces are checked every three minutes by default. TE advertisements are triggered when there is a difference in the available bandwidth and advertised available bandwidth.

Use the **rsvp-periodic-flooding-time** command to set the interval for periodic flooding. The interval is set in seconds. To set the interval as 240 which triggers periodic flooding every four minutes, enter commands such as the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# rsvp-periodic-flooding-time 240
device(config-router-mpls-policy)# no rsvp-periodic-flooding-time
```

Syntax: **[no] rsvp-periodic-flooding-time** *interval*

Use the *interval* parameter to set the length of interval used for periodic flooding (in seconds). Valid range is zero, 30-3600. For value zero, periodic flooding is turned off.

The [**no**] form of this command can be used to set the periodic flooding timer to default value.

MPLS over virtual Ethernet interfaces

Extreme devices support MPLS over *virtual ethernet (VE)* interfaces. MPLS over VE interfaces enables MPLS to be configured over tagged links. With this feature, MPLS can run over a single tag on the port. Other tags on the port can be used for other applications, such as Layer 2 VLANs, VPLS endpoints, and VLL endpoints.

An MPLS enabled VE interface supports the following services.

- IP over MPLS
- Transit LSR
- PBR over MPLS
- LSP Accounting
- MPLS VLL
- MPLS VPLS
- Multicast Snooping over VPLS
- 802.1ag
- MPLS OAM

    NOTE
    Multi-port static ARP configuration in not supported for MPLS uplinks.

Configuration considerations before enabling MPLS on a VE interface

Before enabling MPLS on a VE interface, consider the configuration notes in this section.

- The user must create a VE *vid* virtual interface ID. The virtual interface ID is a decimal number that represents an already configured VE interface.

- At least one IP address must be configured over a VE interface.
- The user can enable MPLS on two or more tags on the same port.
- In the output of the **show vlan** command, MPLS packets that are received on an MPLS enabled VE interface are displayed in the Bytes received field.

MPLS enabled interface

When enabling MPLS on a VE interface, consider the following.

- The user cannot delete a VE interface while MPLS is enabled on it. The user must first remove MPLS from the interface configuration. The following error message is displayed.

```
device(config)# no interface ve 100
%%Error: MPLS is enabled on the interface. First disable MPLS on this interface.
```

- The user cannot delete a VLAN associated with a VE when MPLS is enabled on that VE. The user must first disable MPLS from the VE interface. The following error message is displayed.

```
device(config)# no vlan 20
Error - vlan can't be deleted as MPLS is enabled on associated VE interface
```

- When MPLS is enabled on an interface, the last IP address of a VE cannot be removed. The command is rejected. The following error message is displayed.

```
device(config-vif-54)# no ip address 10.40.40.5/24
IP/Port: Error(31) Can not remove IP address as MPLS is configured on the port
```

VPLS CPU protection

> **NOTE**
> VPLS CPU protection is not applicable to CES 2000 Series or CER 2000 Series devices.

When enabling MPLS on a VE interface with VPLS CPU protection turned on, consider the following.

- VPLS CPU protection must be disabled globally, or disabled on all instances of VPLS that has VE member port as the VPLS endpoint. When VPLS CPU protection is not disabled, then MPLS cannot be enabled on a VE interface. The following error message is displayed.

```
device(config-mpls)# mpls-interface ve 1
Error - Port 4/3 belongs to a VPLS instance that has CPU-protection ON
```

- The user cannot configure a VPLS endpoint on a member of a MPLS VE enabled interface when VPLS CPU protection is configured globally, or for a specified instance. The configuration is rejected, and the following error message is displayed.

```
device(config-mpls-vpls-test-vlan-11)# tagged ethernet 4/3
Error - VPLS instance test has CPU protection ON and port 4/3 belongs to a MPLS VE
```

- When a VPLS endpoint belonging to a VPLS instance (with CPU protection turned on) is on a port that does not belong to the VE, then the user cannot add a port in an untagged or tagged mode to a VLAN which has a MPLS VE on it.

```
device(config-vlan-100)# tagged ethernet 4/7
Error - Port 4/7 belongs to a VPLS instance that has CPU protection ON
```

- On an MPLS VE enabled interface, when a VPLS endpoint is a member of the VE interface, but VPLS CPU protection is not configured for the VPLS instance, then configuring VPLS CPU protection globally does not enable CPU protection for that

instance. When VPLS CPU protection is enabled locally on that instance, the configuration is also rejected. The following error message is displayed.

```
device(config-mpls)# vpls-cpu-protection
Error - Cannot configure CPU protection for VPLS 111 as end-points share the same physical port as MPLS VE
interfaces.
CPU protection feature is not turned on for VPLS 111
```

Reverse path forwarding

When enabling MPLS on a VE interface with reverse path forwarding, consider the following.

- The user cannot configure MPLS on a VE interface that has at least one member port enabled with RPF strict mode. The command is rejected, and the following error message is displayed.

```
device(config-if-eth-4/3)# mpls-interface ve 1
Error - Cannot configure MPLS on VE with RPF strict mode port e 4/3
```

- The user cannot configure RPF strict mode on a port that is a member of a MPLS VE interface. The command is rejected, and the following error message is displayed.

```
device(config-if-eth-4/3)# rpf-mode strict
Error: RPF: Cannot configure RPF strict mode on an MPLS VE enabled interface
```

- The user cannot add a port to a MPLS VE enabled VLAN when the RPF strict mode is already enabled on the port. The command is rejected, and the following error message is displayed.

```
device(config-vlan-100)# tagged ethernet 4/3
Error - Cannot add RPF strict mode port 4/3 to MPLS VE enabled VLAN 100
```

Port mirroring

When enabling MPLS on a VE interface with port mirroring configured, consider the following.

- The user cannot configure MPLS on a VE interface that has at least one member port enabled with port mirroring. The command is rejected, and the following error message is displayed.

```
device(config-mpls)# mpls-interface ve 54
Error - Can not configure MPLS tunnel on ve 54 with mirror port e4/3
```

- The user cannot configure a MPLS VE member port as a mirror port. The command is rejected, and the following error message is displayed.

```
device(config)# mirror-port e 4/3
Error: Cannot mirror a port that has MPLS VE configured
```

- The user cannot add a mirror port to a MPLS VE enabled VLAN. The command is rejected, and the following error message is displayed.

```
device(config-vlan-100)# tagged ethernet 4/3
Error - Cannot add mirror port 4/3 to MPLS VE enabled VLAN 100
```

Protocol-based VLANs

When enabling MPLS on a VE interface associated with a protocol-based VLAN, consider the following.

NOTE

MPLS is supported only on VE interfaces that are configured on port-based
VLANs.

- The user cannot configure MPLS on a VE interface associated with a protocol based VLAN. The command is rejected, and the following error message is displayed.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ve 1
Error: Cannot configure MPLS on VE built on protocol-based VLAN
```

VRF

When enabling MPLS on a VE interface for a VRF instance, consider the following.

NOTE

When configuring MPLS on a VE interface on a VLAN port, the user can also configure a VRF instance on other VLANs of the same port.

- The user cannot configure a VRF instance on a MPLS VE enabled interface. The following error message is displayed.

```
device(config-vif-20)# vrf forwarding test
Error - cannot configure VRF on an MPLS VE enabled interface
```

Class of Service (CoS)

By default, the internal priority of a packet received on a tagged MPLS uplink is mapped from PCP and EXP bits. When determining the internal priority, the first step is to merge the PCP and EXP bits. In this step, when configuring the **qos exp force** command on an interface, the internal priority is mapped only from EXP bits and PCP bits are ignored. The **qos exp force** command does not override the port priority command. In the second step, the port priority is merged with the internal priority so the **qos exp force** command has no effect on this step.

By default, the internal priority of a packet sent out on a tagged MPLS uplink is mapped into EXP and PCP bits. When configuring the **qos pcp-encode policy off** command on an outgoing interface, the PCP bits is zero.

Configuring MPLS

Enabling MPLS

MPLS is disabled by default. To enable MPLS on a device, the user must perform the steps listed below.

1. Enable MPLS on the device
2. Enable MPLS on individual interfaces
3. Set global MPLS policy parameters (optional)
4. Set traffic engineering parameters for MPLS-enabled interfaces (optional)
5. Set RSVP parameters (optional)

Enabling MPLS on the device

To enable MPLS on the device, enter the following commands.

```
device# configure
device(config)# router mpls
```

To disable MPLS on the device, use the [**no**] form of the command.

Enabling MPLS on individual interfaces

After the user enables MPLS globally on the device, the user can enable it on one or more interfaces. For example, to enable MPLS on interface ethernet *3/1* .

```
device(config-router-mpls)# mpls-interface ethernet 3/1
```

Configuration Considerations for enabling MPLS on a LAG interface

When MPLS is globally enabled on the device, a port that is configured in a LAG can be enabled as an MPLS interface port to create an MPLS LAG. The user can do this through either of the following approaches:

- Include a primary LAG port that has already been MPLS-enabled in a new LAG
- MPLS-enable a primary LAG port of a previously configured LAG

The user must consider the following points when configuring MPLS on a LAG:

- MPLS configuration on dynamic lag interfaces are supported
- Switch and LACP LAGs are not supported
- MPLS is enabled on the primary port of the LAG and this enables MPLS on the entire LAG. Secondary ports of the LAG cannot be individually configured for MPLS.

Setting global MPLS policy parameters

The user can optionally set the following global MPLS policy parameters (they apply to all MPLS-enabled interfaces on the device):

- Retry time
- Retry limit
- Administrative group names
- Whether the device sends out IS-IS LSPs with TE extensions for its MPLS-enabled interfaces
- Configuring IP-over-MPLS TTL Propagation Control
- LSP Accounting

Setting the retry time

When a signaled LSP is enabled, the ingress LER attempts to connect to the egress LER over the primary path specified in the LSPs configuration. When the connection is not successful, by default the ingress LER waits 30 seconds before attempting the connection again. The user can configure the amount of time the ingress LER waits between connection attempts.

For example, to specify a retry time of 45 seconds.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# retry-time 45
```

Syntax: **[no] retry-time** *seconds*

Setting the retry limit

When the ingress LER fails to connect to the egress LER in a signaled LSP, the ingress LER tries indefinitely to make the connection unless the user sets a limit for these connection attempts. After this limit is exceeded, the ingress LER stops trying to connect to the egress LER over the primary path.

When a secondary path is configured for the LSP, it is immediately activated after the primary path fails. After the secondary path is activated, the ingress LER continues to try to connect to the egress LER over the primary path either up to the configured retry limit or indefinitely when no retry limit is set. When a connection over the primary path can be established, the secondary path is deactivated, and traffic for the LSP is again sent over the primary path.

To set the number of connection attempts to *20* .

```
device # configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# retry-limit 20
```

Once the connection is established, the retry counter is reset to zero. In the example above, when an LSP needs to be established again, the ingress LER makes 20 attempts to establish a connection to the egress LER.

Establishing administrative group names

Administrative groups, also known as resource classes or link colors, allows the user to assign MPLS-enabled interfaces to various classes. When a device calculates the path for an LSP, it can take into account the administrative group to which a interface belongs; the user can specify which administrative groups the device can include or exclude when making its calculation.

Up to 32 administrative groups can be configured on the device. The user can see an administrative group either by its name or its number. Before the user can see an administrative group by its name, the user must specify a name for the group at the MPLS policy level and associate the name with that administrative group's number.

For example, the following commands establish three administrative group names.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# admin-group gold 30
device(config-router-mpls-policy)# admin-group silver 20
device(config-router-mpls-policy)# admin-group bronze 10
```

Syntax: **[no] admin-group** *admin_name name /number*

The *number* has a range of 0 - 31.

After the user associates an administrative group name with a number, the user can see it by name when assigning interfaces to the group or including or excluding the group from LSP calculations. Refer to Adding interfaces to administrative groups on page 101 and Including or excluding administrative groups from LSP calculations on page 127.

Enabling OSPF-TE LSAs for MPLS interfaces

Information related to traffic engineering is carried in *OSPF traffic engineering (OSPF-TE)* LSAs. OSPF-TE LSAs have special extensions that contain information about an interface's traffic engineering metric, bandwidth reservations, and administrative group memberships.

When an MPLS-enabled device receives an OSPF-TE LSA, it stores the traffic engineering information in its *Traffic Engineering database (TED)* . The device uses information in the TED when performing calculations to determine a path for an LSP.

The user can configure the device to send out OSPF-TE LSAs for all of its MPLS-enabled interfaces. To do this, enter the following commands.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# traffic-engineering ospf
```

Syntax: [no] traffic-engineering ospf [area *area-id*]

The user can use the **area** option to limit the CSPF calculations to the OSPF Area specified by the *area-id* variable. The *area-id* variable can accept area-id in both Decimal and IP address formats.

By default, the device does not send out OSPF-TE LSAs for its MPLS-enabled interfaces. Because information in the TED is used to make path selections using CSPF and information in the TED comes from OSPF-TE LSAs or IS-IS TE LSP, the user must enable the device to send out OSPF-TE LSAs or IS-IS LSPs with TE extensions when the user wants CSPF to perform constraint-based path selection.

The [**no**] option removes an existing OSPF TE database. When the user employs the [**no**] option with the **area** option, the OSPF TE database is removed for only the specified OSPF area.

Refer to Setting traffic engineering parameters for MPLS interfaces on page 99, for information on the traffic engineering information carried in OSPF-TE LSAs.

Enabling IS-IS LSPs with TE extensions for MPLS interfaces

Information related to traffic engineering is carried in IS-IS traffic engineering LSPs. IS-IS TE LSPs have special extensions that contain information about an interface's administrative group memberships, IPv4 interface address, IPv4 neighbor address, maximum link bandwidth, reservable link bandwidth, unreserved bandwidth, and default traffic engineering metrics.

When an MPLS-enabled device receives an IS-IS TE LSP, it stores the traffic engineering information in its *Traffic Engineering database (TED)*. The device uses information in the TED when performing calculations to determine a path for an LSP.

The user can configure the device to send out IS-IS TE LSPs for all of its MPLS-enabled interfaces. To do this, enter the following commands.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# traffic-engineering isis level-1
```

Syntax: [no] traffic-engineering isis level-1 | level-2

The **level-1** option enables LSPs with TE extensions for the IS-IS level-1 domain.

The **level-2** option enables LSPs with TE extensions for the IS-IS level-2 domain.

By default, the device does not send out IS-IS LSPs with TE extensions for its MPLS-enabled interfaces. Since information in the TED is used to make path selections using CSPF, and information in the TED comes from IS-IS LSPs with TE extensions, the user must enable the device to send out IS-IS LSPs with TE extensions when the user wants CSPF to perform constraint-based path selection.

Refer to Setting traffic engineering parameters for MPLS interfaces on page 99, for information on the traffic engineering information carried in IS-IS LSPs with TE extensions.

Configuring CSPF interface constraint

As described in detail in Calculating a path based on an interface address on page 42, under the default condition, hops configured as interface addresses in an LSP path are resolved to the router ID. Consequently, an LSP can be configured that does not traverse a specified interface. The **cspf-interface-constraint** command was introduced that forces the CSPF calculation to include any specified interface when creating an LSP. The operation and constraints of using this command are described in the section mentioned.

The user can configure an Extreme device to always include a specified interface when forming an LSP by configuring the **cspf-interface-constraint** command as shown in the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# cspf-interface-constraint
```

Syntax: [no] cspf-interface-constraint

The default condition is for the CSPF interface Constraint feature to be disabled. When the feature has been enabled, the user can use the [**no**] option to disable it.

The CSPF interface Constraint feature may be dynamically turned on or off. Turning the feature off or on has no effect on LSPs that have already been established (primary and secondary). For LSPs that are currently retried, changing the constraint setting changes the behavior on the next retry such as when an LSP whose path is configured to use that interface fails to come up due to an interface down condition.

Also note that the CSPF interface Constraint feature has significance for the ingress node only, where the CSPF calculation takes place for an LSP or a detour segment.

Configuration changes to route filtering for MPLS and iBGP routes

LDP currently caches all routes known to the *Routing Table Manager (RTM)* except eBGP routes. The following commands allows the user to control the number of routes cached in LDP, and the type of route that MPLS accepts from the RTM.

- filter-inter-as-routes
- filter-intra-as-ibgp routes

Inter-AS routes originate from other BGP autonomous systems. Intra-AS routes originate from within a BGP AS.

Configuration considerations

> **NOTE**
> Extreme recommends that the user makes route filtering configuration decisions when booting the router for the first time.

A system reload is not required when the user changes the filtering configuration. However, when the user enables *inter-as* filtering, RSVP sessions using iBGP routes, and LDP FECs (corresponding to iBGP routes) will flap.

Configuring route filtering for MPLS and iBGP routes

When the user enables inter-as-route filtering, the RTM does not send any inter-AS routes to MPLS. To enable inter-as filtering, enter the following commands in the MPLS policy configuration mode.

```
device# configure
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# filter-inter-as-routes
```

Syntax: [no] filter-inter-as-routes

> NOTE
> **Intra-as** filtering is disabled by
> default.

When the user enables **intra-as** filtering, the RTM does not send any iBGP routes to MPLS. **Intra-as** filtering can only be enabled when **inter-as** filtering is enabled. To enable the **intra-as** filtering, enter the following commands.

```
device# configure
device(config)# router mpls
device(config-mpls)# policy
device(config-router-mpls-policy)# filter-inter-as-routes
device(config-router-mpls-policy-route-filter)# filter-intra-as-ibgp-routes
```

Syntax: [no] filter-intra-as-ibgp-routes

To disable **intra-as** filtering, enter the [**no**] version of this command.

To disable all filtering, including intra-as filtering, enter the following command under the MPLS policy configuration mode.

```
device# configure
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# no filter-inter-as-routes
```

Setting traffic engineering parameters for MPLS interfaces

When using constraints to determine a path for an LSP, the device takes into account information included in IS-IS LSPs with TE extensions. This information can be used to set up a path for a new LSP or to preempt an existing LSP so that an LSP with a higher priority can be established.

IS-IS LSPs with TE extensions include *Type/Length/Value triplets (TLVs)* containing the following information:

- IP address of the local interface
- IP address of the remote interface (must exist with point-to-point links)
- Traffic engineering metric for the link
- Maximum bandwidth on the interface
- Maximum reservable bandwidth on the interface
- Unreserved bandwidth on the interface
- Administrative groups to which the interface belongs

. When configured to do so with the **traffic-engineering isis** command, the device sends out IS-IS LSPs containing this TE information for each of its MPLS-enabled interfaces. Optionally, the user can specify the maximum amount of bandwidth that can be reserved on an interface. In addition, the user can assign interfaces to administrative groups.

Reserving bandwidth on an interface

IS-IS LSPs with TE extensions contain three TLVs related to bandwidth reservation:

- The maximum bandwidth TLV indicates the maximum outbound bandwidth that can be used on the interface. Maximum bandwidth is the operating speed of the port. When calculated for a LAG, the Maximum Bandwidth is the operating speed of the primary port multiplied by the number of active ports in the LAG. This reflects the actual physical bandwidth of the interface. This TLV is not configurable by the user.
- The maximum reservable bandwidth TLV indicates the maximum bandwidth that can be reserved on the interface. By default, the maximum reservable bandwidth is the same as the maximum bandwidth for the interface. The user can optionally change

the reservable bandwidth to an amount greater or less than the maximum available bandwidth of the interface. When a maximum reservable bandwidth is configured on the primary port within a LAG, the value configured applies to the entire LAG regardless of any change to the number of active ports within the LAG. By default, the maximum reservable bandwidth for the LAG is the same as its maximum bandwidth.

- The unreserved bandwidth TLV indicates the amount of bandwidth not yet reserved on the interface. This TLV consists of eight octets, indicating the amount of unreserved bandwidth (in kilobits per second) at each of eight priority levels. The octets correspond to the bandwidth that can be reserved with a hold priority of 0 through 7, arranged in increasing order, with priority 0 occurring at the start of the TLV, and priority 7 at the end of the TLV. The value in each of the octets is less than or equal to the maximum reservable bandwidth. The unreserved bandwidth TLV itself is not user-configurable, although it is affected by modifications to the reservable bandwidth on an interface, as well as changes to LSPs.

Optionally, the user can change the amount of reservable bandwidth on an MPLS-enabled interface (that is, modify the value in the maximum reservable bandwidth TLV in IS-IS TE LSPs sent out for the interface). The maximum reservable bandwidth on an MPLS-enabled interface can be configured in either of two ways: as an absolute value, or as a percentage of the total interface bandwidth.

Configuration considerations

The **reservable-bandwidth** command is configurable on an MPLS-enabled interface at any time. The configuration of the command takes effect immediately upon preemption of the LSP. When LSP preemption occurs, when the reservable bandwidth required for a specific LSP is not supported on the interface, then the LSP immediately goes down. When this occurs, an IGP advertisement of this configuration change is triggered and flooded throughout all ports on the network because the maximum reservable bandwidth configured on the interface is different from the value that was previously configured.

> **NOTE**
> When the maximum reservable bandwidth is configured as a percentage value for LAGs and VE interfaces, and ports go down, or new ports are added to the interface, the reservable bandwidth is recalculated as a percentage of the newly available bandwidth for that interface.

To configure the maximum reservable bandwidth as an absolute value for MPLS LSPs on the interface, enter the following commands as displayed in the following example.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ethernet 1/1
device(config-router-mnpls-if-eth-1/1)# ldp-params
device(config-roouter-mpls-if-eth-1/1-ldp-params)# reservable-bandwidth 10000
```

To configure the maximum reservable bandwidth as a percentage of the total interface bandwidth for MPLS LSPs on the interface, enter the following commands as displayed in the following example.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ethernet 1/1
device(config-router-mpls-if-eth-1/1)# ldp-params
device(config-router-mpls-if-eth-1/1-ldp-params)# reservable-bandwidth percentage 80
```

When maximum reservable bandwidth is changed from an absolute value to a percentage value, and vice versa, the following advisory message is displayed on the console to indicate this configuration change.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ethernet 1/1
device(config-router-mpls-if-eth-1/1)# ldp-params
device(config-router-mpls-if-eth-1/1-ldp-params)# reservable-bandwidth percentage 40
Maximum reservable bandwidth is changed from 30 kbps to 40%
```

NOTE

When the maximum reservable bandwidth is configured as either an absolute value, or a percentage value, the value is recalculated and updated to the latest value.

To set the maximum reservable bandwidth back to the default value (the total physical bandwidth of the interface) when the absolute value or percentage value is used, enter the **no** form of the command as displayed in the following example.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# mpls-interface ethernet 1/1
device(config-router-mpls-if-eth-1/1)# ldp-params
device(config-router-mpls-if-eth-1/1-ldp-params)# no reservable-bandwidth percentage 80
```

By default, the reservable bandwidth is the same as the maximum available bandwidth on the interface. When the amount of reservable bandwidth is greater than the maximum available bandwidth, then the link can be oversubscribed. When the reservable bandwidth is less than the maximum available bandwidth, then LSPs cannot reserve all physical bandwidth on the interface. When the **reservable-bandwidth** command is applied to the primary port within a LAG, the bandwidth configured for that port applies to the entire LAG, regardless of any change to the number of active ports within the LAG.

Changing the amount of reservable bandwidth on an interface causes the amount of unreserved bandwidth to be recalculated. In addition, it may cause an IS-IS TE LSP to be issued, as well as possibly pre-empt existing LSPs when bandwidth reservations can no longer accommodate them.

The output from the **show running-config router mpls** command and the **show router mpls interface** [**ethernet** *slot/port*] command displays the maximum reservable bandwidth configuration. Depending on the interface configuration, the show commands displays the maximum reservable bandwidth as an absolute value, or as a percentage value. When the **no** form of the **reservable-bandwidth** command is used, the default value of the interface bandwidth is also displayed in both show command outputs.

Adding interfaces to administrative groups

The user can place individual interfaces into administrative groups. Administrative groups, also known as resource classes or link colors, allows the user to assign MPLS-enabled interfaces to various classes. For example, the user can define a group called "gold" and assign high-bandwidth interfaces to it. When a device calculates the path for an LSP, it can take into account the administrative group to which a interface belongs. The user can configure up to 32 administrative groups. By default, an interface does not belong to any administrative groups.

Administrative groups are in the range 0 - 31. The user can see an administrative group either by name or number. To see an administrative group by name, first create a name for the group and associate the name with an administrative group number. Refer to Establishing administrative group names on page 96 for details.

To define the administrative group *gold* , enter the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# admin-group gold
```

To assign MPLS-enabled interface e 3/1 to an administrative group called "gold", enter the following.

```
device# configure
device(config)# router-mpls
device(config-router-mpls)# mpls-interface ethernet 3/1
device(config-router-mpls-eth-3/1)# admin-group gold
```

Syntax: [no] admin-group [*number* | *name*]

The *number* can be from 0 - 31. The administrative group name *name* must have been previously configured.

An MPLS-enabled interface can belong to any number of administrative groups. For example, to assign an interface to group "gold" and group 31, enter commands such as the following.

```
device# configure
device(config)# router-mpls
device(config-router-mpls)# mpls-interface ethernet 3/1
device(config-router-mpls-eth-3/1)# admin-group gold 31
```

After the user adds interfaces to administrative groups, the user can specify which groups can be included or excluded from LSP calculations. Refer to Including or excluding administrative groups from LSP calculations on page 127.

IP-over-MPLS TTL propagation control

In the MPLS label header, the TTL field indicates the *Time To Live (TTL)* value for an MPLS packet. For IP-over-MPLS applications, at the ingress LER an IP packet's TTL value is decremented by one and the IP checksum is recalculated. The IP packet's TTL value is then copied to its MPLS TTL field. At each transit LSR hop, the MPLS TTL value is decremented by one. When the MPLS TTL value reaches one or zero, the packet is discarded.

At the MPLS router that pops the label (either the penultimate LSR or the egress LER), the incoming packet's MPLS TTL value is copied to the packet's IP TTL field, the IP TTL field is decremented by one, and the checksum is re-calculated. The result is that each LSR in the MPLS domain is counted as one hop. This is the default behavior.

Optionally, the user can configure TTL propagation so that the entire MPLS domain appears as a two hops. In this case, the ingress LER decrements the IP packet's TTL value by one and then places a value of 255 in the packet's MPLS TTL field. The MPLS TTL value is decremented by one as the MPLS packet passes through each LSR in the MPLS domain. When the label is popped, the value in the MPLS TTL field is discarded, not copied to the packet's IP TTL field. The unlabeled IP packet's TTL is then decremented by one as it passes through the egress LER. This means that the packet's IP TTL is decremented twice from the time it enters the ingress LER to the time it exits the egress LER, making the MPLS domain appear as two hops.

To configure TTL propagation so that the entire MPLS domain appears as two hops, enter the following commands on both the ingress LER and the MPLS router that pops the label (either the penultimate LSR or the egress LER).

```
device(config-mpls)# policy
device(config-mpls-policy)# no propagate-ttl
```

Syntax: [no] propagate-ttl

When **no propagate-ttl** is configured, the ingress LER places a value of 255 into the packet's MPLS TTL field, regardless of the TTL value in the packet's IP header. The packet's IP TTL value is decremented twice: once at the ingress LER and once at the egress LER. With this option, the entire MPLS domain (regardless of the number of transit LSR hops) counts as two hops for the IP TTL value.

> **NOTE**
> When the user chooses to configure TTL propagation in this way, it is important that the user enters the **no propagate-ttl** command at both the ingress LER and the MPLS router that pops the label. When the user omits the **no propagate-ttl** command at the MPLS router that pops the label, the value in the packet's MPLS TTL field would be copied into the packet's IP TTL field. This value could be as high as 255.

Ingress tunnel accounting

Ingress tunnel accounting provides the ability to count the number of traffic bytes and packets forwarded through a specified LSP. Ingress tunnel accounting is supports the following:

- RSVP-signaled LSPs
- LDP signaled LSPs

SNMP agent support for ACL accounting

The SNMP agent supports the LSP tunnel byte count in MIB ifHCOutOctets, and LSP tunnel packet count in MIB ifHCOutUcastPkts within the ifXTable.

```
device(config-router-mpls-policy) ingress-tunnel-accounting
```

LSP accounting statistics for single-hop LSP routes

LSP accounting statistics for single-hop LSP routes are collected on MLX Series and XMR Series series devices. On an ingress LER, the number of packets and traffic bytes forwarded through a single-hop LSP are collected. LSP accounting statistics are also collected for a single-hop RSVP tunnel carrying multiple LDP cross connections. Data traffic and other control protocol traffic that is forwarded by the CPU is not accounted for on single-hop LSP tunnels. LSP accounting for a single-hop LSP is depicted in Figure 19.

> **NOTE**
> The **show mpls statistics lsp** command displays LSP accounting statistics for single-hop and multi-hop LSP routes.

In the case of a LSP switchover from a multi-hop LSP to a single-hop LSP, LSP accounting statistics are accounted for in the following scenarios.

- In the case of an adaptive LSP, a new LSP path is being set up and traffic is re-routed onto the new path from a multi-hop to a single-hop LSP without disabling the existing LSP path.
- For a one-to-one *Fast Reroute (FRR)* failover from a multi-hop protected LSP to a single-hop detour LSP, the LSP riding on the detour path is accounted for.
- In the case of a switchover from a primary LSP (multi-hop LSP) to a secondary LSP (single-hop LSP), the LSP accounting statistics on the secondary LSP are maintained only when the secondary LSP is a hot-standby. When the secondary LSP in a hot-standby mode is already UP, the LSP does not go down in a switchover, and the LSP accounting statistics from the primary LSP are accounted for on the secondary LSP.

Figure 19 depicts a single-hop LSP where a back-to-back connected link exits between the primary LSP and secondary LSP.

FIGURE 19 Single-hop LSP
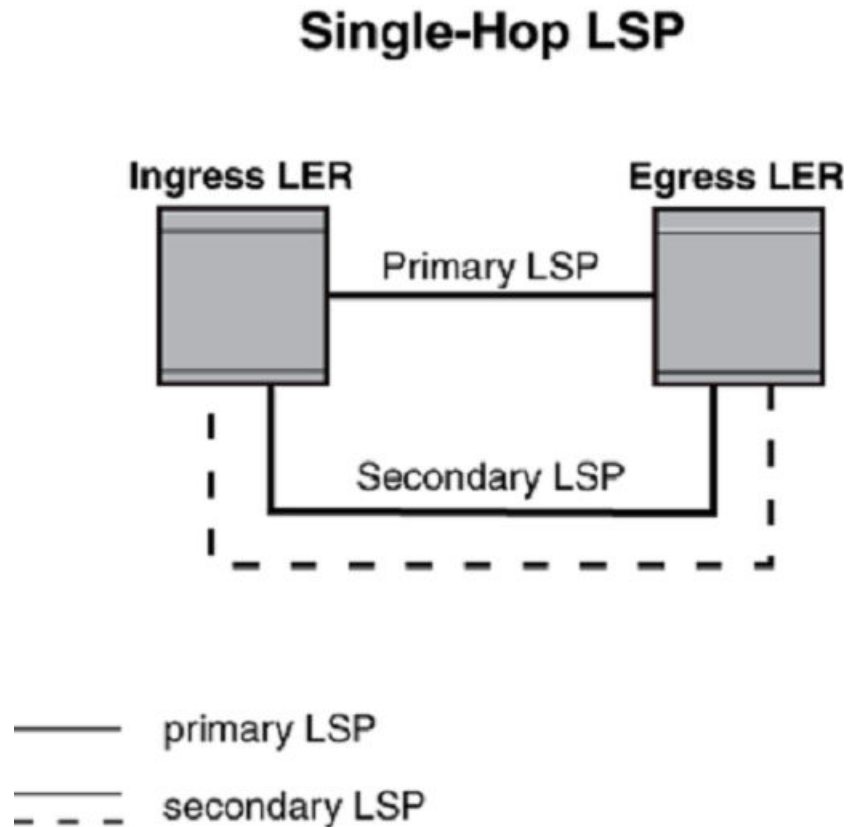
In the case of facility bypass LSP failover, statistics for each individual LSPs riding over the bypass LSP tunnel are collected. Figure 20 depicts an MPLS domain for single-hop bypass LSP tunnel where the merge point is not the egress LER. The accounting statistics for the bypass LSP are not collected. However, the statistics for the individual LSPs riding over the single-hop bypass LSP tunnel are collected. For example, when there are two protected LSP routes riding over a single-hop bypass LSP tunnel and the merge point is not the egress LER, the individual statistics for the two protected LSP routes are collected, and can be displayed using the **show mpls statistics lsp** command.

FIGURE 20 Single-hop bypass LSP
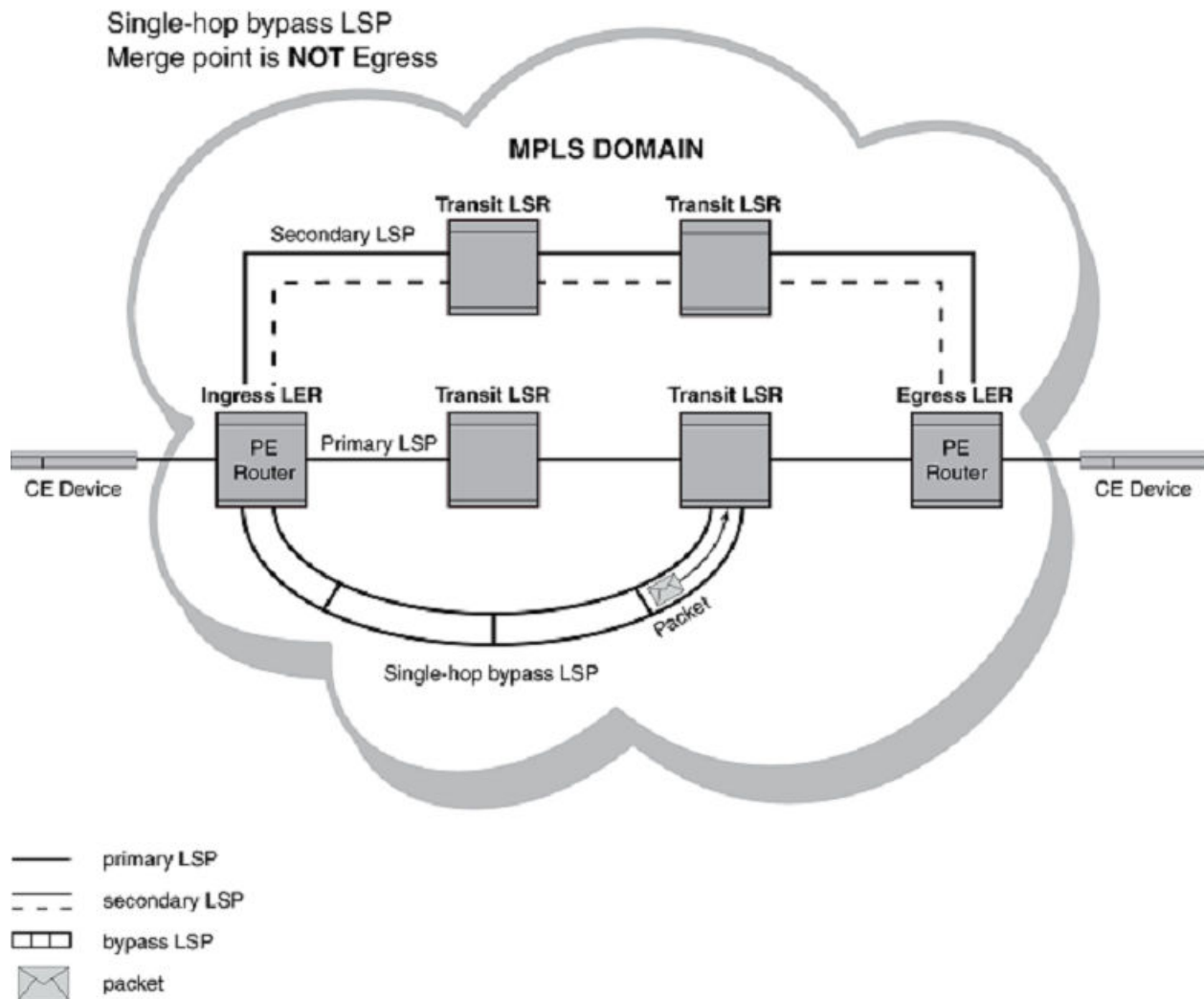
Figure 21 depicts an MPLS domain for a multi-hop bypass LSP tunnel where the merge point is the egress LER. As with the single-hop bypass LSP scenario, the statistics for the individual LSPs riding over the multi-hop bypass LSP tunnel are accounted for. Individual LSP accounting statistics are collected on all LSP routes riding over a single-hop or multi-hop bypass LSP tunnel even when the merge point is the egress LER or not. For example, when there are three protected LSP routes riding over a multi-hop bypass LSP tunnel and the merge point is the egress LER, the individual statistics for all three protected LSP routes are collected.

FIGURE 21 Multi-hop bypass LSP
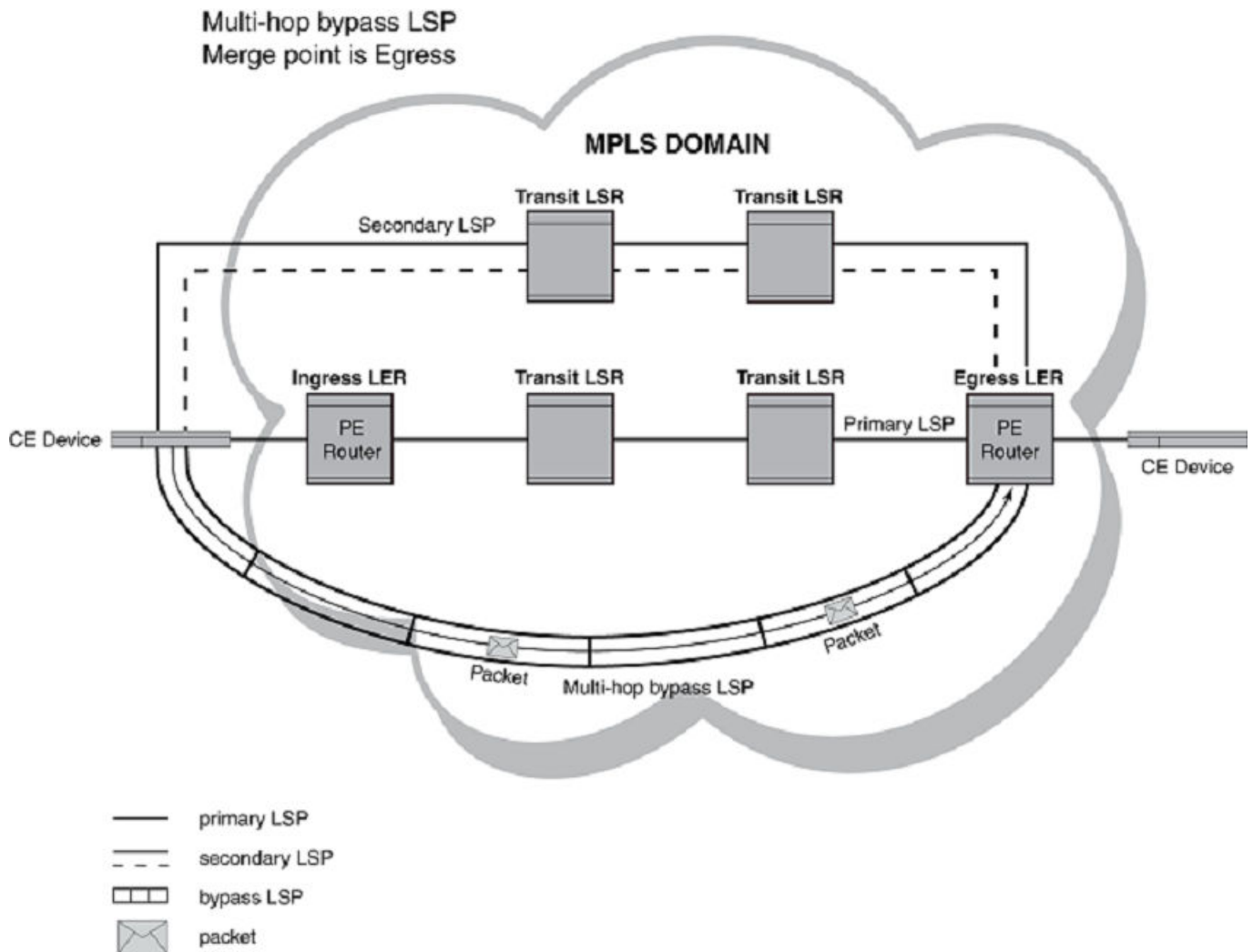
Global RSVP parameters

RSVP is automatically enabled when MPLS is enabled on the device. The user can optionally configure the following RSVP parameters globally at the config-router-mpls level:

- Refresh interval
- Refresh multiple

The user can also optionally configure interface-specific RSVP behaviors (RSVP authentication, RSVP reliable messaging, and RSVP refresh reduction) at the interface level.

> **NOTE**
> The effect of the **refresh-interval** and **refresh-multiple** commands can be overridden by RSVP refresh reduction behaviors.

Configuring the RSVP refresh interval

To maintain path states and resource reservations on the routers in an LSP, RSVP Path and Resv messages are sent at regular intervals. Path messages flow downstream in an LSP, from the ingress LER towards the egress LER. Resv messages flow upstream, in the reverse direction of Path messages.

The user can control how often the Path and Resv messages are sent by setting the refresh interval. By default, the refresh interval is 30 seconds. The user can set the refresh interval from 0 through 2147483 seconds.

use the following commands to set the refresh interval to 20 seconds.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# rsvp
device(config-router-mpls-rsvp)# refresh-interval 20
```

Syntax: [no] refresh-interval *seconds*

Configuring the RSVP refresh multiple

When refresh messages are not received, RSVP path states and resource reservations are removed from the routers in an LSP. By default, the device waits the length of three refresh intervals; when no refresh message is received by the end of that time, the path state or resource reservation is removed.

The refresh multiple is the number of refresh intervals that must elapse without a refresh message before a path state or resource reservation times out. By default, the refresh multiple is three intervals. The user can set the refresh multiple from zero through 65535 intervals.

Use the following commands to set the refresh multiple to five intervals.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# rsvp
device(config-router-mpls-rsvp)# refresh-multiple 5
```

Syntax: [no] refresh-multiple *intervals*

MPLS LSP history in descending order

Glossary

Terms used in the MPLS LSP history in descending order feature.

Term	Meaning
LSP	**L**abel **S**witched **P**ath
LDP	**L**abel **D**istribution **P**rotocol
RSVP	**R**esource Re**S**er**V**ation **P**rotocol

Specifications

Describes the details for MPLS LSP history in descending order.

LSP History by default is displayed in chronological order (oldest entries on top of the display). This feature implements an option to the display the history entries in reverse chronological order as well. This way the user can avoid having to scrolling down to the end of the list every time to check the latest events which may have caused issues for one LSP.

Customer configurations

Shows current and descending options for LSP history.

The current output of "**show mpls lsp name** *lsp_name* **extensive** command is as below:

```
device#show mpls lsp name R46-to-R52-p6-1 ext
LSP R46-to-R52-p6-1, to 172.16.52.1
  From: 172.16.46.1, admin: UP, status: UP, tunnel interface(primary path): tnl0
  Times primary LSP goes up since enabled: 63
...
....
History
  576 Jan  9 15:25:03 : Primary path  instance_id 2. RRO received:
                           -> 172.31.46.3 -> 172.31.26.6 -> 172.31.26.33
                           -> 172.31.52.4
  577 Jan  9 15:25:03 : Primary path  setup successful . Instance id 2
  578 Jan  9 15:25:03 : LSP tunnel is UP with Primary path  as Active
  579 Jan  9 15:25:03 : Tunnel added or updated, out-interface: e2/1, out-label 13385
  .......
  604 Jan  9 16:00:34 : Primary path  setup successful . Instance id 2
  605 Jan  9 16:00:34 : LSP tunnel is UP with Primary path  as Active
  606 Jan  9 16:00:34 : Tunnel added or updated, out-interface: e2/1, out-label 29594
  607 Jan  9 17:03:43 : Re-optimization timer expired for Primary path
device#
```

There is now an option to display the entries in descending order as below:

```
device#show mpls lsp name R46-to-R52-p6-1 ext descending
LSP R46-to-R52-p6-1, to 172.16.52.1
  From: 172.16.46.1, admin: UP, status: UP, tunnel interface(primary path): tnl0
  Times primary LSP goes up since enabled: 63
...
....
History
  607 Jan  9 17:03:43 : Re-optimization timer expired for Primary path
  606 Jan  9 16:00:34 : Tunnel added or updated, out-interface: e2/1, out-label 29594
  605 Jan  9 16:00:34 : LSP tunnel is UP with Primary path  as Active
…
…

  578 Jan  9 15:25:03 : LSP tunnel is UP with Primary path  as Active
  577 Jan  9 15:25:03 : Primary path  setup successful . Instance id 2
  576 Jan  9 15:25:03 : Primary path  instance_id 2. RRO received:
                           -> 172.31.46.3 -> 172.31.26.6 -> 172.31.26.33
                           -> 172.31.52.4
  device#
```

RSVP message authentication

Support was added for RSVP message authentication using MD5 as described in *RFC 2747*. It is implemented on the Extreme devices to prevent spoofing of RSVP messages. *RFC 2747* defines the use of a message digest carried in the RSVP INTEGRITY object. This object carries the following information:

- Key ID: An 8-bit number unique to a given sender

- Sequence Number: A 64-bit monotonically increasing sequence number
- Keyed Message Digest: As implemented here using MD5, it is a 16-bit message digest

In order to support *RFC 2747*, this implementation supports the following:

- An authentication type using the MD5 cryptographic algorithm
- An authentication key for use with the authentication algorithm
- An authentication window of one (1), which specifies that the maximum number of authenticated messages that can be received out of order is one (1)

Configuring RSVP message authentication

RSVP message authentication is disabled by default. This authentication method uses MD5 and is configured within the MPLS configuration mode.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ethernet 1/1
device(config-mpls-if-e100-1/1)# rsvp-authentication key administrator
```

Syntax: [no] rsvp-authentication key *string*

The *string* variable specifies a text string of up to 64 characters that is encrypted and used for RSVP message authentication.

By default, the authentication key is encrypted. When the user wants the authentication key to be in clear text, insert a **0** between **key** and *string* .

```
device(config-mpls-if-e100-1/1)# rsvp-authentication key 0 administrator
```

The software adds a prefix to the authentication key in the configuration. For example, the following portion of the code has the encrypted code "2".

```
rsvp-authentication
```

```
2 key $IUA2PWc9LW9VIW9zVQ=="
```

The prefix can be one of the following:

- 0 = the key string is not encrypted and is in clear text
- 1 = the key string uses proprietary simple cryptographic 2-way algorithm (only for CES 2000 Series and CER 2000 Series devices)
- 2 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for XMR Series and MLX Series devices)

RSVP reliable messaging

The standard RSVP periodically re-sends Resv and Path refresh messages to maintain the state of the path, but many trigger messages signaling a new or changed state (such as PathTear and ResvTear messages) are sent only once. The loss of such a message can cause delays in the reservation or release of resources.

RFC 2961 provides extensions to the RSVP to make the transmission of RSVP trigger messages more reliable by creating an ID for each new RSVP message and allowing the sender to request an acknowledgment of the receipt of trigger messages.

Configuring RSVP reliable messaging

When RSVP reliable messaging is enabled on an interface of the Extreme device, RSVP trigger messages sent out on that interface includes a message ID and a request for acknowledgment from the RSVP neighbor. When acknowledgment is not received, the trigger message is re-transmitted using the retransmission parameters configured on the interface.

> **NOTE**
> RSVP refresh messages never require acknowledgment, even when reliable messaging is enabled.

To enable RSVP reliable messaging on an interface, use commands such as the following.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ethernet 3/13
device(config-mpls-if-e1000-3/13)# rsvp-reliable-messaging
```

The previous commands enable RSVP reliable messaging on interface 3/13 with all parameters set to their defaults (or to settings previously configured on this interface, if any).

Syntax: [no] rsvp-reliable-messaging [rapid-retrans-interval *milliseconds*] [rapid-retry-limit *number*] [rapid-retrans-decay *percentage*]

The **rapid-retrans-interval** option allows the user to specify the interval in milliseconds that the device waits before a message is first resent when no acknowledgment is received. The range is from 100 through 30000 milliseconds, and the default is 500.

The **rapid-retry-limit** option allows the user to specify the maximum number of times a message is to be resent when no acknowledgment is received. The range is from one through 16, and the default is two (2). When set to the default value, a message is sent a total of three times when no acknowledgment is received.

The **rapid-retrans-decay** option allows the user to specify the percentage increase in the interval between successive re-transmissions of an unacknowledged RSVP message. The range is from zero through 100 percent, and the default value is 100 percent. A value of zero percent provides a constant retransmission rate with each interval being equal to the **rapid-retrans-interval** value. A value of 100 percent doubles the retransmission interval after each retry.

RSVP refresh reduction

RSVP control traffic (Path and Resv messages) is initially propagated to establish an RSVP session and reserve resources along the path, or to signal a change of state *(trigger messages)*. However, because it is a soft-state protocol, RSVP also requires periodic refreshing to prevent reserved resources from aging out. The original RSVP as defined in *RFC 2205* achieves this by re-sending identical Path and Resv messages *(refresh messages)* at regular intervals along the reserved path as long as the RSVP session remains unchanged. The bandwidth and processing time required to support these refresh messages increases linearly as more RSVP sessions are established, which can result in scaling problems.

RFC 2961 establishes extensions to RSVP which can help reduce the overhead caused by refresh messages: bundle messages, which allows multiple RSVP messages to be aggregated into a single PDU, and *summary refresh messages*, which replace identical RSVP message re-transmissions with a list of the IDs of all Path and Resv states to be refreshed. Both extensions are available at the interface configuration level on the XMR Series, MLX Series, CER 2000 Series and CES 2000 Series platforms, and each extension can be configured separately.

When the user enables either of the refresh reduction extensions on an interface, outgoing RSVP packets sent on that interface sets the refresh reduction capability bit in the common RSVP header to indicate that the Extreme device is capable of receiving and processing refresh reduction messages and related objects.

Configuring RSVP bundle messages

When RSVP bundle messages are enabled on an interface, the device attempts to combine multiple outgoing RSVP messages on that interface into bundles to reduce overhead.

RSVP bundle messages are disabled by default for all interfaces. To enable bundle messages on an interface, use commands such as the following.

```
device(config)# router mpls
device(config-mpls)# mpls-interface eth 3/13
device(config-mpls-if-e1000-3/13)# rsvp-refresh-reduction bundle-message bundle-send-delay 20
```

The previous commands enable RSVP bundle messages on interface 3/13 with a bundle-send-delay of 20 milliseconds.

Syntax: [no] **rsvp-refresh-reduction bundle-message** [**bundle-send-delay** *milliseconds*]

The **bundle-send-delay** option specifies the maximum period (in milliseconds) that an outgoing message can be delayed in order to create a multi-message bundle before sending. This delay is retained for the interface even when bundle messages are disabled.

When the RSVP neighbor does not support refresh reduction, the interface does not bundle messages even when bundle messages are locally enabled. Use the [**no**] version of the command to disable RSVP bundle messages on this interface.

> NOTE
> Summary refresh is a more effective tool for RSVP refresh message overhead
> reduction.

Configuring RSVP summary refresh

RFC 2961 extends RSVP to create IDs for RSVP messages. When RSVP summary refresh is enabled on an interface, the device suppresses the sending of unchanged Path and Resv messages (refresh messages) and instead sends a summary message listing the IDs of Path and Resv messages that are to be refreshed. Summary refresh does not affect the sending of RSVP trigger messages that signal changes of state.

RSVP summary refresh is disabled by default for all interfaces. To enable summary refresh on an interface, use commands such as the following.

```
device(config)# router mpls
device(config-mpls)# mpls-interface eth 3/13
device(config-mpls-if-e1000-3/13)# rsvp-refresh-reduction summary-refresh
```

The previous commands enable the sending of RSVP summary refresh messages on interface 3/13.

Syntax: [no] **rsvp-refresh-reduction summary-refresh**

When the RSVP neighbor does not support refresh reduction, the interface does not send summary refresh messages, even though the feature is locally enabled. It instead continues to resend full Path and Resv refresh messages. Use the [**no**] version of the command to manually disable summary refresh on this interface.

Enabling both RSVP refresh reduction extensions in a single step

Bundle messages and summary refresh are disabled by default for all interfaces. The user can enable both extensions with default parameters on an interface by using commands such as the following.

```
device(config)# router mpls
device(config-mpls)# mpls-interface eth 3/13
device(config-mpls-if-e1000-3/13)# rsvp-refresh-reduction all
```

The previous commands enable bundle messages and summary refresh on interface 3/13 with the default bundle-send-delay setting of 40 milliseconds (or the previously configured value, when one has been set for this interface) and the refresh interval set for RSVP at the global level.

Syntax: **rsvp-refresh-reduction all**

When the RSVP neighbor does not support refresh reduction, the interface does not send bundled messages or summary refresh messages, even though the extensions are enabled. Use the [**no**] version of the command to manually disable both extensions on this interface.

Displaying refresh reduction information for an interface

The user can display RSVP refresh reduction settings for an interface by using the following command at any level of the CLI.

```
device# show mpls rsvp interface
```

For detailed information about this command and what it displays, refer to the show mpls rsvp interface command in the *MPLS Commands* chapter.

RSVP IGP synchronization

The RSVP IGP synchronization feature enables RSVP to react to an IGP neighbor down event. This feature can help improve the convergence time of RSVP and reduce the latency in removing the resource reservations, thereby improving the overall network efficiency.

When an IGP protocol declares a neighbor down, because hello packets are no longer being received, RSVP brings down all the associated LSPs and sessions that are passing through the down neighbor. However, the IGP protocols and RSVP still act independently when bringing a neighbor up.

RSVP sessions are maintained until either the router stops receiving IGP hello packets or the RSVP Path and Resv messages time out or RSVP states are explicitly torn down by the ingress or egress. Configuring a short time for the IS-IS hello timers allows these protocols to detect node failures quickly. Also, configuring BFD for IGP interface provides sub-second neighbor down detection time. When quick discovery of a failed neighbor is needed, short IGP (IS-IS) hello timers could be configured, or BFD could be enabled on IGP interfaces.

Limitations

The RSVP IGP synchronization feature allows RSVP to react to an IGP neighbor down event. It does not allow RSVP to detect that a neighbor node has gone down. For example, when a pair of RSVP/IGP routers are connected with parallel links, detecting one neighbor down does not actually mean that the entire neighbor node has gone down.

Globally enabling RSVP IGP synchronization

This command globally enables the handling of an IGP neighbor down event by MPLS. This command can be executed on the fly and takes effect immediately. It is possible to enable handling of neighbor down events for IS-IS.

Configuration example

By default, RSVP does not handle IGP neighbor down events. RSVP IGP synchronization must be enabled to handle an IGP neighbor down event.

To configure RSVP IGP synchronization feature, the following commands need to be executed. The following command enables RSVP to handle IGP neighbor down events for IS-IS.

Note that commands to configure basic MPLS are not included below.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# handle-isis-neighbor-down
```

Syntax: [no] [**handle-isis-neighbor-down**]

Limitations

1. This feature is independent of MPLS traffic engineering configurations. Irrespective of MPLS traffic engineering configuration (OSPF or IS-IS), this feature allows MPLS (RSVP) to handle IGP neighbor down events and take action, such as tearing down the associated RSVP sessions. For example, when IS-IS is configured as MPLS TE protocol, the user can still configure MPLS to handle an OSPF neighbor down event (and vice versa).

2. An IGP neighbor down event is handled only by the RSVP sub-component of MPLS by tearing down the associated sessions. This event is not handled by LDP sub-component of MPLS.

3. MPLS/RSVP does not keep track of the current state of IGP neighbor. That is, when an IGP neighbor goes down, RSVP tears down all the associated sessions. But RSVP does not prevent bringing up any session while the IGP neighbor to RSVP next-hop is down (or not yet available). That is, the RSVP session is brought up even when the IGP neighbor to the next-hop does not exist.

4. An IGP neighbor down is treated as upstream neighbor down or downstream neighbor down event by RSVP, depending upon the direction of the LSP. When a downstream IGP neighbor goes down, it results in an LSP teardown or FRR switchover, whichever is applicable.

5. MPLS receives and processes an IGP neighbor down event only for the cases when an IGP neighbor goes down because of hellos not received from the peer.

6. When an IGP neighbor goes down because of an underlying interface down, MPLS does not react to an IGP neighbor down event as RSVP would also receive the interface down event and tears down associated LSPs/sessions. Handling an IGP neighbor down event is redundant in such situations.

7. When BFD is configured on IGP interfaces, an IGP neighbor down is detected quickly and may help RSVP converge faster.

8. Bypass LSPs are treated exactly the same way as regular LSPs. Upon an IGP neighbor down, associated bypass LSPs is torn down.

9. When an IGP neighbor is Nonstop Routing or Graceful Restart (NSR/GR) capable, MPLS does not receive a neighbor down event when NSR is performed on the peer IGP router.

10. Faster FRR feature is not be triggered when MPLS detects that IGP neighbor is down. Instead, each FRR LSP is processed individually to perform local repair.

11. It is highly recommended to observe extreme caution when implementing this feature when BFD is enabled for the underlying IGP. Under some circumstances, unnecessary flapping for RSVP sessions/LSPs can occur with this combination.

RSVP IGP synchronization for Remote Links

The RSVP IGP Synchronization-Phase II feature enables an LSP ingress router to react to neighbor down events from any location in the network. Any non-FRR or Bypass LSP can be rerouted when the router receives an IGP link or neighbor down event.

MPLS will build an IGP Sync database which is independent of the MPLS TE database. The IGP Sync data base links are keyed based on IPv4 address pair (Link IP, Router-ID of Links remote end). The link will be added or updated in the database whenever an LSP Path comes UP. The IGP Link gets deleted or updated whenever an LSP path goes DOWN or an IGP link down event gets generated. The IGP Link database links are associated with individual LSP instances, such that whenever an IGP link down event reaches MPLS, MPLS can correlate it to individual LSP paths.

An IGP Sync-phase II Link down event can lead to any one of the below actions:

- Bring down an LSP and retry it (setup on new path avoiding failed link) OR
- Bring down an LSP and switch to any other instance (secondary path) which is already up OR
- Create a new instance of the LSP first and then switch to the new instance.

Limitations and pre-requisites

- This feature will not be usable when Traffic engineering is not enabled in any of the routers used by the LSP
- No additional or separate configurations are required to enable or disable Phase-I or Phase-II separately
- For full functionality of this feature, it is recommended that:
    - All routers used by all the non-FRR and Bypass LSPs must be Traffic Engineering enabled
    - All Transit routers shall support RRO

Configuring MPLS on a VE interface

To enable MPLS on a VE interface, first create a VE interface and configure an IP address for the VE as shown in the example below.

```
device(config)# vlan 100
device(config-vlan-100)# tagged ethernet 2/1
device(config-vlan-100)# router-interface ve 100
device(config-vlan-100)# exit
device(config)# interface ve 100
device(config-vif-100)# ip address 10.10.10.1/24
device(config-vif-100)# exit
```

Then enable MPLS on the VE interface as shown in the example below.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ve 100
device(config-mpls-if-ve-100)#
```

Syntax: [no] mpls-interface [ve *ve-id*]

The **mpls-interface ve** parameter allows the user to enable MPLS on a VE interface. The *ve-id* variable allows the user to specify a VE interface ID. The **no mpls-interface ve** command removes all configuration for MPLS on a VE enabled interface.

The following MPLS commands are available on a VE interface under the mpls interface configuration mode.

- admin-group - Adding an MPLS VE interface to an administrative group on page 115
- ldp-enable - Configuring LDP on an MPLS VE interface on page 115
- ldp-params - Setting the LDP hello interval on an MPLS VE interface (link only) on page 115
- hello-interval - Setting the LDP hello interval on an MPLS VE interface (link only) on page 115
- hello-timeout - Setting the LDP Hello Holdtime on an MPLS VE interface (link only) on page 116
- reservable-bandwidth - Bandwidth computation for an MPLS VE interface on page 116
- rsvp-authentication - Configuring RSVP message authentication on an MPLS VE interface on page 117
- exclude-interface - Specifying a bypass LSP for an MPLS VE interface on page 117

Adding an MPLS VE interface to an administrative group

The user can place individual interfaces into administrative groups. Administrative groups, also known as resource classes or link colors, allows the user to assign MPLS enabled VE interface to various classes. For more information on assigning an MPLS-enabled interface to an administrative group, refer to Adding interfaces to administrative groups on page 101.

To assign MPLS interface ve 100 to an administrative group called "gold", enter the following.

```
device(config-mpls)# mpls-interface ve 100
device(config-mpls-if-ve-100)# admin-group gold
```

Syntax: [no] admin-group *number* | *name* [*number* | *name*]

The *number* variable can be from 0 - 31. The administrative group *name* variable must have been previously configured. By default, no admin group is configured for any MPLS interfaces, including MPLS VE interface.

An MPLS enabled VE interface can belong to any number of administrative groups. For example, to assign an MPLS interface ve 100 to group "gold" and group 31, enter commands such as the following.

```
device(config-mpls)# mpls-interface ve 100
device(config-mpls-if-ve-100)# admin-group gold 31
```

Configuring LDP on an MPLS VE interface

> **NOTE**
> For more information on configuring LDP on physical interfaces, refer to Configuring LDP on an MPLS VE interface

To configure LDP on MPLS interface ve 100, enter commands such as the following.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ve 100
device(config-mpls)# ldp-enable
```

Syntax: [no] ldp-enable

The [**no**] option removes LDP on an MPLS interface, including LDP on an MPLS VE interface.

Setting the LDP hello interval on an MPLS VE interface (link only)

> **NOTE**
> For more information on setting the LDP hello interval on physical interfaces, refer to Setting the LDP hello interval on an MPLS VE interface (link only)

The user can set the LDP Hello Interval on an MPLS enabled VE interface. This option is only available for Link LDP sessions. The following example configures LDP hello-interval to 30 seconds for MPLS interface ve 100.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ve 100
device(config-mpls-if-ve-100)# ldp-params
device(config-mpls-if-ve-100-ldp-params)# hello-interval 30
```

Syntax: [no] hello-interval *seconds*

The *seconds* variable specifies the value in seconds of the Hello Interval that the user is configuring on an MPLS VE interface for LDP Link Hello messages. The LDP hello interval can be from 1 - 32767 seconds. The default value for LDP hello interval is five seconds.

The [**no**] option removes a previously configured LDP Hello Interval.

Setting the LDP Hello Holdtime on an MPLS VE interface (link only)

> **NOTE**
> For more information on setting the LDP Hello Holdtime on physical interfaces, refer to Setting the LDP Hello Holdtime on an MPLS VE interface (link only).

The user can set the LDP Hello Holdtime on an MPLS enabled VE interface. This Holdtime value is sent in Hello messages from the interface. This option is available for Link LDP sessions only. The following example configures LDP hello-timeout to 18 seconds for MPLS interface ve 100.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ve 100
device(config-mpls-if-ve-100)# ldp-params
device(config-mpls-if-ve-100-ldp-params)# hello-timeout 18
```

Syntax: [no] hello-timeout *seconds*

The value configured in the *seconds* variable is the LDP Hello Timeout value that is sent in LDP Hello messages from this interface. The range for this value is 1 - 65535 seconds. The default value is 15 seconds.

The [**no**] option removes a previously configured LDP Hello Timeout value.

Bandwidth computation for an MPLS VE interface

The maximum reservable bandwidth for a VE interface is computed based on minimum speed of all active members on a physical port. When one of the member ports is a trunk port, MPLS computes the trunk bandwidth before computing the VE bandwidth. The bandwidth of a trunk port is the sum of all active physical member ports of the trunk. For example, there are two ports (One port is 10 gig and other port is one gig), and one trunk port configured on a VE interface. The trunk is carrying two ports, and each port is one gig. To calculate the bandwidth of the trunk, the user takes the sum of all active ports on a physical port. In this example, the bandwidth of the trunk is equal to two gigs. To calculate the bandwidth of the VE interface, take the minimum of all active port members. In this example, the bandwidth of the VE interface is one gig.

Configuration Considerations

A VE interface bandwidth must be re-computed when any one of the following occurs:

- A new member port is added to a VLAN associated with a VE interface
- A new member port is removed from a VLAN associated with a VE interface
- When a member port of a VLAN associated with a VE interface is up
- When an active member port of a VLAN associated with a VE interface that is down

A physical port can be part of more than one VE interface. Each VE interface assumes that it has a full amount of reservable bandwidth for a physical port. However, the amount of reservable bandwidth on one VE is not reflected on another VE interface even though both VE interfaces share the same physical port. For example, when there are two VE interfaces; VE1 and VE2. Each VE interface supports the same amount of reservable bandwidth of 1Gbps. The amount of reservable bandwidth used to set up LSPs on VE1 is not reflected in the amount reservable bandwidth that is available on VE2. This results in an excess amount of reservable bandwidth that can be supported on a physical port. This causes data traffic to be dropped.

The default bandwidth for a VE interface is computed automatically, and is based on the underlying physical links. The user can now override the default bandwidth for a VE interface by executing the **reservable-bandwidth** command. The following example demonstrates how to override the default behavior by configuring reservable bandwidth to 400mbps on MPLS interface ve 100.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ve 100
device(config-mpls-if-ve-100)# reservable-bandwidth 400000
```

Syntax: [no] reservable-bandwidth *number*

The *number* variable refers to the amount of reservable bandwidth that is supported on an MPLS interface, including an MPLS enabled VE interface. The range for this value is 0 - 80000000 kbps. The [**no**] option removes all configuration for reservable bandwidth on an MPLS enabled VE interface.

RSVP message authentication on an MPLS VE interface

Support was added for RSVP message authentication using MD5 as described in *RFC 2747*. It is implemented on the Extreme devices to prevent spoofing of RSVP messages. All inbound RSVP messages on an interface must contain RSVP Integrity object for getting authenticated and accepted by RSVP. Inbound RSVP messages with no Integrity object, or an **incorrect** integrity object is dropped by RSVP. All outbound RSVP messages on an interface contain an RSVP Integrity object. For more information on RSVP message authentication, refer to RSVP message authentication on page 108.

Configuring RSVP message authentication on an MPLS VE interface

> NOTE
> For more information on configuring RSVP message authentication on physical interfaces, refer to Configuring RSVP message authentication on page 109.

RSVP Message Authentication is disabled by default. This authentication method uses MD5 for an MPLS VE interface. The following example configures RSVP message authentication for MPLS interface ve 100.

```
device(config)# router mpls
device(config-mpls)# mpls-interface ve 100
device(config-mpls-if-ve-100)# rsvp-authentication key private
```

Syntax: [no] rsvp-authentication key *string*

The *string* variable specifies a text string of up to 64 characters that is encrypted and used for RSVP message authentication.

Specifying a bypass LSP for an MPLS VE interface

The user can create a bypass LSP by using the **bypass-lsp** command. This is used for facility backup FRR. In the context of bypass LSP, the user can configure an MPLS interface as an exclude (protected) interface against resource failures using a bypass LSP. The user can specify a VE interface as exclude-interface. When a protected LSP egress interface is a VE interface, then any fault on a VE interface could trigger Fast Reroute. The following example configures protection for MPLS interface ve 100 using facility backup FRR.

```
device(config)# router mpls
device(config-mpls)# bypass-lsp to4
device(config-mpls-bypasslsp-to4)# exclude-interface ve 10
```

Syntax: [no] exclude-interface ethernet *slot/port* [**ethernet** *slot/port* | **to** *slot/port*] | **pos** *slot/port* [**pos** *slot/port* | **to** *slot/port*] | **ve** *vid*]

By default, a VE interface is not protected. The **ve** parameter allows the user to configure a ve interface as exclude-interface specified by *vid* .

Setting up signaled LSPs

An LSP consists of an actual path of MPLS routers through a network, as well as the characteristics of the path, including bandwidth allocations and routing metrics. There are two kinds of LSPs: signaled and static. Signaled LSPs are configured at the ingress LER. When the user enables a signaled LSP, RSVP causes resources to be allocated on the other routers in the LSP.

Configuring a signaled LSP consists of the following tasks:

- Specifying a path for the LSP to follow (optional)
- Setting parameters for the signaled LSP
- Specifying which packets are to be forwarded along the LSP (optional)

Setting up paths

A **path** is a list of router hops that specifies a route across an MPLS domain. Once the user creates a path, the user can create signaled LSPs that see the path. Paths are configured separately from LSPs so that a path may be specified once and then used by several LSPs that see the path by name. An LSP may specify a primary and one or more redundant paths.

A path is always configured at the ingress LER and assumes that the ingress LER is the beginning of the path. A path can contain any number of nodes , which correspond to MPLS-enabled routers in the network. Each node has one attribute: whether it is **strict** or **loose**. A strict node means that the router must be directly connected to the preceding node. A loose node means that there can be other routers in between.

Creating a path is not absolutely necessary when configuring an LSP. When the user configures a signaled LSP without naming a path, CSPF uses only information in the *Traffic Engineering Database (TED)*, as well as the user-configured attributes and requirements of the LSP to calculate the path. When the LSP has been configured not to use CSPF, the path between the ingress and egress LERs is determined using standard hop-by-hop routing methods, as if the path consisted of a single loose node.

The following commands set up a path called sf_to_sj that has four nodes.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# path sf_to_sj
device(config-router-mpls-path-sf_to_sj)# hop 2.3.4.5 strict
device(config-router-mpls-path-sf_to_sj)# hop 1.2.3.4 strict
device(config-router-mpls-path-sf_to_sj)# exit
```

Syntax: [no] **path** *pathname*

Syntax: [no] **hop** *ipaddress* [**strict** | **loose**]

The path is assumed to start from the local node. The user specifies the nodes in order from ingress to egress. Specifying the local node itself as the first node in the path is optional. Further, the final node does not necessarily have to be the egress LER in the LSP. (The egress LER is specified at the LSP configuration level with the **to** command.) When the final node in the path differs from the egress LER, the hop between the final node in the path and the egress LER is treated as a hop to a loose node; that is, standard IP routing is used to determine the path between the final node and the egress LER.

Modifying a path

Once the user has created a path, the user can insert or delete nodes from it. For example, to delete a node from the sf_to_sj path defined above.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# path sf_to_sj
```

```
device(config-router-mpls-path-sf_to_sj)# no hop 10.1.1.1
device(config-router-mpls-path--sf_to_sj)# exit
```

Syntax: [no] hop *ipaddress*

To insert a node into a path.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# path sf_to_sj
device(config-router-mpls-path-sf_to_sj)# insert 2.3.4.5 strict before 1.2.3.4
device(config-router-mpls-path-sf_to_sj)# exit
```

Syntax: insert *ipaddress* [**loose** | **strict**] **before** *ipaddress*

> NOTE
> When the user modifies a path, the changes are not carried over to active LSPs that see the path until the LSPs are deactivated and reactivated. For example, path sj_to_sf may be used by an LSP called *lsp1*. After *lsp1* has been activated, any changes to path *sj_to_sf* do not cause the route followed by lsp1 to be modified. To get the LSP to use the modified path, the user must deactivate and then reactivate *lsp1*.

Deleting a path

To delete an entire path from the LSRs configuration, enter a command such as the following.

```
device(config-mpls)# no path sf_to_sj
```

Syntax: [no] path *pathname*

Configuring signaled LSP parameters

Once the user has configured a path, the user can configure signaled LSPs that see it. An LSPs configuration can specify not only the path that label-switched packets follow in a network, but also the characteristics of the path, the resources allocated along the path, and actions applied to the packets by the ingress or egress LERs.

The user can perform the following tasks when configuring a signaled LSP:

- Performing a Commit for an LSP
- Creating the LSP
- Specifying an egress LER for the LSP
- Specifying a primary path for the LSP (optional)
- Configuring secondary or hot-standby paths for the LSP (optional)
- Setting aliases for the egress LER (optional)
- Setting a Class of Service (CoS) value for the LSP (optional)
- Allocating bandwidth to the LSP (optional)
- Configuring the setup and hold priority for the LSP (optional)
- Setting a metric for the LSP (optional)
- Including or excluding administrative groups from LSP calculations (optional)
- Limiting the number of hops the LSP can traverse (optional)
- Specifying a tie-breaker for selecting CSPF equal-cost paths (optional)
- Disabling the Record-Route function (optional)

- Disabling CSPF path calculations (optional)
- Configure Maximum Packet Size without fragmentation
- Enabling the LSP
- Disabling the LSP
- Generating Traps and Syslogs for LSPs

Performing a commit for an LSP configuration command

For LSP configuration commands to take effect, either an explicit or implicit commit must be performed. These are performed as shown in the following:

Performing an explicit commit

The user can perform an explicit commit within the configuration of a specified LSP using the **commit** command. The following example demonstrates the creation of an LSP named **samplelsp** and its primary and secondary paths. After the configuration is entered, the commit command is executed to activate the configuration.

```
device(config)# router mpls
device(config-mpls)# lsp samplelsp
device(config-mpls-lsp-samplelsp)# primary-path pathprimary
device(config-mpls-lsp-samplelsp)# secondary-path pathsecondarya
device(config-mpls-lsp-samplelsp)# secondary-path pathsecondaryb
device(config-mpls-lsp-samplelsp)# select manual pathsecondaryb
device(config-mpls-lsp-samplelsp)# commit
```

Syntax: [no] commit

The **reoptimize** command is another type of explicit commit.

Using the **reoptimize** command, the user can activate all pending LSP configuration changes for specified LSP or use the **all** option to activate all pending LSP configuration changes for all of the LSPs configured on the router. Configuration of this command is described in Re-optimizing LSPs on page 143.

Performing an implicit commit

MPLS allows the user to modify the configurable parameters for RSVP LSPs while the LSP is operational. After modifying the parameters for an operational LSP, the user must execute the **commit** command to apply the changes. Applying these configuration changes requires a new instance of the LSP to be signaled with a modified or new set of parameters, also known as make-before-break. Once the new instance of the LSP is up, the old instance is removed.

To allow changes to be automatically applied, the user can use the **implicit-commit** command under the MPLS policy command to enable certain types of events to trigger implicit commit. When there are any changes to the configuration of the LSP, the make-before-break operation is not triggered.

```
device(config-mpls-policy)# implicit-commit autobw-adjustment
```

Syntax: [no] implicit-commit [all | autobw-adjustment | lsp-reoptimize-timer]

The **autobw-adjustment** parameter configures an implicit commit to be performed for the uncommitted changes before adjusting the bandwidth of an auto bandwidth LSP. Default is changes are not committed and bandwidth adjustment is skipped.

When the **lsp-reoptimize-timer** parameter is set, upon the re-optimization timer expiry, uncommitted changes are applied before initiating make-before-break procedure. Default is changes are not committed and re-optimization is skipped.

When the **all** parameter is set, an implicit commit is performed in all conditions before initiating make-before-break.

Creating an LSP

To create a signaled LSP and enter the LSP configuration level, enter commands such as the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)#
```

Syntax: [no] lsp *name*

Specifying the egress LER

Each LSP requires one and only one egress LER. The egress LER is the router from which packets exit the MPLS domain in this LSP. After the LSP is successfully established, the address of the egress LER is installed as an internal host route on the ingress LER, allowing the ingress LER to direct BGP next-hop traffic into the LSP. The destination address does not necessarily have to be the final node in the primary path specified for the LSP. When the final node in the path differs from the destination address, the hop between the final node in the path and the egress LER is treated as a loose hop.

To specify 10.100.1.1 as the address of the egress LER for LSP tunnel1.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# to 10.100.1.1
```

Syntax: to *ipaddress*

The egress LER is the only required parameter in an LSP. All other parameters are optional.

> NOTE
> When IS-IS is used as the IGP, the egress LER advertises the tunnel destination in Extended IP Reachability TLV 135 in order for the LSP to be properly mapped by CSPF. To ensure that this happens, connect to the egress LER and enable IS-IS on the interface which has the IP address of the tunnel destination. When none of the interfaces on the egress LER has the IP address of the tunnel destination (for example, when the tunnel destination address is the egress LER's router ID) rather than an interface address -- to manually set the router ID, then the tunnel destination address must be included in Traffic Engineering router ID TLV 134 in the LSP originated by the egress LER. This is accomplished by setting the egress LER's traffic engineering policy to IS-IS with the **traffic-engineering isis level** command (see Enabling IS-IS LSPs with TE extensions for MPLS interfaces on page 97).

Specifying a source address for an LSP

The user can optionally specify a source IP address for a signaled LSP. RSVP path messages carry this address.

To specify a source IP address of 10.2.3.4 for LSP tunnel1.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-routr-mpls-lsp-tunnel1)# from 10.2.3.4
```

Syntax: from *ipaddress*

The **from** command specifies the source IP address to be carried in RSVP Path messages for the LSP. This command is optional. When the **from** command is specified, then the address is always carried in RSVP Path messages as the source IP address for the LSP. When the **from** command is not specified, then when the LSP is enabled, the device dynamically determines the source address of the LSP (using the device's router ID or the address of the first loopback as the source address).

Note that the IP address specified in the **from** command affects only the address carried in the RSVP Path messages for the LSP. It does not affect the outgoing interface (and thus the actual path) that the Path messages are sent out.

Specifying the primary path for an LSP

The primary path is the route that packets generally travel when going through an LSP. The user can specify a user-defined path or no path at all. Refer to Setting up paths on page 118 for information on defining a path. Once the LSP is enabled, the ingress LER attempts to signal the other LSRs in the path so that resources can be allocated to the LSP. When the user does not specify a primary path, the path used in the LSP is the shortest path to the egress LER, as determined from standard IP routing methods, or CSPF when it is enabled.

To specify the sf_to_sj path as the primary path for LSP tunnel1.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# primary-path sf_to_sj
```

Syntax: [no] **primary-path** *pathname*

Configuring redundant paths for an LSP

> **NOTE**
> This section describes the behavior of redundant paths. However, the user can exercise further control over the path selection process by specifying the path selection mode and preferred path using the **select-path** command. This process is described in detail in Configuring path selection on page 123.

A signaled LSP has a primary path, which is either user-defined or computed by the ingress LER. Optionally, the user can configure one or more redundant paths to serve as a backup. When the primary path fails, traffic for the LSP can be forwarded over the redundant path. When no redundant path is configured for the LSP, when the primary path fails, the ingress LER automatically attempts to compute a new path to the egress LER, establish the new path, and then redirect traffic from the failed path to the new path.

Configuring a redundant path allows the user to exercise greater control over the rerouting process than when the ingress LER simply calculated a new path to the egress LER. When a redundant path is configured, when the primary path fails, the ingress LER attempts to establish the redundant path. As with the primary path, a redundant path follows an explicit route of loose or strict hops.

By default, the redundant path is established only when the primary path fails. The user can optionally configure a redundant path to operate in **hot-standby** mode. A hot-standby path is established at the same time the primary path in the LSP is established. Resources are allocated to the hot-standby path, although no packets for the LSP are sent over the hot-standby path until the primary path fails. When the primary path fails, the already-established hot-standby path immediately takes over from the primary path. Since the hot-standby path is already active, service outages that can arise from the process of signaling and establishing a new path are eliminated.

After the redundant path has been activated, the ingress LER continues to try to connect to the egress LER over the primary path, either indefinitely or up to the configured retry limit. When a connection over the primary path can be established, the redundant path is deactivated, and traffic for the LSP is again sent over the primary path. Once the primary LSP becomes available again, the redundant path is torn down; when the path is a hot-standby path, it reverts to its backup status.

The user can configure multiple redundant paths. When the primary path fails, the ingress LER attempts to establish a connection to the egress LER using the first redundant path configured for the LSP. When a connection cannot be established using the first redundant path, the second redundant path is tried, and so on. When a connection cannot be established after trying each redundant path in the configuration, the first redundant path is tried again, and the process repeats. (This behavior can be further modified using the **select-path** command; see Configuring path selection on page 123.)

To configure a secondary path, first create a path, as described in Setting up paths on page 118. After the user creates the path, the user can specify that it is to be used as a redundant path. For example, the following commands cause a path called alt_sf_to_sj to be used when the primary path in LSP tunnel1 fails.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# secondary-path alt_sf_to_sj
device(config-router-mpls-lsp-tunnel1-sec-path)#
```

Syntax: [no] secondary-path *pathname*

Issuing the **secondary-path** command enters the secondary path configuration level. From this level, the user can specify that this path is to operate in hot standby mode.

```
device(config-router-mpls-lsp-sec-path)# standby
```

Syntax: [no] standby

Once the LSP is enabled, both the primary and hot-standby paths are activated, although packets are directed over only the primary path.

> **NOTE**
> At the secondary path level, the user can configure separate values for the following parameters: *Class of Service (CoS)*, setup and hold priority, bandwidth allocations, and inclusion or exclusion of interfaces in administrative groups. When the user does not configure these parameters at the secondary path level, the secondary path uses the default values for these parameters.

Configuring path selection

The user can exercise further control over the paths used by an LSP by setting the select mode and by specifying a preferred path using the **select-path** command as described below.

By default, an LSP with primary and secondary paths configured immediately uses the primary path. When the primary path fails, a secondary (redundant) path is used. When the primary path comes back up, traffic reverts to the primary path and the secondary (redundant) path returns to a back-up state. However, path selection can be configured to operate in any of the following three modes:

- **auto select mode** - This is the default mode of the router and no special configuration is required. When this mode is operating, the router always tries to use the primary path to carry traffic when the primary path has stayed operating in the working state for at least the amount of time specified in **revert-timer** configuration command. When [**no**] **revert-timer** is configured for the LSP, a value of zero second is used which causes immediate switching of the path.

- **manual select mode** - In this mode, traffic is switched to a user-specified path after the selected path has stayed operating in the working state for at least the amount of time specified in **revert-timer** configuration. In **manual select** mode, traffic stays on the selected path as long as the path remains in working condition and only switches to an alternative path, such as the primary path, when the selected path experiences a failure. Once the selected path comes back into working condition for the amount of time specified by the revert-timer configuration, traffic is switched back to it.

When an LSP is configured in manual select path mode with at least one other hot standby secondary path, the operation is as follows: when the selected path goes down, the system tries to bring up one hot standby secondary path to protect the primary path, but when the selected path is up, system brings down the hot standby secondary path since the selected path is already serving as a hot standby for the primary path.

- **unconditional select mode -** In this mode, traffic is switched to and stays on the selected path regardless of the path's condition even when it is in a failure state. The main difference between manual and unconditional select mode is the test of the working condition of the user selected path. When configured in unconditional mode, the router starts the signaling for the selected path if has not already done so and brings down all other paths; this includes the primary path and the path carrying traffic when it is

not the selected path. Because the speed at which the selected path comes up cannot be guaranteed, traffic forwarding might be disrupted.

> **NOTE**
> The **auto-select** and **manual-select** mode configurations use the **revert-timer** configuration that is described in Configuring a Path Selection Revert Timer on page 124.

The following example configured the LSP named **samplelsp** with a primary path named **pathprimary** and two secondary paths named **pathsecondarya** and **pathsecondaryb** . The path named **pathsecondaryb** is configured as a selected path in the **manual select** mode.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp samplelsp
device(config-router-mpls-lsp-samplelsp)# primary-path pathprimary
device(config-router-mpls-lsp-samplelsp)# secondary-path pathsecondarya
device(config-router-mpls-lsp-samplelsp)# secondary-path pathsecondaryb
device(config-router-mpls-lsp-samplelsp)# select-path manual pathsecondaryb
device(config-router-mpls-lsp-samplelsp)# commit
```

After configuring this example, traffic for **samplelsp** travels over the **pathsecondaryb** path whenever this path is in working condition because no revert-timer has been configured. When a revert-timer is configured, the router waits for the **pathsecondaryb** path to be up for at least the amount of time specified in the configuration of the **revert-timer** command. When the select mode is changed to **unconditional**, as shown in the following, traffic is switched to the **pathsecondaryb** path regardless of its working condition.

```
device(config-router-mpls-lsp-samplelsp)# select-path unconditional pathsecondaryb
```

Syntax: [no] select-path [manual | unconditional] [*path-name* | primary]

The [**no**] option returns an LSP to the default auto select mode when it has been previously configured to the manual select mode or unconditional select mode.

The *path-name* variable is the name of the path that the user wants to assign manual select mode or unconditional select mode to. The user can optionally specify the primary path by using the **primary** keyword.

The **manual** option configures the specified path to operate in the manual select mode. When the user selects **primary** as the specified path with the **manual** option, the primary path is selected as the preferred path, which is the same as the default operation.

The **unconditional** option configures the specified path to operate in the unconditional select mode. When the user selects **primary** as the specified path with the **unconditional** option, the primary path is selected as the preferred path regardless of the condition of the primary path.

Configuration changes made to the select mode do not take effect for an already enabled LSP until the change is activated implicitly using the **commit** command or explicitly using a **reoptimize** command as described in Performing a commit for an LSP configuration command on page 120 or a system reboot is performed.

> **NOTE**
> When the user configures a primary path to be the selected path, a message is generated that states that it is already the default system behavior because the primary path is the default preferred path. In this instance, no configuration information is saved in the configuration file.

Configuring a Path Selection Revert Timer

The Path Selection Revert Timer feature provides an option to stabilize a path before traffic is switched to it. Without a configured Path Selection Revert Timer, the router switches between a primary and secondary path immediately after the current working path goes down. A problem with this mode of operation is that it can cause flapping when the current path goes up and down frequently. Also, the LSP to which the route is switching traffic might be unstable, which causes the router to fail back to the current LSP almost immediately.

The Path Revert Timer insures the stability of the LSP to which the traffic is switched by specifying the number of seconds that the LSP must be running before it actually carries traffic.

To configure a Path Selection Revert Timer for an LSP, use the **revert-timer** command in the LSP configuration context, as shown in the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp samplelsp
device(config-router-mpls-lsp-samplelsp)# revert-timer 10
```

Syntax: **[no] revert-time** *timer-value*

The *timer-value* value is the number of seconds that the router waits after the primary or selected path comes up before traffic reverts to that path. The range is 1- 65,535 seconds.

Usage considerations:

- The **revert-time** command has no effect on the unconditional select mode. Traffic is unconditionally switched to the user-selected path and stays on it.
- The path stability test used with the Revert Timer feature is based on the uptime of the latest instance of the path. This value can be different when the selected path has gone through a "make-before-break" procedure.
- For an LSP going through re-optimization, the new LSP does not carry traffic until the revert timer expires.
- When a user changes the revert timer, the basis of counting is the uptime of the path and is independent of the sequence or combination of configurations. Take, for example, a path that is configured in the manual select mode to be a secondary path with a revert-timer of 10 seconds. After the secondary path comes up, a 10-second timer starts, but after five seconds, the user changes the revert-timer value to four. Now the path has already been stable beyond the new configured revert-timer, so the original timer is canceled and traffic immediately switches over. However, if the user were to change the revert-timer value to eight seconds after running for five seconds, the existing count would terminate and start a new count of three seconds from the moment the first count terminated.

To configure a Path Selection Revert Timer, for an LSP, use the **revert-timer** command within the LSP configuration as shown in the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp samplelsp
device(config-router-mpls-lsp-samplelsp)# revert-timer 10
```

Syntax: **[no] revert-time** *timer-value*

The *timer-value* value specifies an amount of time in seconds that the router waits after the primary or selected path comes back up before reverting to it.

Setting a Class of Service value for the LSP

The 3-bit EXP field in the MPLS header can be used to define a *Class of Service (CoS)* value for packets traveling through the LSP. The user can manually set a CoS value for the LSP. The CoS value that the user sets is applied to the CoS (EXP) field in the MPLS header of all packets entering this LSP. This lets all packets traveling through an LSP to be treated with the same priority as they travel the MPLS domain. The user can assign the LSP a CoS in the range 0-7.

To assign a CoS value of 7 (highest priority) to all packets traveling through LSP tunnel 1.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp)# cos 7
```

Syntax: [no] cos *number*

The MPLS CoS value is used for determining priority within an MPLS domain only, so when the label is popped, the CoS value in the MPLS header is discarded; it is not copied back to the IP ToS field.

Allocating bandwidth to an LSP

The user can specify the allocation of bandwidth for an LSP, including the maximum and average rates for packets that travel over it. Allocating bandwidth to an LSP lets the LSRs determine how much bandwidth the LSP can consume and how much of the available bandwidth resources can be advertised.

The user can specify an average *mean-rate* kbps for the data on the LSP. When necessary, data can travel at *max-rate* Kbps, as long as the burst sent at the maximum rate contains no more than *max-burst* bytes.

To set the maximum rate of packets that can go through an LSP (in Kbps).

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# traffic-eng max-rate 20
```

Syntax: [no] traffic-eng max-rate *rate*

To set the average rate of packets that can go through an LSP (in Kbps).

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# traffic-eng mean-rate 10
```

Syntax: [no] traffic-eng mean-rate *rate*

To set the maximum size (in bytes) of the largest burst the LSP can send at the maximum rate.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# traffic-eng max-burst 10
```

Syntax: [no] traffic-eng max-burst *bytes*

Configuring a priority for a signaled LSP

The user can specify a priority for each signaled LSP for which this is the ingress LER. The priority determines the relative importance of the LSP during setup or preemption. The priority for an LSP has two components the setup priority and the hold priority.

When multiple LSPs are enabled at the same time, such as when the device is booted, LSPs that have a higher setup priority are enabled before LSPs that have a lower setup priority.

When an LSP is assigned a high setup priority, it may preempt an LSP that is already established, causing resources assigned to the lower priority LSP to be diverted to the higher priority LSP. The hold priority specifies how likely an established LSP is to give up its resources to another LSP. To be preempted, an LSP must have a lower hold priority than the preempting LSPs setup priority. In addition, an established LSP can be preempted by a higher priority LSP only if it would allow the higher priority LSP to be established successfully.

To configure LSP tunnel1 with a setup priority of 6 and hold priority of 1.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# priority 6 1
```

Syntax: **[no] priority** *setup-priority /hold-priority*

Possible values are 0 (highest priority) through 7 (lowest priority). An LSP setup priority must be lower than or equal to the hold priority. The default LSP setup priority is seven, and the hold priority is zero.

Assigning a metric to the LSP

The user can assign a metric to the LSP, which can be used by routing protocols to determine the relative preference among several LSPs towards a given destination.

To assign a metric of five to LSP tunnel1

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# metric 5
```

Syntax: **[no] metric** *number*

The metric has a range of 1 - 65535. By default, all LSPs have a metric of one. A lower metric is preferred over a higher one. When multiple LSPs have the same destination LSR, and they have the same metric, the traffic load is shared among them.

Including or excluding administrative groups from LSP calculations

Administrative groups, also known as resource classes or link colors, lets the user assign MPLS-enabled interfaces to various classes. When a device uses CSPF to calculate the path for an LSP, it takes into account the administrative group to which an interface belongs; the user can specify which administrative groups the device can include or exclude for this calculation.

For example, to include interfaces in either of the administrative groups "gold" and "silver" in the path calculations for LSP tunnel1, do the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# include-any gold silver
```

Syntax: **[no] include-any** *groups*

The value specified for *groups* can be one or more valid administrative group names or numbers. In this example, the device includes any of the interfaces that are members of groups "gold" or "silver" when calculating the path for this LSP. Only those interfaces in the "gold" or "silver" groups are considered for the LSP. Interfaces that are not part of these groups, as well as interfaces that are not part of any group, are eliminated from consideration.

To exclude interfaces in either administrative group "gold" or "silver" when the path for LSP tunnel1 is calculated.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# exclude-any gold silver
```

Syntax: **[no] exclude-any** *groups*

In this example, the device excludes any interface that is a member of group "gold" or "silver" when it calculates the path for this LSP. Only interfaces that are not part of either group are considered for the LSP.

To specify that an interface must be a member of both the "gold" or "silver" administrative groups in order to be included in the path calculations for LSP tunnel1.

```
device# configure
device(config)# router mpls
```

```
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# include-all gold silver
```

Syntax: [no] include-all *groups*

In this example, an interface must be a member of all the groups specified in the **include-all** command in order to be considered for the LSP. Any interface that is not a member of all the groups is eliminated from consideration.

Limiting the number of hops the LSP can traverse

By default, the path calculated by CSPF can consist of no more than 255 hops, including the ingress and egress LERs. The user can optionally change this maximum to a lower number.

For example, to limit CSPF to choosing a path consisting of no more than 20 hops for LSP tunnel1.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# hop-limit 20
```

Syntax: [no] hop-limit *number*

The range for the number of hops is 0 - 255.

Specifying a tie-breaker for selecting CSPF equal-cost paths

CSPF may calculate multiple, equal-cost paths to the egress LER. When this happens, the device chooses the path whose final node is the physical address of the destination interface. When more than one path fits this description, by default, the device chooses the path with the fewest hops. When multiple paths have this number of hops, the device chooses one of these paths at random. The user can optionally configure the device to choose the path that has either the highest available bandwidth or the lowest available bandwidth.

For example, the following commands cause CSPF to select the path with the highest available bandwidth when choosing among equal-cost paths calculated for LSP tunnel1.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# tie-breaking least-fill
```

Syntax: [no] tie-breaking least-fill | most-fill | random

The **least-fill** parameter causes CSPF to choose the path with the highest available bandwidth (that is, the path with the least utilized links).

The **most-fill** parameter causes CSPF to choose the path with the lowest available bandwidth (that is, the path with the most utilized links).

The **random** parameter causes CSPF to choose the path randomly from the equal-cost paths. This is the default.

Disabling the record route function

The RSVP *RECORD_ROUTE object (RRO)* allows an LSPs path to be recorded. An RRO consists of a series of sub-objects that can contain the addresses of the LSRs in the path. This information can be viewed with the **show mpls lsp detail** command. The path information is recorded in the RRO by default, but the user can disable path recording.

To disable path recording in the RRO.

```
device# configure
device(config)# router mpls
```

```
device(config-router-mpls)# lsp tunnel1
device(config-routermpls-lsp-tunnel1)# record disable
```

Syntax: record [disable | enable]

Disabling CSPF path calculations

By default, CSPF is enabled for signaled LSP calculations. When the device is the ingress LER for the LSP, it uses the information in the TED to help determine a path for the LSP.

To disable constraint-based path selection for LSP *tunnel1*.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# cspf disable
```

Syntax: cspf [disable | enable]

LDP tunnel metric

The user can use LDP tunnel metric information to decide whether LDP or RSVP is the preferred method for BGP nexthop resolution.

Configuration considerations

The user can change the tunnel metric configuration at any time. The new value applies only to tunnels that are brought up after the change. Metrics for existing tunnels do not change.

Configuring an LDP tunnel metric

To set all LDP tunnels to metric 2 (for example), enter the following command under the MPLS LDP configuration.

```
device(config-mpls-ldp)# tunnel-metric 2
```

The new metric is applied only to newly created LDP tunnels

Syntax: [no] tunnel-metric *metric-value*

The *metric-value* is a value from 1 through 65535. The default is zero (0).

To revert to the default value, enter the [**no**] version of this command.

To use the LDP tunnel metric for BGP nexthop resolution, enter the following command at the BGP configuration mode.

```
device(config-bgp)# next-hop-mpls compare-lsp-metric
```

Syntax: next-hop-mpls compare-lsp-metric

Displaying tunnel metric configuration settings

To display tunnel metric configuration settings, enter the **show mpls ldp** command.

```
device# show mpls ldp
Label Distribution Protocol version 1
  LSR ID: 10.1.1.1, using Loopback 1 (deleting it stops LDP)
  Hello interval: Link 5 sec, Targeted 17 sec
  Hold time value sent in Hellos: Link 15 sec, Targeted 41 sec
  Keepalive interval: 6 sec, Hold time multiple: 6 intervals
  Load sharing: 8
  Tunnel metric: 10
```

For additional information, see the CLI command **show mpls ldp** in the *MPLS Commands* chapter.

Configuring the maximum packet size

This feature allows the user to set a maximum IP packet size for packets that traverse an LSP without being fragmented. It can be configured for both primary and secondary paths.

To configure a maximum IP packet size for an LSP, enter commands such as the following.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# ipmtu 1500
```

Syntax: [no] ipmtu *packet-size*

The *packet-size* variable specifies the maximum packet size in bytes for IP packets transiting the LSP without being fragmented.

Enabling a signaled LSP

After the user sets the parameters for the signaled LSP, the user can enable it. Enabling the LSP causes the path to be set up and resources reserved on the LSRs in the LSPs primary path. Enabling the LSP is the final step in configuring it.

To enable LSP tunnel1:

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# enable
```

Syntax: [no] enable

Disabling an LSP

Disabling an LSP de-activates it, but does not remove the LSP from the device's configuration. (To remove the LSP from the device's configuration, use the **no lsp** *name* command.) To make changes to an active LSP, first disable the LSP, modify parameters on the LSP, and then enable the LSP.

To disable LSP *tunnel1*.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp tunnel1
device(config-router-mpls-lsp-tunnel1)# no enable
```

Syntax: [no] enable

FRR bypass LSPs

The LSP ping and traceroute facilities support FRR bypass LSPs. The user can ping or trace the protected LSP and bypass tunnel separately.

The user can ping or trace the ingress-originated or transit-originated bypass tunnel by specifying either the name of bypass LSP (as the user would any regular LSP name) or the entire RSVP session ID (including the tunnel endpoint, the tunnel ID, and the extended tunnel ID).

NOTE
In the current facility backup implementation, the bypass LSP name must be unique in the system (for example, the name cannot be the same as the regular LSP name).

The traceroute output of a backup tunnel depends on the setting of the **propagate-ttl** and **label-propagate-ttl** options. When both **propagate-ttl** and **label-propagate-ttl** options are turned on, the traceroute output shows the detail of the bypass path. When both options are turned off, the bypass path is shown as a single hop. The options are either both ON or both OFF.

To trace the route of a backup path, the TTL of the bypass and protected labels (when they are not implicit NULL labels) are set as in the following example:

- Both **propagate-ttl** and **label-propagate-ttl** are ON: TTL = 1, 2, 3, and so on, are set for both labels.
- Otherwise: bypass label TTL is set to 255. Protected label TTL is set to 1, 2, 3, and so on.

IP TTL is set to top most label TTL. Otherwise, it is set to 255.

Resetting LSPs

The **clear mpls lsp** command allows the user to reset an RSVP LSP session. Changes in the routing table after an LSP path is established do not take effect unless the LSP is brought down and then brought up again. After the user resets the LSP, it realigns to the new routing topology. The **clear mpls lsp** command can be used on the ingress LSR of the LSP.

Resetting normal LSPs

The **clear mpls lsp** command allows the user to reset normal LSPs. The user has the option of supplying the **primary | secondary** parameter for a normal LSP to reset only the primary/secondary path of the LSP.

When the user resets an LSP with the **clear mpls lsp** command, the following information message is displayed.

```
"Disconnecting signaled LSP name"
```

```
"Connecting signaled LSP name"
```

Syntax: clear mpls lsp *lsp-name* [**primary | secondary**]

When a **primary** or **secondary** optional keyword is not specified when the user resets a normal LSP, then both the **primary** and **secondary** LSP paths associated with the *lsp-name* is reset and restarted.

Resetting Bypass LSPs

This command allows the user to reset bypass LSPs.

NOTE
The **primary** or **secondary** optional keywords are not applicable for bypass LSPs.

Reset LSP considerations

The **clear mpls lsp** command resets and restarts an MPLS RSVP LSP.

NOTE
These commands are disruptive. Data traffic forwarding is impacted as the LSP is not in active state for sub-seconds after teardown. Resetting an LSP could trigger a series of actions depending upon the current state of the LSP.

The following describes the actions and state changes when an LSP is reset.

Resetting an LSP also resets the associated backup/detour LSPs:

- Resetting the primary path of an LSP causes the secondary LSP path to become active, when a hot-standby secondary path for the LSP is available. However, when the primary path comes up after the reset operation, the active path switches over from the secondary to the primary again. When the "revert-timer" is configured, the LSP path switchover may be dampened and obeys the usual revert-timer rule. There is no change in the revert-timer behavior due to the reset LSP feature.

    NOTE
    The above state changes are described here for informational purposes only. There could be several other intermediate state changes that are not listed here.

- Resetting the primary path of an adaptive LSP also resets the "other" new instances of the LSPs primary path, when available at the time of reset.
- Resetting the secondary path for an LSP resets the current secondary path of the LSP. It also resets the selected secondary path, when available at the time of reset.
- Resetting the secondary path for an LSP whose primary path is down may trigger the secondary path selection process to choose a new secondary path. When a new secondary path is found, it is signaled and may become the active path. When no secondary paths are found, then the current secondary may become the active path again after successful RSVP signaling.
- The primary path is UP but not active, and the secondary path is UP and active. The secondary to primary switchover occurred because the revert-timer has been configured (using a large value). Resetting the secondary LSP path still forces the path switchover from secondary to primary path in spite of the revert-timer configuration.
- For an adaptive LSP, when reset is performed before the **commit** command, then the LSP is reset and comes-up with a new set of configuration parameters. However, this is disruptive for data traffic, unlike the **commit** command, because the current instance of the LSP is reset while there is no new instance of the LSP available (because the **commit** command has not been executed yet).

Generating traps and syslogs for LSPs

Multi-Service IronWare software supports the ability to enable and disable SNMP traps and syslogs for LSPs. LSP traps and syslogs are enabled by default.

To enable LSP traps after they have been disabled, enter the following command.

```
device(config)# snmp-server enable traps mpls lsp
```

Syntax: [no] snmp-server enable traps mpls lsp

Use the [no] form of the command to disable LSP traps.

To enable LSP syslogs after they have been disabled, enter the following command.

```
device(config)# log enable mpls lsp
```

Syntax: [no] log enable mpls lsp

Use the [no] form of the command to disable the syslog for LSPs.

Ingress Fast FRR Convergence

Ingress Fast Re-route (FRR) Convergence

Ingress Fast Re-route (FRR) convergence improves the failover time for ingress LSPs.

Ingress FRR convergence accomplishes the following:

1. The LP keeps the Detour and Bypass information up-front, so there is no delay in downloading the information from the LP to program the hardware for re-route.

2. Instead of sending multiple re-route messages for all FRR LSPs belonging to MPLS interface, which is going down, a small IPC message indicating "interface is going/went down" to the LP is sent, reducing multiple Inter- Process Communications (IPCs) and data processing on both the LP and the MP.

3. Aggregation of multiple re-route messages into one or more switchover IPCs to LP, reducing the IPC processing on both the MP and the LP. The number of LSPs in a single switchover IPC may be limited to an optimal number, which the LP can process without starving other LP tasks.

Download backup XC information to LP in advance

The LP must keep all the information required to make the switchover happen immediately in case of a re-route of the tunnel. This helps to minimize the switchover time by avoiding information download from the Management Pack (MP) when required. To achieve this, FRR information for the tunnel is also downloaded to the LP and kept available ahead of time in LP.

Messages are made capable of accommodating both the protected and associated backup and detour information together.

While adding a protected information, the detour and backup information may be absent, but while adding detour and backup information, the protected information must be present so that LP can make a mapping from protected to backup and detour. The message is capable of updating and deleting individual information explicitly. An assumption is that backup information is never received at Router Label Database Forwarder (RLDF) without protected and is being configured in RLDF and LP.

Interface status IPC to LP to indicate the MPLS if down

To avoid overhead of so many messages and their processing including the delay for switchover of data path, one IPC message "MPLS_IPC_SUBMSG_MPLS_INTERFACE_STATUS" is sent to the LP, indicating if the interface has transitioned to an Up or Down state. Though in Ingress FRR convergence, it indicates only the transition from the Up to Down state by status field in the message set to *"false"*. On receipt of this message, the LP takes required action on the all the tunnels associated with that interface. The LP is expected to re-route the protected tunnels and ignores the messages for the unprotected tunnels. The message now contains enough information, which is identified to be the interface index of the egress interface, to identify the interface in LP.

At MP, all the tunnels which are supposed to be impacted by this message are marked as their re-route is communicated to the LP. In case of receipt of re-route XC message for those tunnels from Label Manager for whom re-route is already being sent to LP, the request is ignored as they are already re-routed in the LP.

Switchover due to bandwidth reduction on the interface

When the bandwidth of the MPLS interface goes down, the RLDF receives a bandwidth update on the interface; this reduction in bandwidth triggers the preemption of tunnels on the LAG. The bandwidth is updated for the interface in RLDF and sends IPS to the RSVP. Later the RSVP initiates re-route for FRR capable tunnels and sends the IPS to the RLDF by way of the label manager. The RLDF later sends the IPC to the LPs one per tunnel.

To avoid delay in sending re-route for impacted tunnels due to decrease of bandwidth in the interface, instead waiting for the IPS from the Label Manger and then sending an IPC message for each re-route request, it is better to send one message containing greater or equal to one reroute message to the LP, reducing CPU load. Instead of sending all the information about re-route, "MPLS_IPC_SUBMSG_MPLS_SWITCHOVER_LSPS" IPC message is sent to the LP, having just enough information to find out the tunnels or LSPs to be re-routed for greater or equal to one tunnel or LSPs. IPC message contains the number of tunnels or LSPs to be re-routed as a result of this message and an array of information having details about the re-route. The maximum number of switch over messages in a single IPC is decided for optimal performance. A Dy-Sync component is used to send an IPC to the LP. The Dy-Sync buffer is flushed to the LP as soon as the optimal number being agreed is reached upon, or the buffer gets full or done with the processing of all of the impacted tunnels or LSPs. The MP does not send a switchover request for tunnels or LSPs that do not have backup information.

All the tunnels which are supposed to be impacted by this message are marked as their re-route is communicated to the LP so that a duplicate switch over message is not being sent to the LP. A switchover message for any tunnel is sent to the LP only in case it is a protected one and backup or detour information is already updated to the LP including protected is active. If this condition is not met, the tunnel is ignored for re-route. In case of receipt of re-route, XC message for those tunnels from the Label Manager for whom re-route is already being sent to LP, the request is ignored as they are already rerouted in LP.

Regular re-route from the label manager

All re-route messages for tunnels received from the Label Manager as a part of normal RSVP processing of different messages are sent to LP as and when they are received in RLDF without being aggregated provided new re-route being communicated flag in the RLDF_LSP_CB is not set to TRUE otherwise re-route message is discarded in RLDF.

Inherit FRR LSPs bandwidth for backup path

Glossary

Terms used for inherit FRR LSPs bandwidth for backup path.

Term	Meaning
LSP	Label Switched Path
MPLS	Multiprotocol Label Switching
RSVP	Resource ReserVation Protocol
RESV	Reserve (RSVP control packet for Reservation of resources)
TE	Traffic Engineering
FRR	Fast Re-route
BI	Ingress Facility FRR LSP backup
DI	Ingress one-to-one Detour FRR LSP backup

Introduction

Allows the bandwidth configuration of the Primary protected LSP to be used for signaling their FRR backup LSPs.

This feature supports both 'Facility FRR LSPs' and 'One-to-one Detour LSPs'. This feature also includes an option to treat the bandwidth requested by backup paths to be either a strict requirement or a loose requirement.

Specifications

FRR LSPs can request bandwidth on their backup paths as well. The 'FAST_REROUTE' object and the 'SESSION_ATTRIBUTE' object for RSVP are used to control the backup for a protected LSP. This specifies various attributes to be used for backup path. This object is inserted into the PATH message by the head-end LER (ingress).

The FAST_REROUTE object has the following format:

```
Class-Num = 205
C-Type = 1
          0               1               2               3
          +-------------+-------------+-------------+-------------+
          |      Length (bytes)       | Class-Num   |   C-Type    |
          +-------------+-------------+-------------+-------------+
          | Setup Prio  | Hold Prio   | Hop-limit   |    Flags    |
          +-------------+-------------+-------------+-------------+
          |                    Bandwidth                         |
          +-------------+-------------+-------------+-------------+
          |                    Include-any                       |
          +-------------+-------------+-------------+-------------+
          |                    Exclude-any                       |
          +-------------+-------------+-------------+-------------+
          |                    Include-all                       |
          +-------------+-------------+-------------+-------------+
```

'SESSION_ATTRIBUTE' object included in the PATH message includes 'SESSION_ATTRIBUTE Flags'. The 'Bandwidth Protection Desired' flag in this is used for bandwidth requests over the backup.

The SESSION_ATTRIBUTE object has the following format:

SESSION_ATTRIBUTE without resource affinities:

```
SESSION_ATTRIBUTE class = 207
LSP_TUNNEL C-Type = 7

   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Setup Prio  | Holding Prio  |     Flags     | Name Length  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                             |
   //          Session Name      (NULL padded display string)   //
   |                                                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SESSION_ATTRIBUTE without resource affinities:

```
SESSION_ATTRIBUTE class = 207
LSP_TUNNEL_RA C-Type = 1
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Exclude-any                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Include-any                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Include-all                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Setup Prio  | Holding Prio  |     Flags     | Name Length  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                             |
   //          Session Name      (NULL padded display string)   //
   |                                                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Flags:

Bandwidth protection desired flag: 0x08

This flag indicates to the PLRs along the protected LSP path that a backup path with a bandwidth guarantee is desired. The bandwidth to be guaranteed is that of the protected LSP, if no FAST_REROUTE object is included in the PATH message. If a FAST_REROUTE object is present in the PATH message, then the bandwidth specified therein is to be guaranteed.

To report whether bandwidth is provided as requested, a flag is defined in the RRO IPv4/IPv6 sub-object flags.

RRO IPv4 Sub-object format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |     Length    | IPv4 address (4 bytes)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv4 address (continued)      | Prefix Length |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
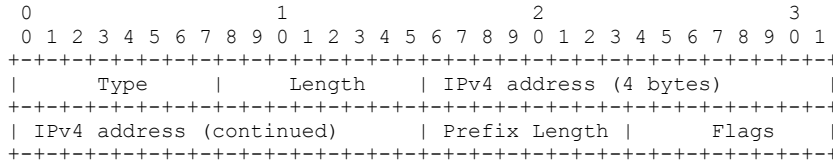
RRO IPv6 Sub-object format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |     Length    | IPv6 address (16 bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)       | Prefix Length |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
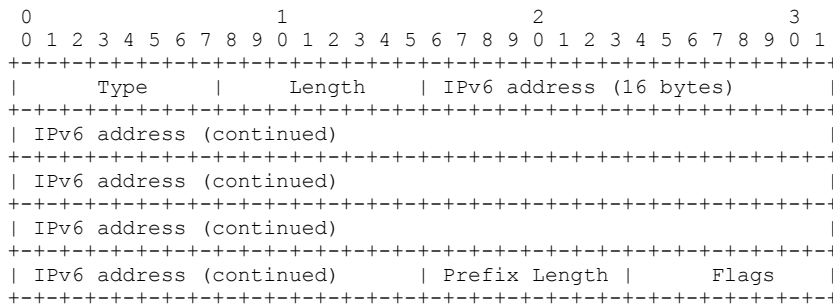
Flags:

Bandwidth protection: 0x04

The PLR will set this bit when the protected LSP has a backup path that is guaranteed to provide the desired bandwidth that is specified in the FAST_REROUTE object or the bandwidth of the protected LSP, if no FAST_REROUTE object was included. The PLR sets this whenever the desired bandwidth is guaranteed. The PLR sets this flag when the desired bandwidth is guaranteed and the "bandwidth protection desired" flag was set in the SESSION_ATTRIBUTE object; if the requested bandwidth is not guaranteed, the PLR does not set this flag.

Requirements

The following are the requirements for this feature:

1. A command to allow bandwidth inheritance from protected LSPs to backup LSPs (at ingress router only).

2. A command to treat bandwidth requirement by backups as 'strict' versus 'loose'.

Customer configurations

When a guarantee of bandwidth protection is desired, then the "bandwidth protection desired" flag in the SESSION_ATTRIBUTE object is set; otherwise, this flag is cleared. A PLR considers an LSP to have asked for local protection if the "local protection desired" flag is set in the SESSION_ATTRIBUTE object and/or the FAST_REROUTE object is included. If the "bandwidth protection desired" flag is set, the PLR tries to provide a bandwidth guarantee. If this is not feasible, then the PLR

- Either tries to provide a backup without a guarantee of bandwidth in "best-effort" mode. OR
- Does not provide a backup at all in case of "Guarantee" mode.

The following treatment for the RRO IPv4 or IPv6 sub-object's 'bandwidth protection flag' are followed if an RRO is included in the protected LSP's RESV message.

- The PLR sets the "bandwidth protection" flag in RRO if the backup path offers a bandwidth guarante. If the path does not, the PLR clears the "bandwidth protection" flag.

The provisioning of bandwidth for the backup is based solely on the local configuration of the router over which the backup LSP is passing. The 'Guarantee' or 'Best-effort' configuration of the PLR router cannot be transferred to the other routers over which the backup path is passing.

PLR treats bandwidth requirement as "Guarantee"

When the global configuration on this router is set to 'Guarantee', all the backup paths signaling on this router treats the bandwidth requirement by all the LSPs requesting backups with bandwidth as a strict requirement. The backup successfully sets up only when the bandwidth requirement is satisfied. When the requested bandwidth is not available, then the backup path for the LSP is not setup.

PLR treats bandwidth requirement by backups as 'best-effort'

When the global configuration on this router is set to 'best-effort', then all the backup paths signaling on this router treats the bandwidth requirement by all the LSPs requesting backups with bandwidth as a loose requirement. The PLR router tries to setup the backup with bandwidth. When a backup with bandwidth is available, the "bandwidth protection" flag in RRO is set and the backup is signaled with bandwidth. When a backup path without the bandwidth is available, the "bandwidth protection" flag in RRO is cleared and the backup is signaled without bandwidth.

In short, the following table briefly describes the behavior of the PLR router with respect to backup paths availability:

Bandwidth	Guarantee	Best-effort
Available	Local Protection + BW Protection	Local Protection + BW Protection
Unavailable	Backup not available	Local Protection

RRO Flags:

P = Local Protection Available

B = Bandwidth Protection Guaranteed

Effect of changing the treatment of backup bandwidth

On changing the global configuration of treating the bandwidth requested by the backup path from 'Guarantee' to 'Best-effort' or vice versa:

- Existing backup paths are not affected and stay as is.
- All newly signaled backup paths after the configuration change accept the newly applied rule to setup their backup paths.

Inherit bandwidth from protected LSP for signaling backup

A new configuration command is included under the (config-mpls-lsp-frr) context. The 'bandwidth inherit' command inherits the bandwidth of the protected LSP path to its backup. Only one of the following two can be configured at a time on the FRR backup LSP:

- **bandwidth** *dec*
- **bandwidth inherit**

Configuring one will automatically overwrite the other configuration (if any). When the 'bandwidth inherit' configuration is present in the LSP FRR configuration, the FAST_REROUTE object in PATH message carries the bandwidth of the protected LSP so that any router trying to provide protection for this LSP tries to provide bandwidth protection.

Changing the bandwidth inheritance configuration on the fly

The **bandwidth inherit** command, for an adaptive LSP, can be configured or unconfigured on the fly when LSP is enabled. This configuration acts like any other adaptive LSP configuration that can be changed on the fly. A **commit** command confirms the modified configuration.

When un-configuring this inheritance command, if there is no specific bandwidth configuration for the backup LSP, then the newly signaled backup is signaled with a ZERO bandwidth requirement.

Link protection for FRR

A *Label Switched Path (LSP)* set up across an MPLS network is used to switch traffic across MPLS network. The path used by a LSP across the network is based upon network resources or any other traffic engineering constraints provided by the user. Based on TE-constraints, the ingress MPLS router computes the path to be taken by LSP and signals it using RSVP protocol.

By nature, nodes and links in a MPLS networks are prone to failure. It is likely that the link or the nodes through which LSP is traversing can fail. In the event of a failure of a node or link, RSVP protocol has mechanisms that informs the ingress node about the failure the to ingress node. On receipt of failure message for LSP across the path, the ingress router re-signals the LSP using a new path.

Due to messaging and other network delays, the ingress router cannot respond fast enough to minimalize the loss of traffic. Traffic is lost from the moment the failure occurs and until the new path is setup for the LSP, which is quite large in quantum for service provider networks.

In order to avoid loss of traffic, *Fast Reroute (FRR)* protects the LSP and allows a broken LSP to be repaired immediately at the point of failure. Point of failure is termed as *"Point of local repair" (PLR)*, where the LSP can be repaired locally without intimating or waiting for the ingress router. PLR is the MPLS router which detects the failure and redirects the traffic appropriately to its backup path with minimal loss.

Typically at the PLR, two type of protection can be provided to LSP:

Link Protection: In this protection, the backup is selected in such a way that it avoids the failed link which was used earlier by the LSP. Traffic merges back to main stream from the backup on the very next MPLS router. Refer to following Link protection for FRR illustrating link protection provided at R2 to LSP ingressing from R1 to R4.

FIGURE 22 Link protection
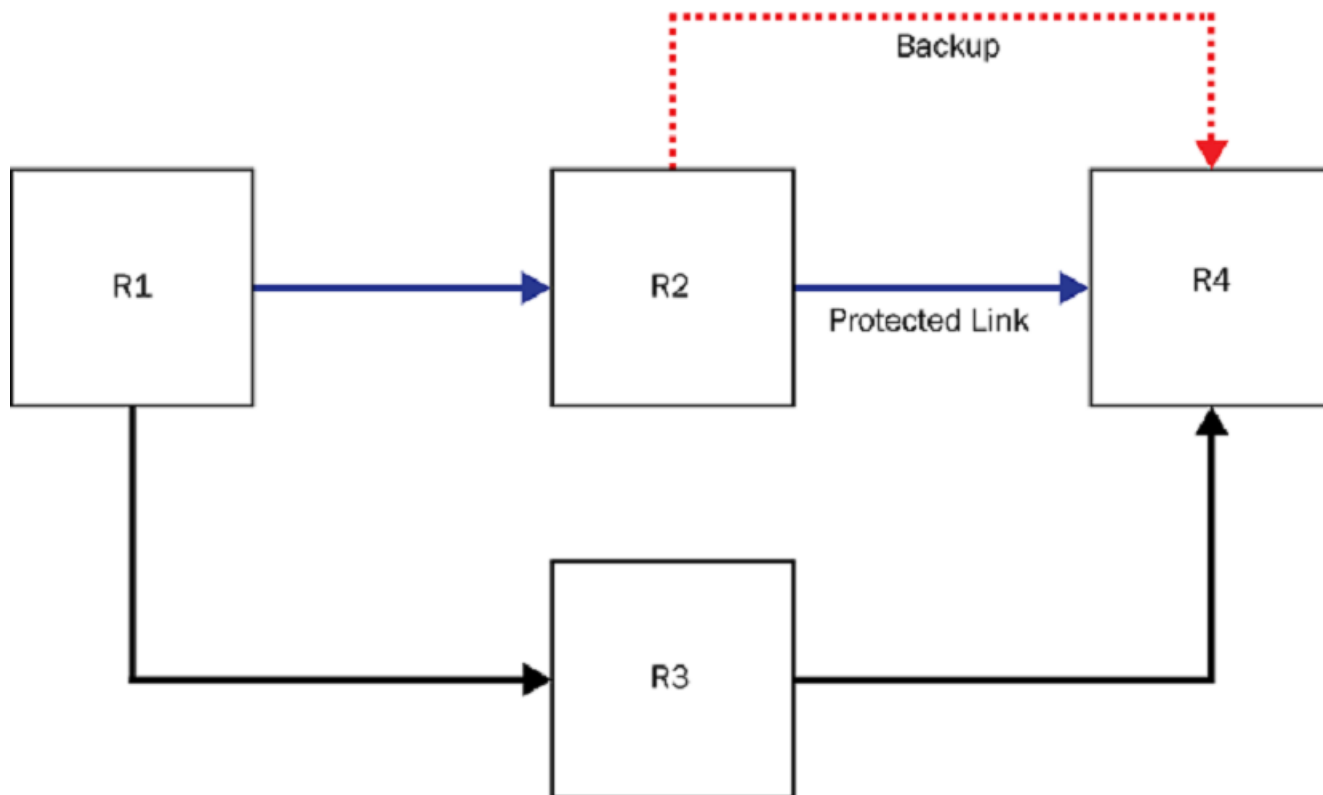
Node Protection: In this protection, backup is selected in such a way that it avoids the failed link along with router to which this link connects to. The node which was responsible for link failure, is avoided altogether in its entirety, which was used earlier by the LSP. Traffic merges back to main stream from backup on somewhere downstream from the node, which is being avoided. Refer to Link protection for FRR illustrating node protection provided at R1 to LSP ingressing from R1 to R4.

FIGURE 23 Node protection
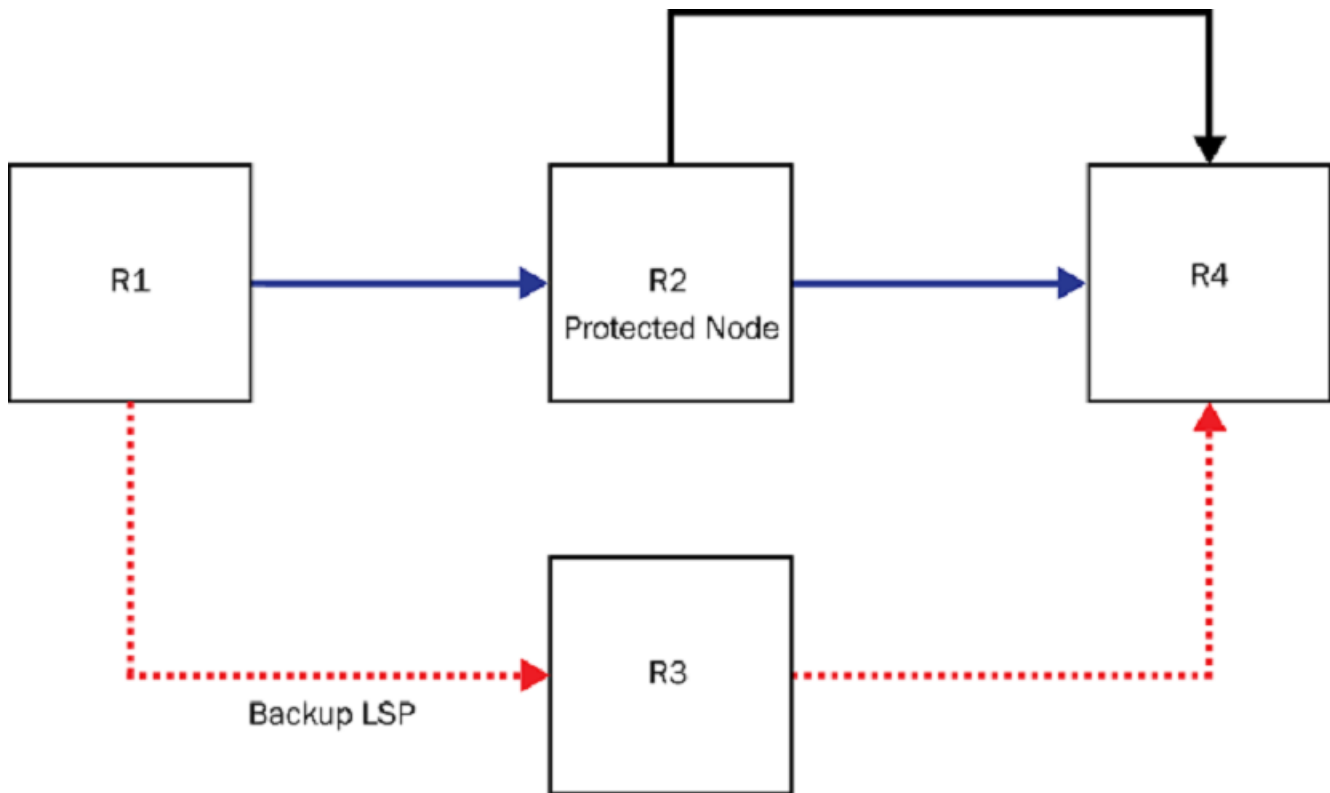
As part of this feature, ingress routers are allowed to expose this property of MPLS RSVP LSP to you and lets the user choose or prefer between Link protection or Node protection. Once the Node protection is chosen, PLR first tries to establish a backup LSP which provides Node protection. When Node Protection is not possible, it attempts to fall back to Link protection.

When the user chooses link protection over node protection, this is communicated to all routers participating in LSP. Each PLR in this case limits its search for backup LSP which provides link protection. In cases where link protection cannot be offered, PLR falls back to node protection.

The above feature is applicable to both one to one protection and many to one FRR protection.

The feature provides options to the user to set preferential method requested for local protection. When RSVP LSP is enabled with FRR (local protection), the user would be able to configure either Link protection or Node protection. Node protection remains the default.

Configuration steps for adaptive and non-adaptive LSPs have inherent differences with respect to their make-before break capabilities. The default behavior for both type of LSPs remains node protection.

Configuring protection type preference for Non-Adaptive LSPs

You are able to change protection type preference (Node protection to Link protection or vice versa) only on admin down state of a non-adaptive LSPs. Any non-adaptive LSP which is already enabled by the user for signaling, cannot be changed.

Configuring protection type preference for Adaptive LSPs

Because adaptive LSPs TE-property can be changed without restarting LSP and changed values takes effect through the make-before-break process, you are allowed to change the protection type preference (Node protection to Link protection or vice versa) at any point of time during life cycle of an adaptive LSPs, irrespective of its administrative or operational state. When you change the preferential protection type and it commits to the configuration, configuration takes effect. Signaling of the changed property depends upon state of LSP. For example, when the admin is UP or DOWN, it is operationally UP or DOWN. There is no change in the MBB trigger because of this feature. All MBB aspects including, but not limited to, implicit and explicit commits remain unchanged.

TABLE 5 Protection type preferences for adaptive LSPs

	Requesting Node Protection		Requesting Link Protection	
	Earlier Request: Link protection.	Earlier Request: Node protection.	Earlier Request: Link protection.	Earlier Request: Node protection.
Adaptive LSP	LSP requests node protection on next commit operation.	No change	No change	LSP requests link protection on next commit operation.
Non-Adaptive disabled LSP	LSP requests node protection once user enables LSP.	No change	No change	LSP requests link protection once user enables LSP.

> **NOTE**
> If you try to configure the feature on a non-adaptive enabled LSP, the following error is displayed:

```
Error: Must disable lsp before changing parameters
```

Configuring an adaptive LSP

The Multi-Service IronWare software supports Adaptive LSPs. Using this feature, the user can change the following parameters of an LSP while it is in the enabled state:

- cspf
- exclude-any
- hop-limit
- include-all
- include-any
- primary-path
- priority
- tie-breaking
- traffic-eng

When one of these parameters is changed on a Adaptive LSP, a new instance of the same LSP is signaled using the newly defined parameters. Once the new LSP comes up, traffic is moved to the new LSP instance and the old LSP instance is torn down.

To configure an LSP named to20 as an Adaptive LSP, use the following commands.

```
device(config)# router mpls
device(config-mpls)# lsp to20
device(config-mpls-lsp-to20)# adaptive
```

Syntax: [no] adaptive

Once an LSP is configured to be adaptive, it can have the parameters described above changed. In the following example, the Setup and hold priorities for adaptive LSP to20 are changed to seven and one.

```
device(config-mpls)# lsp to20
device(config-mpls-lsp-to20)# priority 7 1
```

The new parameters are not changed for the adaptive LSP until the **commit** command is issued for the LSP.

> **NOTE**
> Once the **commit** command has been issued, there may be a 30 ms traffic
> disruption.

In the following example of the **show mpls lsp** command for lsp to20, the priorities are not changed in the output.

```
device(config-mpls-lsp-to212)# show mpls lsp to212
LSP to212, to 10.5.1.1
 From: 10.4.1.1, admin: UP, status: UP, tunnel interface: tnl1
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0  Adaptive
  Maximum retries: 0, no. of retries: 0
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  OTHER INSTANCE PRIMARY: NEW_INSTANCE admin: DOWN, status: DOWN
  Maximum retries: 0, no. of retries: 0
  Setup priority: 7, hold priority: 1
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl1, outbound interface: e1/2
   Tunnel index: 4, Tunnel instance: 1 outbound label: 3
   Path calculated using constraint-based routing: yes
   Explicit path hop count: 1
    10.2.1.2 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.2.1.2
```

The following **commit** command makes the new parameter settings active in Adaptive LSP to20's configuration

```
device(config-mpls)# lsp to20
device(config-mpls-lsp-to20)# commit
```

Syntax: [no] commit

After the commit command runs, the user can see that the priorities have changed by using the **show mpls lsp** command for lsp to20.

```
device(config-mpls-lsp-to212)# show mpls lsp to212
LSP to212, to 10.5.1.1
  From: 10.4.1.1, admin: UP, status: UP, tunnel interface: tnl1
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0  Adaptive
  Maximum retries: 0, no. of retries: 0
  Setup priority: 7, hold priority: 1
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl1, outbound interface: e1/2
   Tunnel index: 4, Tunnel instance: 2 outbound label: 3
   Path calculated using constraint-based routing: yes
   Explicit path hop count: 1
    10.2.1.2 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.2.1.2
```

Re-optimizing LSPs

Under ordinary conditions, an LSP path does not change unless the path becomes inoperable. Consequently, the router needs to be directed to consider configuration changes made to an LSP and to optimize the LSP path based on those changes. This is accomplished using the **mpls reoptimize** command as shown in the following.

```
device# mpls reoptimize lsp to20
```

Syntax: [no] mpls reoptimize all | lsp *lsp-name*

The **all** option directs the router to re-optimize the paths for all LSPs configured.

The **lsp** option directs the router to re-optimize the path for the LSP specified by the *lsp-name* .

> NOTE
> On re-optimization of an adaptive LSP, LSP accounting statistics might miss the accounting of some of the packets.

Time-triggered re-optimizing

The user can set a timer to optimize a specific LSP path on a periodic basis. Upon expiration of this timer, the LSP is optimized for a new path when the new path has a lower cost than the existing path. This timer can be configured when the LSP is in a disabled state, and the timer value can be adaptively changed when the LSP is in an enabled state by issuing a **commit** to take effect. Until a **commit** is issued the re-opt timer is disabled.

To set the LSP re-optimization timer, use the **reoptimize_timer** command during LSP configuration, as the following shows.

```
device(config)# router mpls
device(config-mpls)# lsp to20
device(config-mpls-lsp-to20)# reoptimize_timer 1000
```

Syntax: [no] reoptimize_timer *seconds*

The *seconds* variable specifies the number of seconds from the beginning of one re-optimization attempt to the beginning of the next attempt. The range of values for *seconds* is 300 - 65535.

The [**no**] option can be used to disable a timer that has been configured. By default, a timer is not configured.

Configuring a re-optimization timer does not interfere with running the manual **reoptimize** command as described in Re-optimizing LSPs on page 143.

> NOTE
> When upgrading software, configured adaptive LSPs are initialized with no re-optimization
> timer.

> NOTE
> This feature does not apply to LSPs within a FRR network.

Static transit LSP

Static Transit LSP allows you to configure a transit cross-connect, consisting of the inbound-label, an optional outbound-label, and a next-hop IPv4 address. The next-hop configured will be used to get the outbound-interface and the interface next-hop address, and send the packet.

Configuring Static Transit LSP

To configure Static Transit LSP, perform the following procedure:

1. Perform label range splitting (optional)
2. Configure static transit LSP

Label range splitting (optional)

> NOTE
> This procedure requires a reload.

Use the following configuration procedure to split the label-ranges.

1. Configure the static range with the start and end of the range using the **label-range static min-value** *min* **max-value** *max* command. The dynamic range will start from the next label-value after the end of the static range.
2. Save the configuration and reload.

   The label range takes effect when the router comes up.

   > NOTE
   > Perform this procedure only if you do not want to use the default range values.

   The default values for static LSP is the range of 16 - 2047.

Static transit LSP configuration

Use the following procedure to configure a static transit LSP.

1. Create the LSP and name using the **static-lsp transit** *name* command.
2. Configure the inbound-label that will be received in the packets from upstream.
3. Configure an outbound-label if it needs to do a swap operation. If it is a next-to-last hop doing PHP, this will be optional; the default implicit-null label value of three is assumed.
4. Configure the next-hop address for the packet using the **next-hop** *x.x.x.x* command.
5. Enable the LSP.

Configuration example

Configuration is required at the transit node.

Splitting label space into static and dynamic

```
device# configure
device(config)# router mpls
device(config-router-mpls)# label-range static min 16 max 4095
```

Static LSP configuration at transit

```
device# configure
device(config)# router mpls
device(config-router-mpls)# static-lsp transit t1
device(config-router-mpls-static-transit-lsp-t1)# in-label 16
device(config-router-mpls-static-transit-lsp-t1)# next-hop 10.10.10.2
```

```
device(config-router-mpls-static-transit-lsp-t1)# out-label 17
device(config-router-mpls-static-transit-lsp-t1)# enable
```

Functional Considerations

The following configuration behaviors must be considered before the configuration.

Changes in label range configuration

1. Configuration of in-label values outside of the label range will not be allowed.
2. If the label range is increased and reloaded, you will get a wider label range to use. Refer to Label range splitting (optional) on page 144.

Enable with no in-label or next-hop

If you attempt to enable a static LSP which does not have either the inbound-label or next-hop configured, the enable will not be allowed. Inbound-label and next-hop are mandatory configurations.

Configuring MPLS Fast Reroute using one-to-one backup

To configure MPLS Fast Reroute by using the one-to-one backup method for a defined LSP named **frr_tunnel,** use the **ffr** command as in the following example.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp frr_tunnel
device(config-router-mpls-lsp-frr_tunnel)# to 10.1.1.1
device(config-router-mpls-lsp-frr_tunnel)# primary-path direct_path
device(config-router-mpls-lsp-frr_tunnel)# secondary-path alt_path
device(config-router-mpls-lsp-frr_tunnel)# frr
device(config-router-mpls-lsp-frr_tunnel-frr)# bandwidth 100
device(config-router-mpls-lsp-frr_tunnel-frr)# hop-limit 20
device(config-router-mpls-lsp-frr_tunnel-frr)#
```

Syntax: **[no] frr**

This command enables MPLS Fast Reroute using the one-to-one backup on the LSP under whose configuration it is enabled. Options for this command are described in the sections that follow.

MPLS Fast Reroute using one-to-one backup configuration options

The following options can be set for a MPLS Fast Reroute using one-to-one backup configuration:

- Bandwidth
- Exclude any
- Hop limit
- Include all
- Include any
- Priority

Configuring bandwidth for a MPLS Fast Reroute

To define a bandwidth constraint for the Fast Reroute path, use the following command.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp frr_tunnel
device(config-router-mpls-lsp-frr_tunnel)# frr
device(config-router-mpls-lsp-frr_tunnel-frr)# bandwidth 100
```

Syntax: [no] bandwidth *rate*

The *rate* variable specifies the bandwidth in Kbps for the bypass route.

Acceptable value can be between zero (0) and two (2) Gbps.

A value of zero (0) means that the detour route uses a best-effort value for bandwidth.

The default value is zero (0).

Configuring a hop limit for a MPLS Fast Reroute

By default, a detour route can consist of no more than 255 hops. the user can optionally change this maximum to a lower number.

For example, to limit any detour route in the LSP named **frr_tunnel,** to no more than 20 hops.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp[ frr_tunnel
device(config-router-mpls-lsp-frr_tunnel)# frr
device(config-router-mpls-lsp-frr_tunnel-frr)# hop-limit 20
```

Syntax: [no] hop-limit *number*

The number of hops can be from 0 - 255.

Configuring priority for a MPLS Fast Reroute

The user can specify setup and hold priorities for the detour routes within a specified LSP. These priorities are available to any LSP and function exactly the same on standard LSPs as they do on detour LSPs. The priority determines the relative importance of the detour routes during setup or preemption. The priority has two components the setup priority and the hold priority.

When a detour LSP is assigned a higher setup priority, it can preempt any LSP (detour or otherwise) that is already established and has a lower holding priority, causing resources assigned to the lower priority LSP to be diverted to the higher priority LSP. The hold priority specifies how likely an established LSP is to give up its resources to another LSP. To be preempted, an LSP must have a lower hold priority than the preempting LSPs setup priority. In addition, an established LSP can be preempted by a higher priority LSP only when it would allow the higher priority LSP to be established successfully.

To configure the detour routes of LSP frr_tunnel with a setup priority of six and hold priority of one.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp frr_tunnel
device(config-router-mpls-lsp-frr_tunnel)# frr
device(config-router-mpls-lsp-frr_tunnel-frr)# priority 6 1
```

Syntax: [no] priority [*setup-priority* |hold-priority]

Possible values are zero (highest priority) through seven (lowest priority). A setup priority must be lower than or equal to the configured hold priority on an LSP. By default, the setup priority is seven and the hold priority is zero.

Including or excluding administrative groups from LSP calculations

Administrative groups, also known as resource classes or link colors, allows the user to assign MPLS-enabled interfaces to various classes. When a device calculates the path for a detour LSP, it takes into account the administrative group to which a interface belongs; the user can specify which administrative groups the device can include or exclude when making its calculation.

For example, to include interfaces in either administrative group "gold" or "silver" in the path calculations for detour routes of the LSP frr_tunnel.

```
device(config-mpls-lsp-frr_tunnel)# frr
device(config-mpls-lsp-frr_tunnel-frr)# include-any gold silver
```

Syntax: [no] include-any *groups*

The value specified for *groups* can be one or more valid administrative group names or numbers. In this example, the device includes any of the interfaces that are members of groups "gold" or "silver" when calculating detour routes for this LSP. Only those interfaces in the "gold" or "silver" groups are considered for the detour routes. Interfaces that are not part of these groups, as well as interfaces that are not part of any group, are eliminated from consideration.

To exclude interfaces in either administrative group "gold" or "silver" when detour routes for LSP frr_tunnel are calculated.

```
device(config-mpls-lsp-frr_tunnel)# frr
device(config-mpls-lsp-frr_tunnel-frr)# exclude-any gold silver
```

Syntax: [no] exclude-any *groups*

In this example, the device excludes any of the interfaces that are members of groups "gold" or "silver" when calculating detour routes for this LSP. Only interfaces that are not part of either group can be considered for the detour routes.

To specify that an interface must be a member of both the "gold" or "silver" administrative groups in order to be included in the detour routes for LSP frr_tunnel.

```
device(config-mpls-lsp-frr_tunnel)# frr
device(config-mpls-lsp-frr_tunnel-frr)# include-all gold silver
```

Syntax: [no] include-all *groups*

In this example, an interface must be a member of all the groups specified in the **include-all** command to be considered in a detour route for the LSP. Any interface that is not a member of all the groups is eliminated from consideration.

Protecting MPLS LSPs through a bypass LSP

Implementing a bypass LSP to back up one or more MPLS LSPs requires the following tasks:

- The LSPs that the user intends to have the protection of a bypass LSP must be enabled for Fast Reroute and then must be specified as needing facility backup. (The user does not need to create the LSP before the bypass LSP is created because the bypass LSP identifies the LSPs to protect by interface IDs, not by LSP names.)
- An LSP is configured to be a bypass LSP with enough bandwidth for all the LSPs that it protects.
- The interfaces that get the protection of a bypass LSP are identified to that particular LSP. Protected LSPs can be identified by individual interfaces, ranges of interfaces, interface groups, or a LAG.

    NOTE
    The name of the bypass LSP must be unique among all bypass LSPs and all protected LSPs.

The sections that follow describe the items unique to the bypass LSP feature. The common LSP parameters are described elsewhere throughout this chapter.

Specifying an LSP to request facility backup

LSP xmr3-199 is configured for Fast Reroute and then configured to request facility backup.

```
device# configure
device(config)# router mpls
device(config-router-mpls)# lsp xmr3-199
device(config-router-mpls-xmr3-199)# frr
device(config-router-mpls-xmr3-199-frr)# facility-backup
```

Syntax: [no] facility-backup name

A subsequent iteration of the **show** command in the bypass LSP context shows that this LSP is a candidate for protection by a bypass LSP. The display for protected LSP xmr3-199 shows that, under **frr** , the facility-backup line shows this protection is requested.

```
device(config-router-mpls-bypasslsp-123)# show mpls config lsp xmr3-199
lsp xmr3-199
to 10.33.33.33
primary xmr3-100
priority 4 3
secondary xmr3-101
   standby
frr
   facility-backup
revert-timer 10
enable
```

Syntax: show mpls configuration lsp *name*

Specifying a bypass LSP

The user can create a bypass LSP by using the **bypass-lsp** command. Thereafter, in the bypass LSP context, the user must specify at least one interface as an exclude (protected) interface. This interface can be on a LAG. In this example, xm4 is specified to be a bypass LSP; the protected LSP interfaces are specified; and then the options for a bypass LSP are displayed.

```
device(config)# router mpls
device(config-mpls)# bypass-lsp xm4-by
device(config-mpls-bypasslsp-xm4-by)#
device(config-mpls-bypasslsp-xm4-by)# exclude-interface e 1/2, e 1/2, e 2/5-e 2/8
device(config-mpls-bypasslsp-xm4-by)# ?
clear             Clear table/statistics/keys
cos               Class of service
disable           Tear down the LSP
enable            Establish the LSP
end               End Configuration level and go to Privileged level
exclude-any       Exclude any of the administrative groups
exclude-interface Choose the interface to avoid as well as protect
exit              Exit current level
from              Set ingress router of the LSP
hop-limit         Limit of hops the LSP can traverse
include-all       Include all of the administrative groups
include-any       Include any of the administrative groups
metric            Set the LSP metric
no                Undo/disable commands
primary-path      Set primary explicit path
priority          Setup/hold priorities
quit              Exit to User level
record            Enable or disable recording path routes
show              Display system information
tie-breaking      Choose the tie breaking mode for cspf
to                Set egress router of the LSP
traffic-eng       Set traffic engineering parameters
write             Write running configuration to flash or terminal
```

Syntax: [no] bypass-lsp *name*

The *name* must be unique among all regular LSPs and bypass LSPs.

Syntax: [no] **exclude-interface** *linkid* , *linkid-begin-linkid-end*

Syntax: [no] **exclude-any** *group*

Configuring a bypass LSP to be adaptive

The user can configure a bypass LSP to be adaptive using the **adaptive** command. The user can further configure an adaptive bypass LSP as follows:

- Control when the bypass LSP path gets re-optimized either by manually causing immediate path re-optimization using the **mpls reoptimize** command, or by setting a timer to re-optimize the path periodically using the **reoptimize-timer** command.
- Change parameters for an enabled bypass LSP without having to disable it.

A new instance of the LSP is signaled following path re-optimization or a change in LSP parameters and the old path is released.

> **NOTE**
> Unlike regular adaptive LSPs, new path signaling does not take place on an adaptive bypass LSP that has traffic flowing through it.

Specifying a bypass LSP to be adaptive

To specify an LSP to be adaptive, use the **adaptive** command in the bypass LSP context. By default, bypass LSPs are not adaptive. The LSP must be in the disabled state before the user enters the **adaptive** command. When the user has specified the LSP to be adaptive, the user can enable the LSP. In this example, the xm4-by LSP is configured to be adaptive.

```
device(config)# router mpls
device(config-mpls)# bypass-lsp xm4-by
device(config-mpls-bypasslsp-xm4-by)# adaptive
device(config-mpls-bypasslsp-xm4-by)# enable
```

Syntax: [no] **adaptive**

Reoptimizing a bypass LSP

Under ordinary conditions, an LSP path does not change unless the path becomes inoperable. Consequently, the device needs to be directed to consider configuration changes made to an LSP and to optimize the LSP path based on those changes. As with regular LSPs, one way to do this for a bypass LSP is to enter the **mpls reoptimize** command as shown in the following example.

```
device# mpls reoptimize lsp xm4-by
```

Syntax: **mpls reoptimize all** | **lsp** *lsp-name*

The **all** option directs the router to re-optimize the paths for all LSPs configured, including bypass LSPs.

The **lsp** *lsp-name* option directs the router to re-optimize the path for the specified LSP.

The *lsp-name* variable specifies the LSP to be optimized. This LSP can be a regular LSP or a bypass LSP.

Time-triggered re-optimizing a bypass LSP

As with regular LSPs, the user can set a timer to optimize a specific bypass LSP path on a periodic basis. By default, the timer is disabled. Upon expiration of this timer, the bypass LSP is optimized for a new path when the new path has a lower cost than the existing path.

The user can configure the re-optimization timer when the bypass LSP is in either the disabled state or the enabled state. When configured with the bypass LSP in the disabled state, the new timer value takes effect immediately. When configured with the bypass LSP in the enabled state, the new timer value takes effect when the user performs an explicit commit operation by entering the **commit** command; until the user enters the **commit** command, the timer expires according to its previous setting.

When the bypass LSP is carrying traffic when the re-optimization timer expires, the path is not re-optimized, and the bypass LSP is evaluated for optimization the next time the timer expires.

To set the re-optimization timer for a bypass LSP, use the **reoptimize-timer** command in the bypass LSP context.

```
device(config)# router mpls
device(config-mpls)# bypass-lsp xm4-by
device(config-mpls-bypasslsp-xm4-by)# reoptimize-timer 1000
device(config-mpls-bypasslsp-xm4-by)# commit
```

Syntax: **[no] reoptimize-timer** *seconds*

The *seconds* variable specifies the number of seconds from the beginning of one re-optimization attempt to the beginning of the next attempt. The range of values for *seconds* is 300 through 65535. The [**no**] option can be used to disable a timer that has been configured.

Modifying parameters on an enabled bypass LSP

When an adaptive bypass LSP is enabled, the user can change the following parameters:

- exclude-any
- exclude-interface
- hop-limit
- include-all
- include-any
- primary-path
- priority
- reoptimize-timer
- tie-breaking
- traffic-eng

    NOTE
    For a bypass LSP, the user cannot change the CSPF parameter.

To change the value of one of these parameters, enter the command by the same name in the bypass LSP context, and then enter the **commit** command. The following example changes the limit on the number of hops a bypass LSP can traverse.

```
device(config)# router mpls
device(config-mpls)# bypass-lsp xm4-by
device(config-mpls-bypasslsp-xm4-by)# hop-limit 20
device(config-mpls-bypasslsp-xm4-by)# commit
```

For descriptions of all LSP parameters and the syntax of the commands that set them, refer to Configuring signaled LSP parameters on page 119. For bypass LSPs however, the user must execute the commands in the bypass LSP context.

After entering the **commit** command, a new bypass LSP is signaled and includes the changes. However, considerations apply depending on whether the enabled adaptive bypass LSP is currently protecting any LSPs, and if so, whether it is actively carrying traffic. When the adaptive bypass LSP is not currently protecting any LSP, no additional considerations exist on configuring LSP parameters.

When the adaptive bypass LSP is carrying traffic from a locally repaired LSP, then the signaling of the new LSP instance is delayed until the bypass LSP is no longer actively backing up any LSP.

When the adaptive bypass LSP is protecting LSPs, some of those protected LSPs might become unprotected by this bypass LSP when the user changes any of the following parameters:

- exclude-any
- include-any
- include-all
- traffic-eng max-rate
- traffic-eng mean-rate

For example, when the traffic-eng mean-rate is decreased, one of the following actions takes place:

- The bypass LSP reroutes, which affects whether the backup is still valid on that bypass LSP.
- The bandwidth of the bypass LSP is reduced, which does not affect the path, but it does affect any backup that is reserving bandwidth on the bypass LSP. In this case, the protected LSPs are evaluated, and backups are removed from the bypass LSP until a consistent state is achieved.

The following example changes the traffic-eng mean-rate for a bypass LSP.

```
device(config)# router mpls
device(config-mpls)# bypass-lsp xm4-by
device(config-mpls-bypasslsp-xm4-by)# traffic-eng mean-rate 2000
device(config-mpls-bypasslsp-xm4-by)# commit
```

Syntax: commit

Dynamic Bypass LSPs

When the user configures node protection or link protection on a device, bypass LSPs are created to the next-hop or next-next-hop routers for the LSPs traversing the device. Multiple protected LSPs use the same bypass LSP in case of protected LSP link or node failures.

There are two ways to establish a bypass LSP:

1. **Static:** The user manually configures in a MPLS enabled network so the protected LSPs uses the bypass LSP for link or node protection.
2. **Dynamic:** Computes and establishes a bypass LSP at runtime when there is a requirement to provide a FRR link or node protection to a facility protected LSP at its PLR points. Dynamic bypass LSPs are these bypass LSPs. A protected LSP requiring backup-LSP protection at every PLR triggers a dynamic bypass path computation and setup.

Bypass LSP terminology

Table 6 dynamic bypass LSPs terms.

TABLE 6 Common terms associated with dynamic bypass LSPs

Term	Meaning
LSP	Label Switched Path
MP	Merge Point
NHOP	Next Hop

TABLE 6 Common terms associated with dynamic bypass LSPs (continued)

Term	Meaning
NNHOP	Next Next Hop
NNNHOP	Next Next Next Hop
PLR	Point of Local Repair
MMB	Make-Before-Break

Figure 24 illustrates a protected dynamic bypass interface.

FIGURE 24 Protected dynamic bypass interface
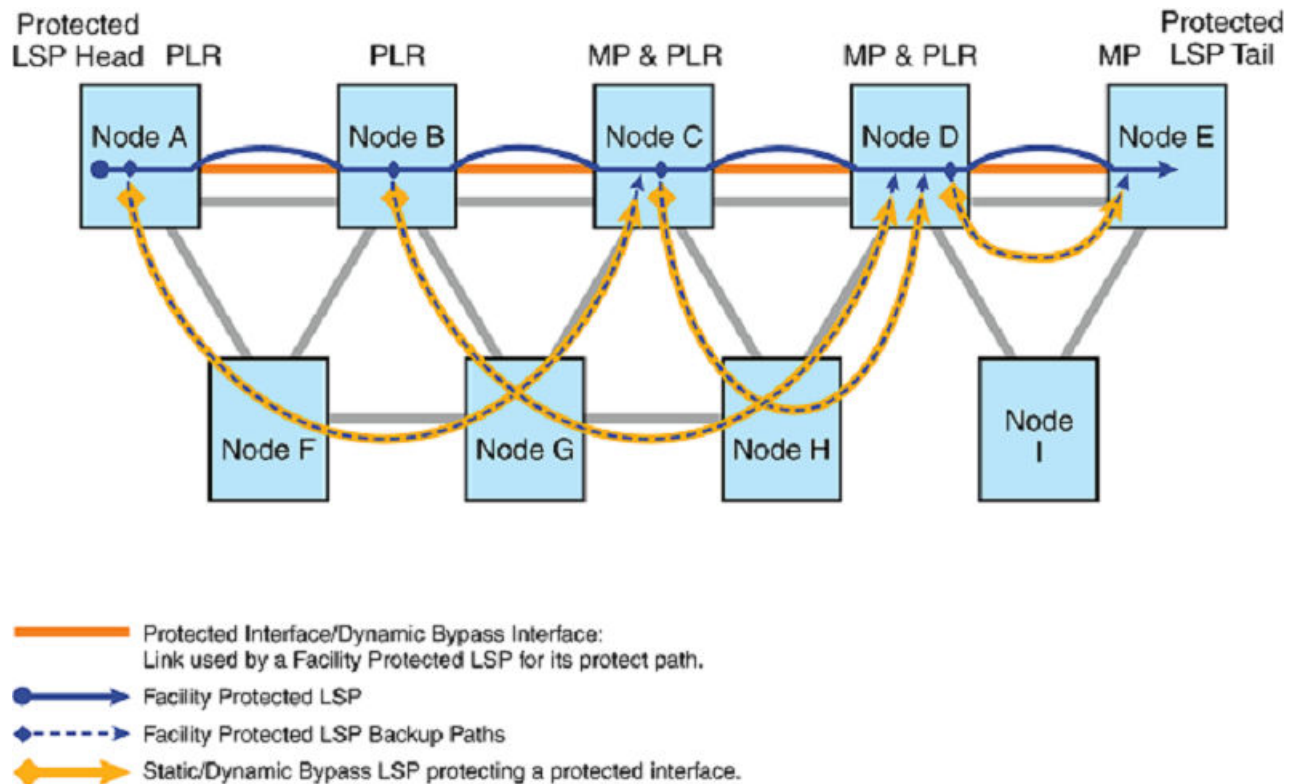
Facility protected LSPs use Bypass LSP for their FRR backup paths. Multiple backup paths can share a common bypass LSP provided their source and destination are same.

With Dynamic bypass feature, a PLR of a Facility Protected LSP shall automatically create a bypass LSP (if required to do so) such that the backup path can ride on it.

An automatically created bypass LSP can provide Link protection or Node protection based on its destination node, which is a merge point for the protected LSP.

A failure in protected interface will switch the protected path traffic to backup path which is riding on a bypass LSP.

When establishing a facility protected LSP with link or node protection, each LSR on the primary path verifies when there are any existing bypass LSPs that require protection constraints. When finding a bypass, it updates its bandwidth, depending on the requesting backup path bandwidth. The protected LSP backup path uses this bypass to reach its merge point. When there is no bypass available, the LSR computes and establishes a new bypass LSP, addressing the backup path constraints.

When an LSR is created for an interface, any number of facility protected LSPs may reuse or share a bypass LSP. All of the protected LSPs must use the same protected interface and the bypass LSP must satisfy the new LSPs backup path constraints. A periodic optimization of dynamic bypass LSPs is performed using the make-before-break procedure.

Configuration parameters, such as bandwidth, hop-limit, and priority are set when creating the dynamic bypass LSP. The system makes use of these parameters when creating a new dynamic bypass LSP. Modifications on these parameters are taken into consideration during the next cycle of re-optimization or can be manually initiated at the interface level re-optimization.

Bandwidth of the newly triggered bypass LSP is zero by default, unless it has an explicit configuration. When a new facility protected LSP requests a bandwidth which cannot be accommodated within an existing dynamic bypass LSP, there is no automatic make-before-break for the existing dynamic bypass LSP. Instead, a new dynamic bypass is created, depending on the configurations and system limits.

A link which is protected by the bypass LSP is called a protected-link, protected interface, or an excluded interface. Multiple facility protected LSPs use a common downstream link which becomes a protected link for all of them. This signifies that there is at least one dynamic bypass LSP to the node connected by the interface (NHOP) and many NNHOP dynamic bypass LSPs based on the path taken by the facility protected LSP. Similarly, there are bypass LSPs to several different NNNHOP nodes. Bypass LSP terminology shown below is an example:

FIGURE 25 Two facility protected LSPs using a common link

LSP1 from A to E
LSP2 from A to G
NHOP Bypass for LSP1 and LSP2 - Link protection
NNHOP Bypass for LSP1 - Node protection
NNHOP Bypass for LSP2 - Node protection
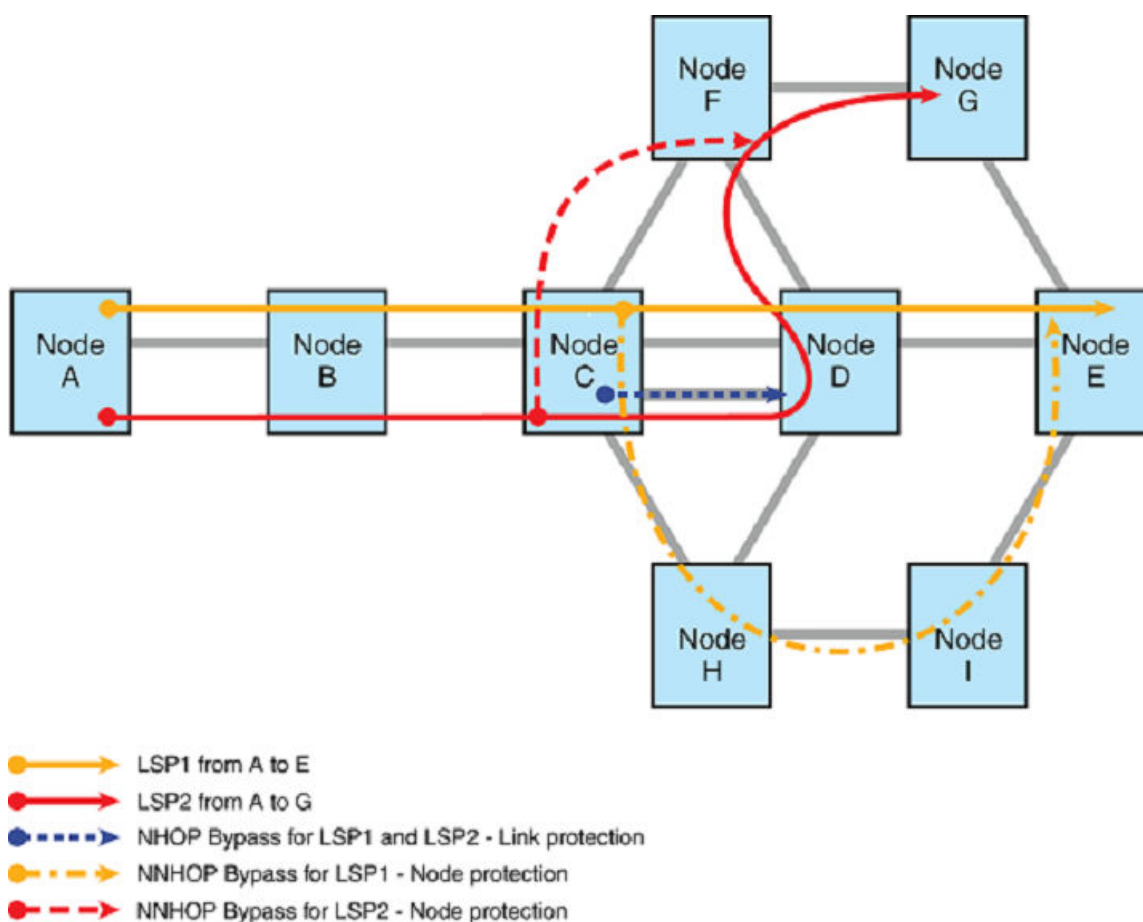
Two facility Protected LSPs use a common link between Node C and D.
For this protected interface there can be one common NHOP Bypass,
two different NNHOP Bypasses.

Configuration considerations

- Dynamic bypass LSP modification (*make-before-break (MBB)*) is not supported except for a case of re-optimization. When a new protected LSP requests more bandwidth, no automatic MBB is performed for the existing dynamic bypass LSP. Re-optimization makes use of make-before-break, to modify the LSP, so it can create the new instance of the dynamic bypass LSP. The newly created dynamic bypass LSP uses the latest attributes from the global and interface level dynamic bypass configurations. Re-optimization is carried out as to a re-optimization timer expiry.

- Dynamic bypass optimization with modified bandwidth is performed using MBB during the optimization process (timer expiry or user initiated). No detour or backup re-optimization to minimize dynamic bypass LSPs is completed in this release. Here, the 'detour/backup re-optimization' does not refer to LSP path re-optimization; it refers to the realignment of backup paths on the bypass LSPs so there is a minimum number of bypasses with optimal use of their bandwidths.

- One dynamic bypass is created and ten backups occupy this dynamic bypass.

- A second dynamic bypass is created and the 11th backup occupies this dynamic bypass.

- After some time, one backup from the first dynamic bypass goes away, which leave one megabyte per second unoccupied in the first dynamic bypass.

- Now the first dynamic bypass has nine megabits per second and the second dynamic bypass has one megabyte per second occupied and there are 10 backup paths. Look at this at a bandwidth use optimization point of view; all 10 backups are accommodated in only one dynamic bypass LSP. Having an algorithm which realigns the backups so there is only one dynamic bypass, instead of two bypasses is not supported in this release.

- All dynamic bypass creation or reuse by a protected path depends on the protected LSP backup path request triggers. This means a backup path established over a dynamic bypass depends on a protected path request and retries.

- Node or link protection attributes are signaled through the session attributes of the protected LSP. Based on the protected path requested backup protection type, node or link protection dynamic bypasses are created.

- Deciding on a possible merge point from a PLR on an existing dynamic bypass LSP depends on the existing backup path re-setup mechanism. A failure in the existing dynamic bypass LSP leads to a new backup retry from a protected LSP PLR and is considered as new backup path setup sequence.

- Dynamic bypass LSP functionality as a bypass LSP is by way of an existing bypass LSP. All timer driven or user initiated re-optimization MBB is the same as the existing bypass LSP MBB.

- Dynamic bypass LSP modification (make-before-break) is not allowed, except for a case of re-optimization. When a new protected LSP requests more bandwidth, no automatic MBB is performed for the existing dynamic bypass LSP. Re-optimization makes use of make-before-break, to modify the LSP, so it can create the new instance of the dynamic bypass LSP. The newly created dynamic bypass LSP uses the latest attributes from the global and interface level dynamic bypass configurations. Re-optimization is carried out upon a re-optimization timer expiry.

- The total number of bypass LSPs that are created on a device is limited to 1000. This is due to an existing limitation relating to number and content of next-hop entries. This limit of 1000 is applicable for the cumulative total of both the static and dynamic bypass LSPs. Therefore, the maximum number of dynamic bypass LSPs created on a system is always equal to (1000 - (current number of configured static bypass LSPs)).

- Dynamic bypass optimization with a modified bandwidth is performed using MBB during the optimization process (timer expiry or user initiated). No detour or backup re-optimization to minimize dynamic bypass LSPs is completed in this release.

- Dynamic bypass interface mean bandwidth validation with MPLS interface maximum or reservable bandwidth is not completed because MPLS interface bandwidth changes dynamically based on configurations like LAG.

Creating a dynamic bypass LSP

The dynamic bypass LSP feature is activated by enabling the dynamic bypass at the global level under the router MPLS and dynamic-bypass.

Dynamic bypass LSP creation is controlled by a combination of control at global level and the interface level. A MPLS interface with dynamic bypass enabled is able to create dynamic bypass LSP to provide backup path for protected LSP traversing out of this interface. Refer to Figure 26 for creating a dynamic bypass LSP.

FIGURE 26 Creating a dynamic bypass LSP

A PLR tries to set Dynamic Bypass LSP with destination as the MP and MP selection will be in the order of NNHOP, NHOP, NNNHOP, NNNNHOP...
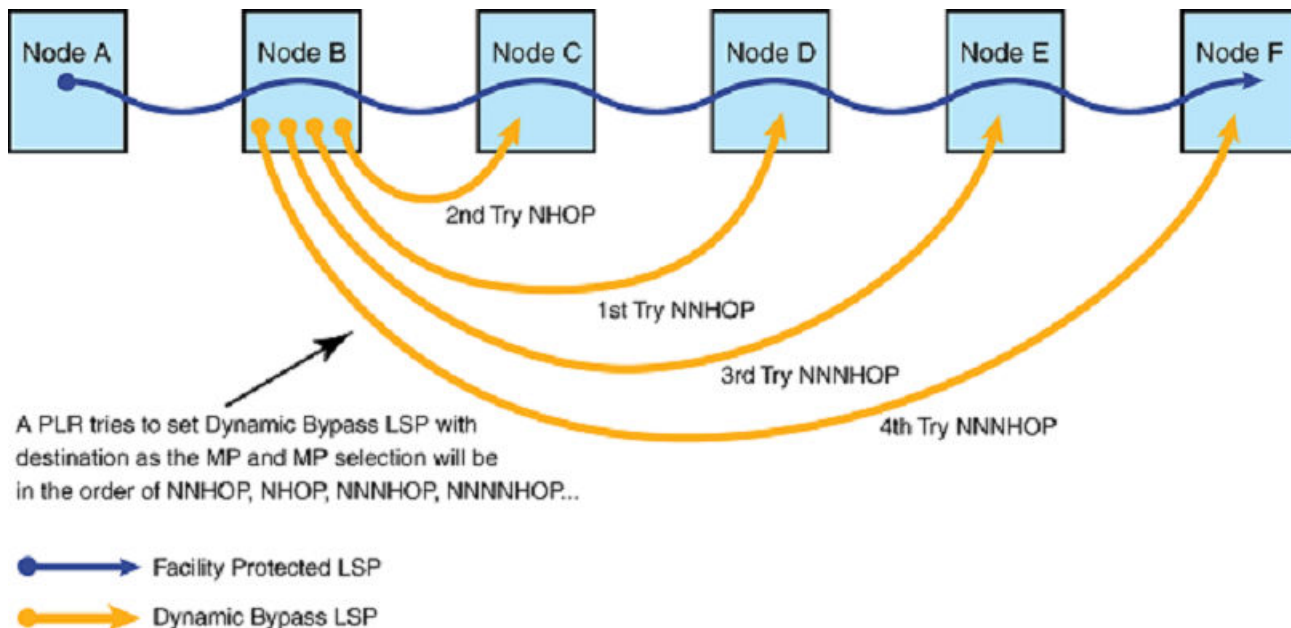
Facility protected LSP PLR will originate Dynamic Bypass LSP with Merge Point as destination.

Merge point selection for dynamic bypass creation use the existing order of backup path Merge Point selection. If node protection is not requested, reverse the order of steps 1 and 2.

1. Tries to create a Dynamic bypass LSP to NNHOP Merge Point
2. If step 1 fails, then tries to create a Dynamic bypass LSP to NHOP Merge Point
3. If step 2 fails, then tries to create a Dynamic bypass LSP to NNNHOP Merge Point
4. If step 3 fails, then tries to create a Dynamic bypass LSP to NNNNHOP Merge Point
5. and so on, till the Protected LSP destination node is Merge Point

Dynamic bypass LSP creation must meet the following criteria:

- There are no existing static bypass or dynamic bypass LSPs to satisfy the facility protected LSP backup path request.
- Dynamic bypass is allowed to be created under current configuration for the protected interface.
- Dynamic bypass creation does not exceed the configured or default system limits under current state.
- There is a path available to setup the dynamic bypass LSP to fulfill backup request constraints.
- Facility protected LSP PLR originate the dynamic bypass LSP with a merge point as the destination.

- Merge point selection for dynamic bypass creation uses the existing order of backup path merge point selection. When node protection is not requested, reverse the order of step 1 and 2.

Steps:

1. Tries to create a dynamic bypass LSP to NNHOP merge point.
2. When step 1 fails, then attempts to create a dynamic bypass LSP to NHOP merge point.
3. When step 2 fails, then attempts to create a dynamic bypass LSP to NNNHOP merge point.
4. When step 3 fails, then attempts to create a dynamic bypass LSP to NNNNHOP merge point.
5. This continues until the protected LSP destination node is merge point.

Configuration steps

Any modifications to the dynamic bypass interface or the router mode configuration parameters are applied to the new creation of dynamic bypass LSPs.

Dynamic bypass parameter changes made at the interface level only apply to the existing dynamic bypass LSPs protecting this interface, when triggered by events such as timer or user intervention.

The configurable parameters include, but are not limited to: bandwidth, hop-limit, priority, cos, adaptiveness, and primary path. These apply to all dynamic bypass LSPs created to protect this interface.

The dynamic bypass feature configurations steps are as follows:

1. Enable dynamic bypass on MPLS router mode.
2. Set global dynamic bypass configurable parameters. This step is optional.
3. To enable a dynamic bypass on all the MPLS interfaces without going to each individual interface, use the **enable-all-interfaces** command in the global mode. Otherwise, go to next step to enable dynamic bypass on individual MPLS interfaces and customize the way in which dynamic bypass get created for a protected interface. The user can also override the **enable-all-interfaces** commands effect on individual MPLS interfaces by configuring the dynamic bypass in those interfaces explicitly as in the next steps. This step is optional.
4. Enable a dynamic bypass on one or more MPLS interfaces. This step is optional when using step 3.
5. Set interface level dynamic bypass configurable parameters. This step is optional.

Configuring the dynamic bypass LSP

A link protecting bypass has its source as the protected link's near end node and the destination as the protected link's far end node. The link protecting the bypass does not require it to pass through the directly connected link from source to destination.

A node that protects the bypass may have any downstream node (of protected LSP) as its destination node, with the exception of the protected link's far end node.

Network diagram

A simple topology for dynamic bypass LSP illustration is as shown in Figure 27 below. A protected LSP with facility backup FRR enabled is setup from node A to G passing through node B, C and D. LSP uses link 1 between node B and node C. Node B is one of the PLR for the protected LSP.

FIGURE 27 Dynamically created bypass LSP
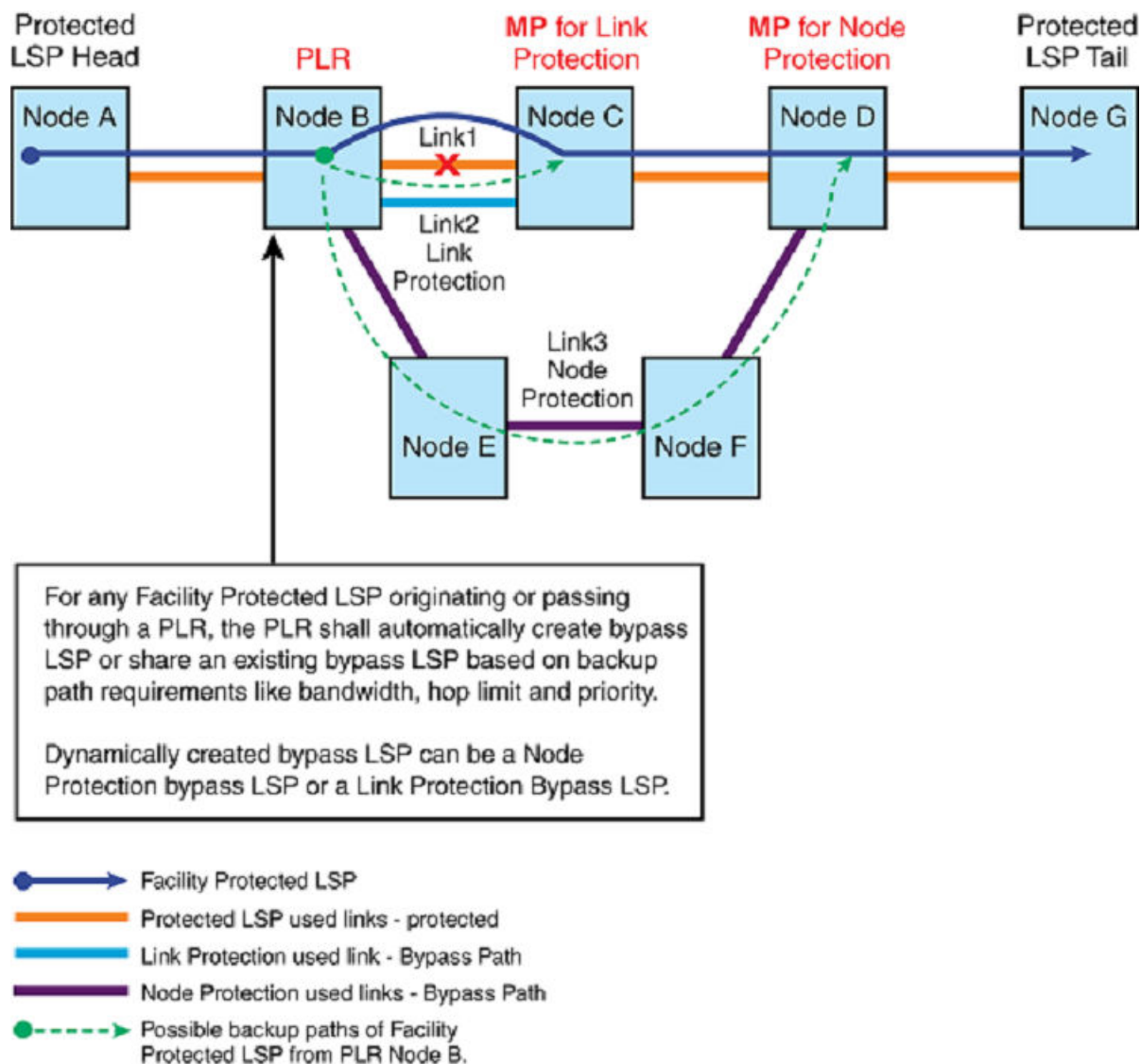
Globally enabling dynamic bypass

Using the **dynamic-bypass** command in MPLS router configuration mode for the first time enables the dynamic bypass feature in the system. When using the **dynamic-bypass** command in the MPLS router configuration mode which is already configured, there is no change in the existing status (enabled or disabled) of global dynamic bypass. To enable the dynamic bypass on MPLS router mode, enter a commands such as the following.

```
device(config-mpls)# dynamic-bypass
device(config-mpls-dynamic-bypass)# enable
```

Syntax: [no] dynamic-bypass

Syntax: [no] enable

The [no] form of the command disables the feature on the router.

Enabling all interfaces

Use the **enable-all-interfaces** command to enable dynamic bypass on all MPLS interfaces on a router. This is applicable to all MPLS interfaces where the user has not configured dynamic bypass manually. To set the optional global dynamic bypass configuration, enter a command such as the following.

```
device(config-mpls-dynamic-bypass)# enable-all-interfaces
device(config-mpls-dynamic-bypass)#
```

Syntax: **[no] enable-all-interfaces**

Use the **[no]** form of the command inside global dynamic-bypass configuration mode to disable dynamic bypass on all existing MPLS interfaces.

Setting the maximum number of dynamic bypass LSPs

The maximum number of dynamic bypass LSP is configurable in global mode. This is the limit for the total number of dynamic bypass LSPs that can be created on a router. This number must be less than, or equal to, the global maximum number of bypass LSPs that can be configured on a router. The maximum number of bypass LSPs supported on a device is currently limited to 1000. The means that the maximum number of dynamic bypass LSP that can be configured on a system is always (1000 – (current number of configured Bypass LSPs)). When the max-bypasses limit is changed to a value which is less than current active number of dynamic bypasses, the limit is changed to the new value and this limit is considered for next new creations. Existing exceeding number dynamic bypasses are not deleted. To enable dynamic bypass on one or more MPLS interfaces, enter a command such as the following.

```
device(config-mpls-dynamic-bypass)# max-bypasses 150
device(config-mpls-dynamic-bypass)#
```

Syntax: **[no] max-bypasses** *number*

The *number* parameter is the maximum number of dynamic bypasses that can be created on the system.

Use the **[no]** form of the command to returns the settings to the default value.

Setting the maximum number of dynamic bypass LSPs per MP

Use the **max-bypasses-per-mp** command to set the maximum number of dynamic bypass LSP configurable on a MPLS interface mode. This global value is taken as the interface mode maximum bypasses per MP default value. This is the limit for the total number of dynamic bypass LSPs created to a merge point corresponding to a protected interface. A PLR may have 'M' number of merge points with respect to a protected LSPs. There may be 'N' number of protected LSPs riding on an interface with dynamic bypass enabled. Max-bypasses configurations limits the maximum number of dynamic bypass LSPs to each merge point. When the **max-bypasses-per-mp** limit changes to a value which is less than the current active number of dynamic bypasses, the limit changes to the new value and this limit is considered for the next creations. Existing dynamic bypasses exceeding the new limit do not delete. When the **max-bypasses-per-mp** limit changes to a value which is more than the system max-bypasses limit, this creates a warning message.

Use the **max-bypasses-per-mp** command to change the maximum number of dynamic bypass LSP on an MPLS interface mode.

```
device(config-mpls-dynamic-bypass)# max-bypasses-per-mp 8
device(config-mpls-dynamic-bypass)#
```

Syntax: **[no] max-bypasses-per-mp** *number*

Use the *number* parameter to set the maximum number of dynamic bypass that may be created to a merge point from a PLR. The range of valid values is from 1 to 1000. The default value is the same as the max-bypasses parameter value.

Use the **[no]** form of the command to return the settings to the default values.

Setting the re-optimizer-timer

When the re-optimization value is set to a non-zero value and the timer sets the amount of seconds, the **reoptimizer-timer** command enables the dynamic bypass LSP re-optimization. The re-optimization timer value is configurable on all MPLS interface modes. The global set value is applicable to all dynamic bypass LSPs for which corresponding interface level re-optimization timer value is not set.

```
device(config-mpls-dynamic-bypass)# reoptimize-timer 300
device(config-mpls-dynamic-bypass)#
```

Syntax: [no] **reoptimizer-timer** *number*

Use the *number* parameter to set the re-optimization timer value in seconds for the dynamic bypass LSPs. The range of valid values is from 300 to 65535. The default value is zero, which means re-optimization is disabled.

Use the [**no**] form of the command to return the settings to the default value.

Enabling dynamic bypass per interface

Use the **enable-all-interfaces** command to configure a dynamic bypass as enabled on a MPLS interface. There is no change in the dynamic bypass configured state (enabled or disabled), when already configured on the interface. When the user configures a dynamic bypass on a MPLS interface using this command, this is called a user configured interface level dynamic bypass configuration. Dynamic bypass is disabled, by default, in interface mode unless it is enabled through global configured **enable-all-interfaces** command.

When the dynamic bypass is already enabled on the interface through global **enable-all-interfaces** command, this command changes the interface status to the user configured interface level dynamic bypass configuration. When the user configures the interface level dynamic bypass to the disabled status, this command retains the existing disabled state. When an interface level dynamic bypass is enabled, a facility protected LSP does not use a dynamic bypass LSP.

Use the **dynamic-bypass** command to enable the dynamic bypass on a MPLS interface.

```
device(config-mpls-if-e100-2/3)# dynamic-bypass
device(config-mpls-if-e100-2/3-dynamic-bypass)#enable
```

Syntax: [no] **dynamic-bypass**

Syntax: **enable**

The [**no**] form of the command disables dynamic bypass on the MPLS interface.

Setting the maximum number of dynamic bypass LSPs per MP

Use the **max-bypassesper-mp** command to set the maximum number of dynamic bypass LSPs that are configurable MPLS interface mode. This is the limit for total number of dynamic bypass LSPs that can be created to a merge point. When this parameter is not configured under interface mode, the **global max-bypasses-per-mp** parameter value is considered for this parameter. A PLR can have 'M' number of merge points with respect to a Protected LSP. There can be 'N' number of protected LSPs riding on an interface with a dynamic bypass enabled. By default, the maximum number of dynamic bypasses per MP that can be created per MP is as per the corresponding global configuration. When the max-bypasses-per-mp limit is changed to a value which is less than the current active number of dynamic bypasses per mp, then the limit changes to the new value and used for the next new creations. Existing dynamic bypasses exceeding this number are not deleted.

```
device(config-mpls-if-e100-2/3-dynamic-bypass)# max-bypasses-per-mp5
device(config-mpls-if-e100-2/3-dynamic-bypass)#
```

Syntax: [no] **max-bypasses-per-mp** *number*

Use the *number* parameter to set the number of bypass LSPs that can be created to a MP router ID.

Use the [**no**] form of the command to return the settings to the global mode set **max-bypasses-per-mp** parameter value.

Specifying the name prefix

Use the **name-prefix** interface command to specify a name prefix for the dynamic bypass LSP. When configured, the dynamic bypass LSPs have their LSP names starting with this name prefix, appended by interface IP and instance number. Default name for the prefix string is **dbyp**. The name prefix configuration is allowed only when there no existing dynamic bypasses corresponding to a dynamic bypass interface. When the user wants to change the name prefix, the user must disable the dynamic bypass on the interface and reconfigure the name prefix, then re-enable the dynamic bypass on the interface.

```
device(config-mpls-if-e100-2/3-dynamic-bypass)# name-prefix "mydps"
device(config-mpls-if-e100-2/3-dynamic-bypass)#
```

Syntax: [no] **name-prefix** *name-string*

Use the *name-string* parameter to name the prefix that has to be prefixed to the auto generated dynamic bypass LSP name while creating a dynamic bypass LSP.

The [**no**] form of the command returns the settings to default.

Setting the priority

The **priority** command is an interface level setup and holding priority. This can be configured with priority levels from zero to seven for a dynamic Bypass LSP corresponding to a protected link. These priority values are used while creating dynamic bypass LSPs. By default, setup priority is seven and the hold priority is zero. When the interface mode priority values are not configured and there are riding backups on the dynamic bypass, the dynamic bypass re-optimization new holding priority is the maximum priority of the currently riding backups.

```
device(config-mpls-if-e100-2/3-dynamic-bypass)# priority 3 6
device(config-mpls-if-e100-2/3-dynamic-bypass)#
```

Syntax: [no] **priority** [*setuppriority* | *holding-priority*]

Use the [**setuppriority** | **holding-priority**] parameters to create a dynamic bypass.

Use the [**no**] form of the command to return the settings to the default value.

Enabling the record route option

An interface level record route parameter can be configured for a dynamic bypass LSP corresponding to a protected link. Use the **record** command to enable or disable the dynamic bypass LSP record route options. Based on the value of this parameter, dynamic bypass LSPs are created with their record route option enabled or disabled.

```
device(config-mpls-if-e100-2/3-dynamic-bypass)# record
device(config-mpls-if-e100-2/3-dynamic-bypass)#
```

Syntax: [no] **record**

Use the [**no**] form of the command to disable the record route option. The interface dynamic bypass configuration shows 'no record'.

Configuring non-adaptive LSPs

Dynamic Bypass LSPs are by default, adaptive in nature. There is a provision to create a dynamic bypass LSP with non-adaptive nature. When configured as **no adaptive**, this reflects on interface configuration as 'no adaptive' and any dynamic bypass LSP which is created further is non-adaptive by default. To re-enable the adaptive nature, use the **adaptive** command. Based on the value of this parameter, the dynamic bypass LSPs is created as an adaptive bypass LSP or non-adaptive bypass LSP.

```
device(config-mpls-if-e100-2/3-dynamic-bypass)# no adaptive
device(config-mpls-if-e100-2/3-dynamic-bypass)#
```

Use the [no] form of the command to disable the adaptive parameter. All of the dynamic bypasses to be created are non-adaptive. The interface dynamic bypass configuration shows 'no adaptive'.

Configuring administrative groups

Use the interface level **exclude-any | include-all | include-any** command to configure administrative groups for dynamic bypass LSPs to be created corresponding to a protected link.

```
device(config-mpls-if-e100-2/3-dynamic-bypass)# include-all 4 5
device(config-mpls-if-e100-2/3-dynamic-bypass)#
```

Syntax: [no] **include-all include-any | exclude-any** *group-no or name*

Use the *group-no or name* parameter for the following:

- **include-all** *group-no or name* : include all type admin group number or group name.
- **include-any** *group-no or name* : include any type admin group number or name.
- **exclude-any** *group-no or name* : include any type admin group number or name.

Group number value ranges for all three types are 0-31. The default value is *no admin group* configured for all three types.

Use the [no] form of the command to return the settings to the default value.

Setting the re-optimize timer

Use the **reoptimize-timer** command to configure a re-optimization timer value for all the dynamic bypass LSPs that are being created corresponding to a protected interface. When configured, this value overrides the global mode configured value. Re-optimization can be disabled corresponding to an interface by setting it to value zero. When a dynamic bypass is non adaptive, the re-optimization timer is not be considered for the dynamic bypass LSP.

```
device(config-mpls-if-e100-2/3-dynamic-bypass)# reoptimize-timer 300
device(config-mpls-if-e100-2/3-dynamic-bypass)#
```

Syntax: [no] **reoptimize-timer** *number*

Use the *number* parameter to set the re-optimization timer value when creating dynamic bypass LSPs.

Use the [no] form of the command to return the settings to the global mode re-optimization timer value.

Displaying dynamic bypass information

The **show mpls dynamic-bypass** interface command has three fields: Active Status, Admin Type, and Admin Status.

- **Admin Type:** Indicates dynamic bypass configuration on the interface is because of local (interface) or global (MPLS device) mode configuration.
- **Admin Status:** Indicates when the user has enabled the dynamic bypass on the interface. **UP** indicates interface dynamic bypass is admin configuration enabled, **DOWN** implies admin configuration disabled. This admin configuration represents user explicit interface configuration (when present), otherwise enable-all-interfaces (when present).
- **Active Status:** Enabled indicates net effect of global and local configuration enable leads to status is **UP**. Disabled indicates either local or global admin is **DOWN** and net effect is disabled add admin status.

Sample configurations

Global dynamic bypass configuration example

```
device(config-mpls)# dynamic-bypass
device(config-mpls-dynamic-bypass)# enable
device(config-mpls-dynamic-bypass)# enable-all-interfaces
device(config-mpls-dynamic-bypass)# max-bypasses 150
device(config-mpls-dynamic-bypass)# max-bypasses-per-mp 8
device(config-mpls-dynamic-bypass)# reoptimize-timer 300
device(config-mpls-dynamic-bypass)# disable
```

Dynamic bypass interface configuration example

```
device(config-mpls-if-e100-2/3)# dynamic-bypass
device(config-mpls-if-e100-2/3-dynamic-bypass)# enable
device(config-mpls-if-e100-2/3-dynamic-bypass)# max-bypasses-per-mp 6
device(config-mpls-if-e100-2/3-dynamic-bypass)# primary-path "mydbyp-path"
device(config-mpls-if-e100-2/3-dynamic-bypass)# traffic-eng mean-rate 1000
device(config-mpls-if-e100-2/3-dynamic-bypass)# reoptimize-timer 300
device(config-mpls-if-e100-2/3-dynamic-bypass)# priority 3 6
device(config-mpls-if-e100-2/3-dynamic-bypass)# hop-limit 4
device(config-mpls-if-e100-2/3-dynamic-bypass)# tie-breaking least-fill
device(config-mpls-if-e100-2/3-dynamic-bypass)# record
device(config-mpls-if-e100-2/3-dynamic-bypass)# cos 4
device(config-mpls-if-e100-2/3-dynamic-bypass)# include-any 4 5
device(config-mpls-if-e100-2/3-dynamic-bypass)# include-all 1 2
device(config-mpls-if-e100-2/3-dynamic-bypass)# exclude-all 3 6
device(config-mpls-if-e100-2/3-dynamic-bypass)# from 10.11.11.11
device(config-mpls-if-e100-2/3-dynamic-bypass)# no adaptive
device(config-mpls-if-e100-2/3-dynamic-bypass)# name-prefix "MyDbyp"
device(config-mpls-if-e100-2/3-dynamic-bypass)# reoptimize
device(config-mpls-if-e100-2/3-dynamic-bypass)# disable
```

Supported scenarios

Scenario A: Dynamic bypass creation to NNHOP

A facility protected LSP triggers the creation of a dynamic bypass LSP. A dynamic bypass LSPs destination is based on the merge point selection order of FRR backup path. When there is an existing static or dynamic bypass that satisfies the backup path constraints, it chooses to ride the backup LSP or it creates new dynamic bypass. A path is calculated by considering NNHOP as the first destination. When the path computation is successful, the dynamic bypass is signaled. This creates a node protection dynamic bypass LSP as shown in Figure 28 below.

FIGURE 28 Automatic creation of dynamic creation and merge point selection
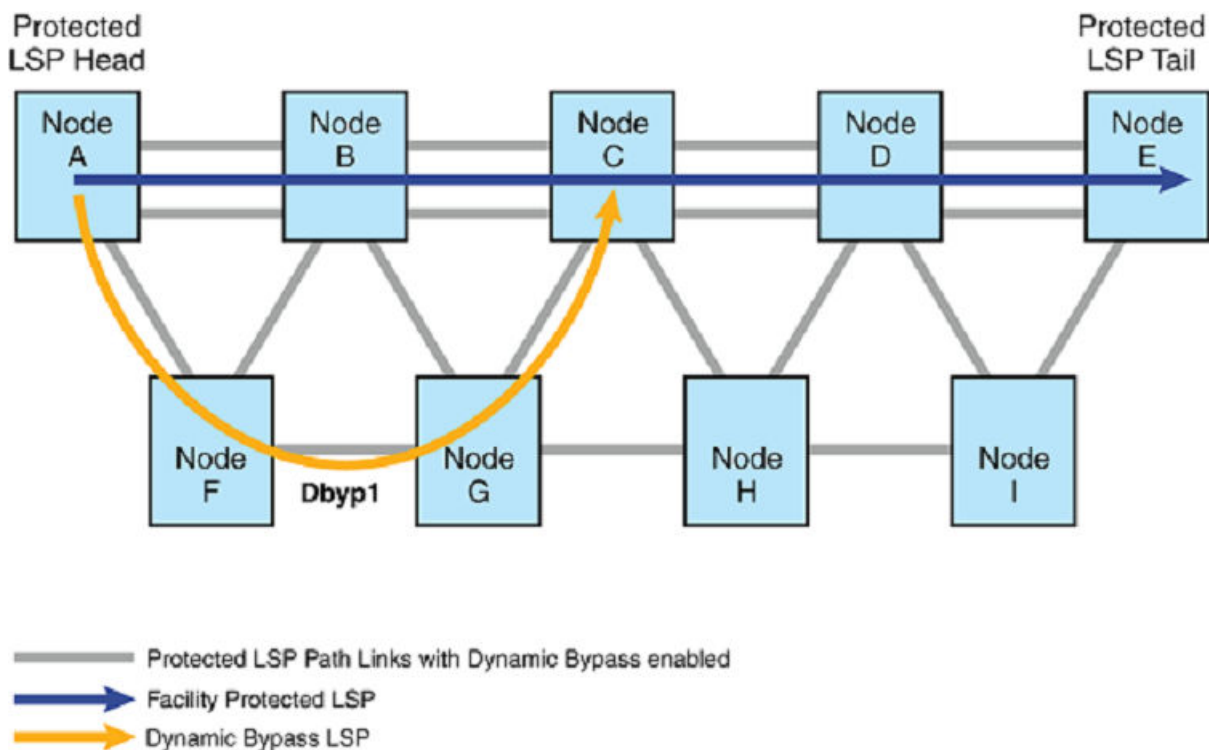
Auto creation of Dynamic Bypass LSP and Merge Point selection.

- NNHOP MP first - Tries to create a dynamic bypass LSP to NNHOP Node
- If Path to NNHOP not available then tries to create a dynamic bypass LSP to NHOP Node
- If Path to NHOP not available then tries to create a dynamic bypass LSP to NNNHOP Node
- If Path to NNNHOP not available then tries to create a dynamic bypass LSP to NNNNHOP Node

Scenario B: Dynamic bypass creation to NHOP

When the path computation to NNHOP node fails, a path is calculated by considering NHOP as the destination. When the path computation is successful, the dynamic bypass is signaled. This creates a link protection dynamic bypass LSP as shown in Figure 29 below.

FIGURE 29 Dynamic bypass creation to NHOP
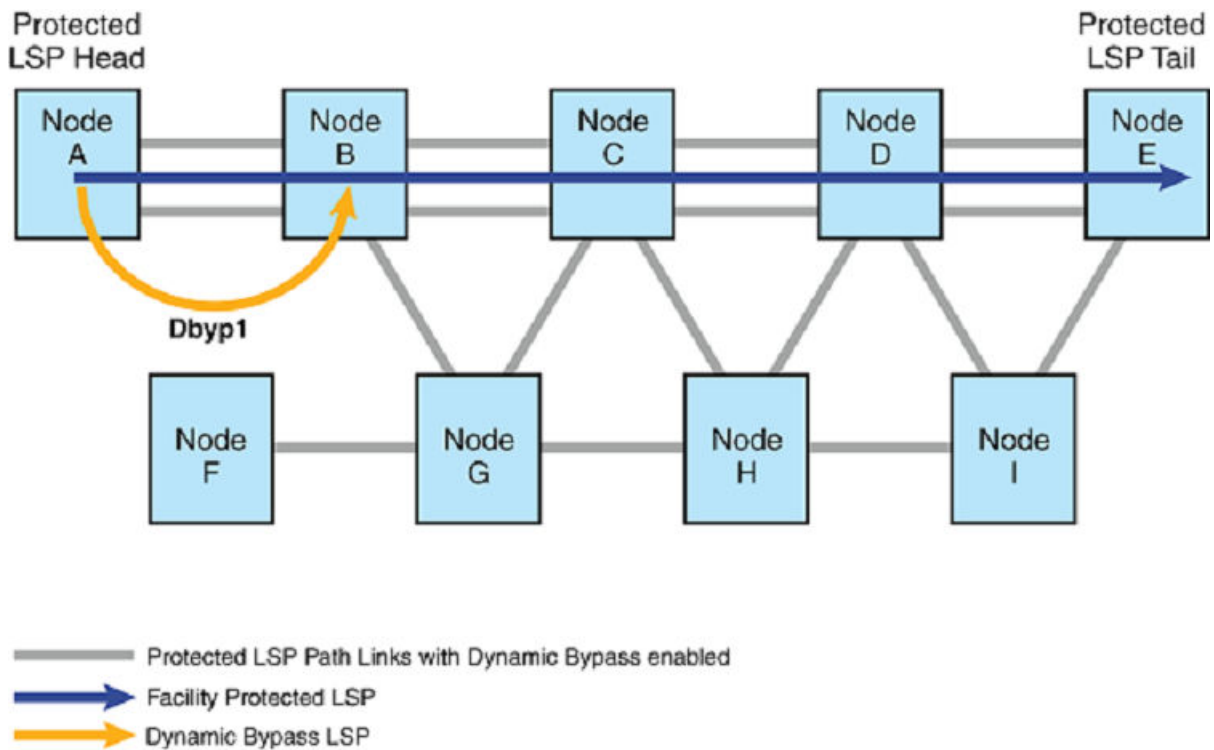
Auto creation of Dynamic Bypass LSP and Merge Point selection.

- NNHOP MP first - Tries to create a dynamic bypass LSP to NNHOP Node
- If Path to NNHOP not available then tries to create a dynamic bypass LSP to NHOP Node
- If Path to NHOP not available then tries to create a dynamic bypass LSP to NNNHOP Node
- If Path to NNNHOP not available then tries to create a dynamic bypass LSP to NNNNHOP Node

Scenario C: Dynamic bypass creation to NNNHOP

When a path computation to NHOP node fails, a path is calculated by considering NNNHOP as the destination. When a path computation is successful, the dynamic bypass is signaled. This creates a node protection dynamic bypass LSP as shown in Figure 30 below:

FIGURE 30 Dynamic bypass creation to NNNHOP
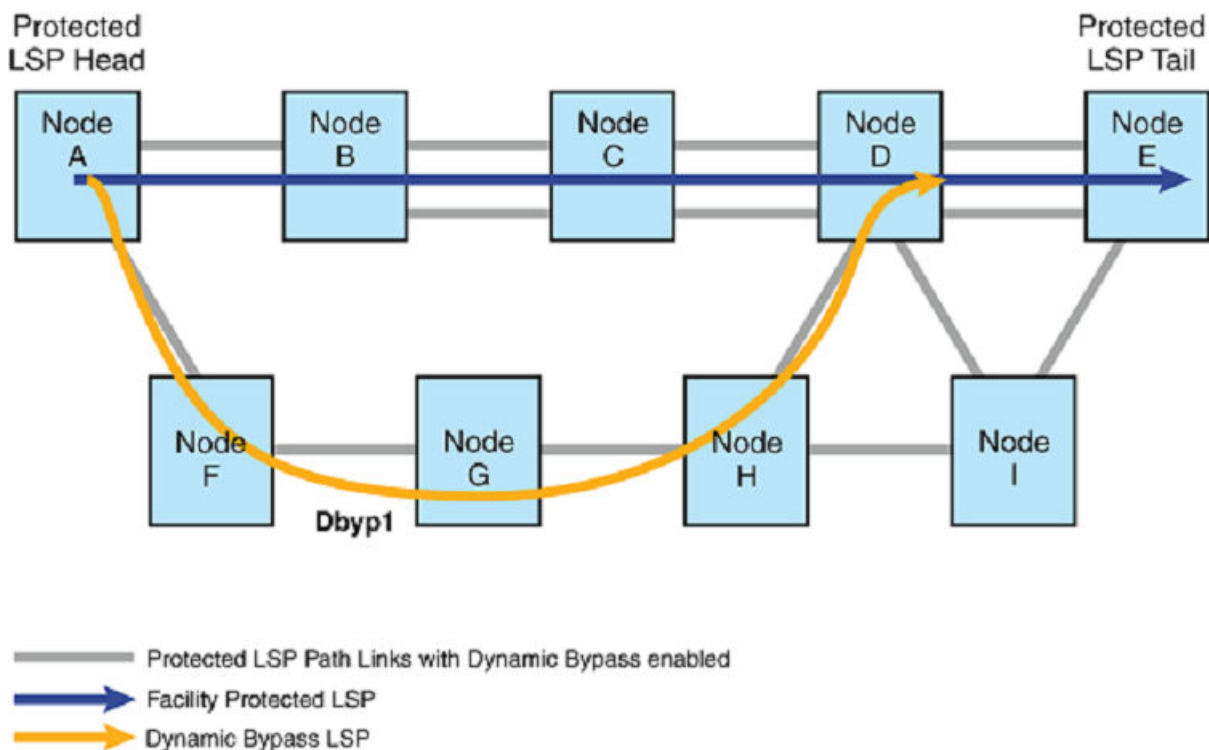
Auto creation of Dynamic Bypass LSP and Merge Point selection.

- NNHOP MP first - Tries to create a dynamic bypass LSP to NNHOP Node
- If Path to NNHOP not available then tries to create a dynamic bypass LSP to NHOP Node
- If Path to NHOP not available then tries to create a dynamic bypass LSP to NNNHOP Node
- If Path to NNNHOP not available then tries to create a dynamic bypass LSP to NNNNHOP Node

Scenario D: Dynamic bypass creation from all PLRs

Dynamic bypass LSP creation on a fully connected network is as below. When there is path available, All PLRs, except penultimate node, creates dynamic bypass LSPs with node protection. This is illustrated in Figure 31 below:

FIGURE 31 Automatic creation of a dynamic bypass LSP for a facility protected LSP
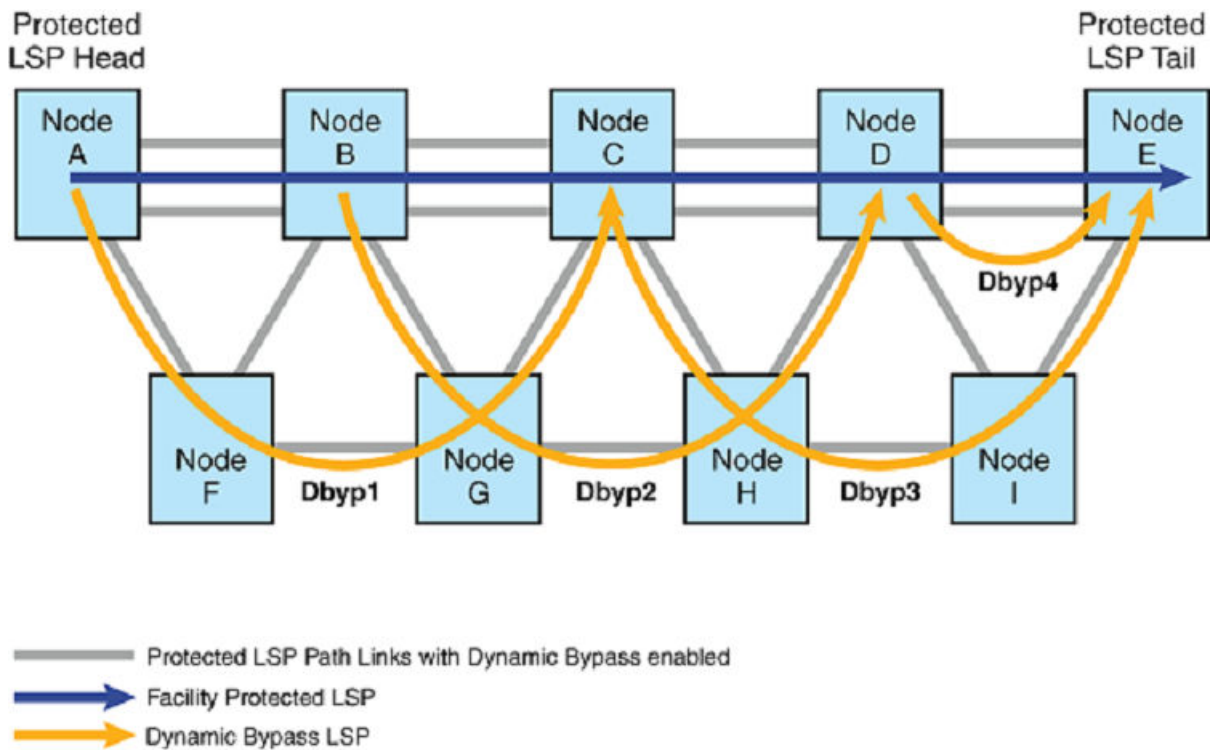
Auto creation of Dynamic Bypass LSP for a Facility Protected LSP.
- PLR A, B, C creates dynamic Bypass to NNHOP - Node protection
- PLR D creates dynamic Bypass to NHOP - Link protection

Scenario E: Dynamic bypass creation from all PLRs

Dynamic bypass LSP creation on a fully connected network is as below. When there is a path available, all PLRs, except penultimate node, creates dynamic bypass LSPs with node protection. This is illustrated in Figure 32 below.

FIGURE 32 Dynamic bypass creation from all PLRs
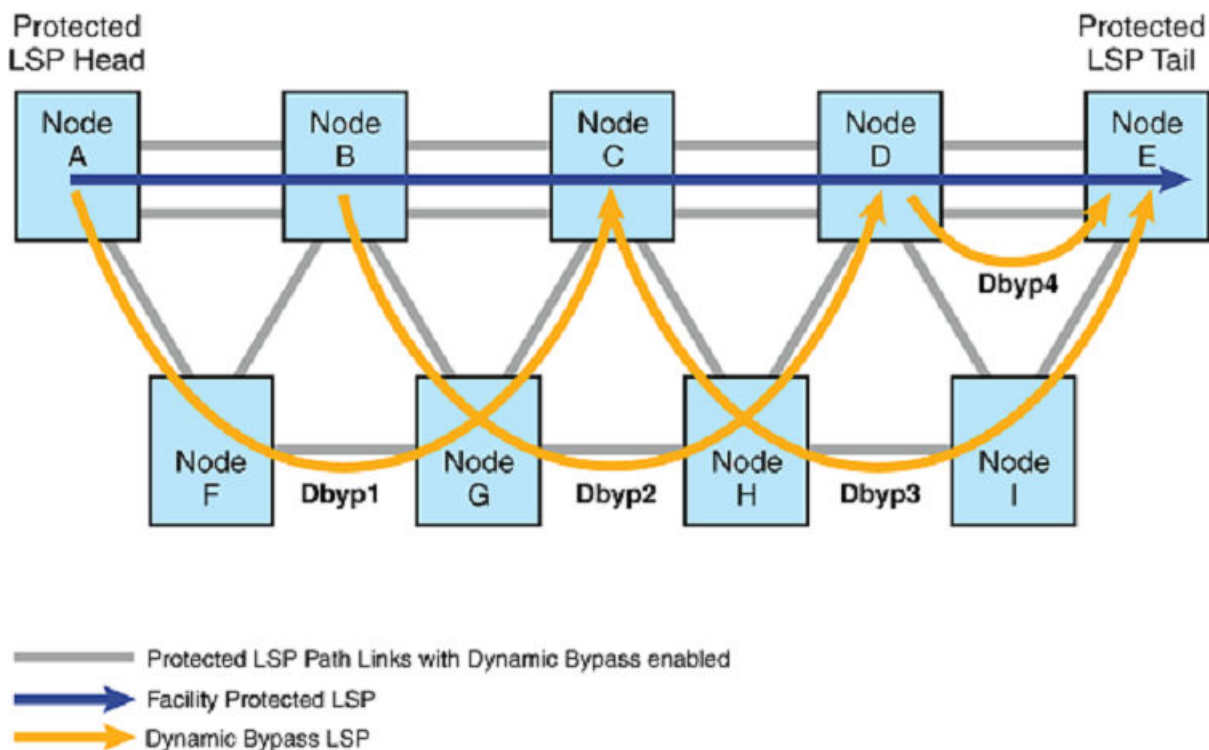
Auto creation of Dynamic Bypass LSP for a Facility Protected LSP.
* PLR A, B, C creates dynamic Bypass to NNHOP - Node protection
* PLR D creates dynamic Bypass to NHOP - Link protection

Scenario F: Dynamic bypass creation with link protection at PLRs

When there is no path for NNHOP node protection and there is path for NHOP, link protection dynamic bypass LSP is created as shown in Figure 33, Figure 33, and Figure 35 below.

FIGURE 33 Dynamic bypass creation with link protection PLRs.a
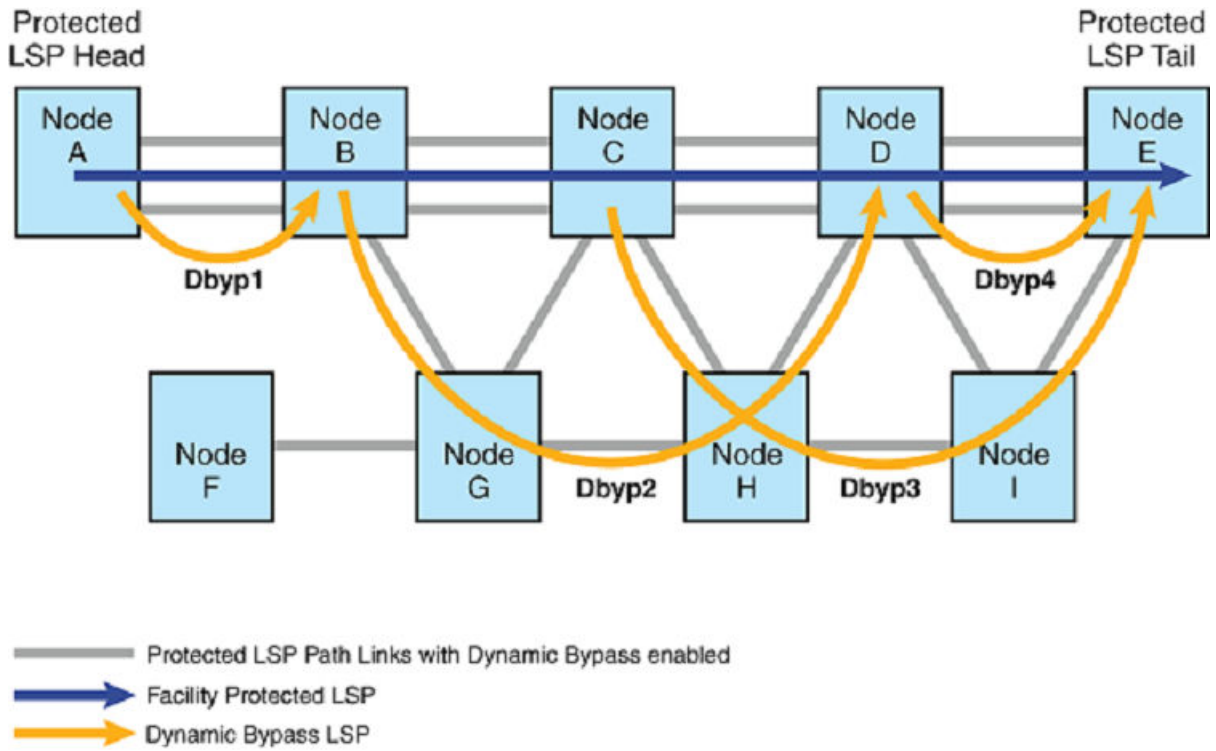
Auto creation of Dynamic Bypass LSP for a Facility Protected LSP.

- PLR A creates dynamic Bypass to NHOP, since NNHOP path not available - Link protection
- PLR B, C creates dynamic Bypass to NNHOP - Node protection
- PLR D creates dynamic Bypass to NHOP - Link protection

FIGURE 34 Dynamic bypass creation with link protection PLRs.b
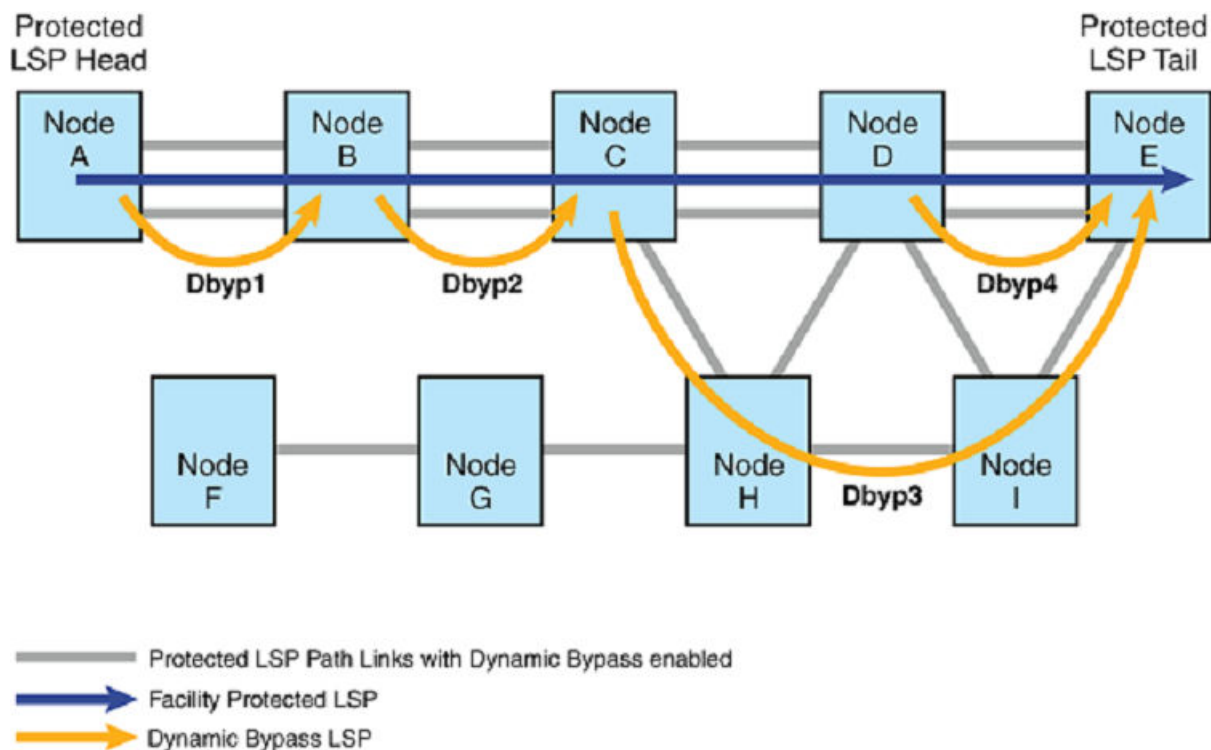
Auto creation of Dynamic Bypass LSP for a Facility Protected LSP.

- PLR A creates dynamic Bypass to NHOP, since NNHOP path not available - Link protection
- PLR B creates dynamic Bypass to NHOP, since NNHOP path not available - Link protection
- PLR C creates dynamic Bypass to NNHOP - Node protection
- PLR D creates dynamic Bypass to NHOP - Link protection

FIGURE 35 Dynamic bypass creation with link protection PLRs.c
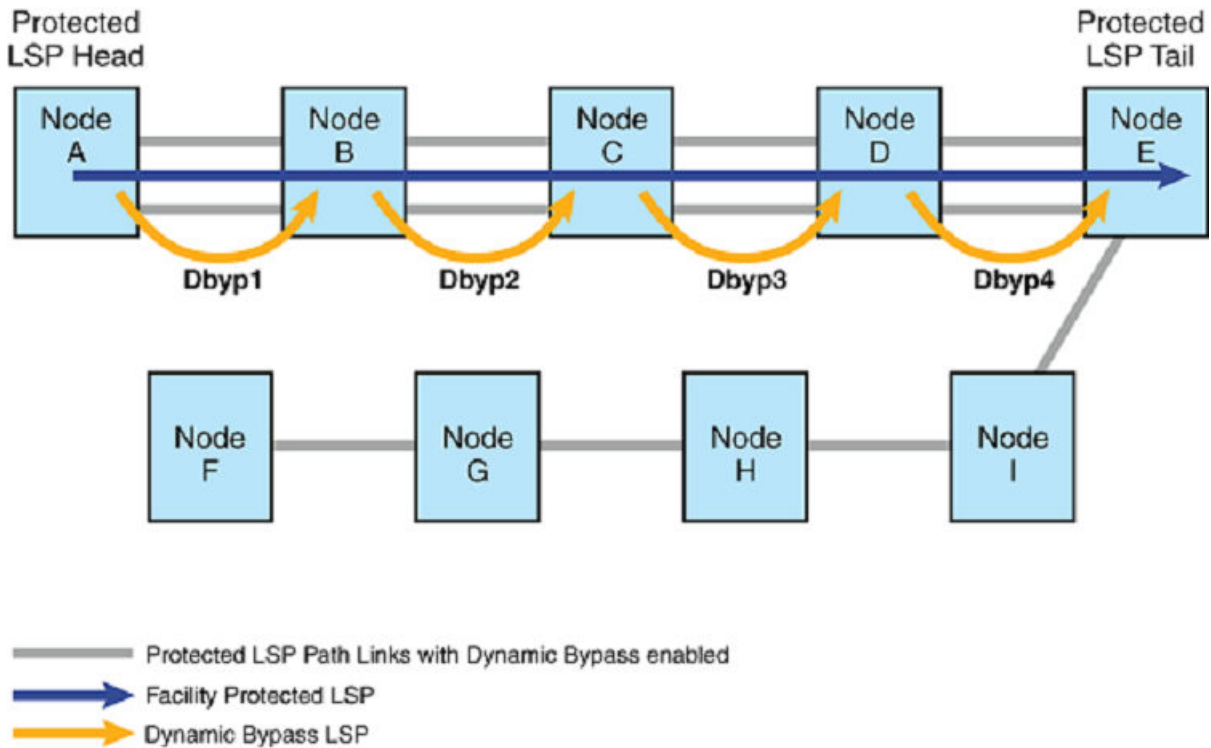
Protected LSP Path Links with Dynamic Bypass enabled
Facility Protected LSP
Dynamic Bypass LSP

Auto creation of Dynamic Bypass LSP for a Facility Protected LSP.
- PLR A creates dynamic Bypass to NHOP, since NNHOP path not available - Link protection
- PLR B creates dynamic Bypass to NHOP, since NNHOP path not available - Link protection
- PLR B creates dynamic Bypass to NHOP, since NNHOP path not available - Link protection
- PLR D creates dynamic Bypass to NHOP - Link protection

RSVP LSP with FRR

RSVP LSP with *fast reroute (FRR)* protection can use two different protection methods. This documentation will use the terms of **detour** backup and **facility** backup going forward for differentiation. Detour backup establishes backup path with detour sessions from *point of local repair (PLR)* to *merge point (MP)* while Facility backup uses bypass LSP as tunnel to establish backup path from PLR to MP. This feature focuses on the algorithm of facility backup path computation and allows you to change from existing very restricted qualification criteria of selecting bypass LSP to much relaxed set of criteria.

This feature is disabled by default and Bypass selection or Dynamic bypass creation for a backups is as per existing selection mechanism. When the user enables liberal bypass option, new Bypass selection, or dynamic bypass, creation comes into effect.

TABLE 7 RSVP bypass LSP terms

Term	Meaning
FRR, Protected LSP	RSVP Fast Reroute enabled LSP
PLR	FRR Point of Local Repair. Can be any node along Protected LSP path, except egress

TABLE 7 RSVP bypass LSP terms (continued)

Term	Meaning
MP	FRR Merge Point. Can be any node along Protected LSP path, except ingress
Backup Path	Facility backup FRR protecting path from PLR to MP
Bypass LSP	LSP tunnels used by Backup paths for Facility backup FRR. Backup Paths Ride on Bypass LSP from PLR to MP. Bypass LSP provides N:1 FRR protection.
Static Bypass	User created Bypass LSP
Dynamic Bypass	Bypass LSP which are created on demand, if there are no existing bypass LSPs satisfying backup Path constraints
Backup Selection	Mechanism in which a protected LSP selects a Bypass LSP for its backup path at a PLR.

Specifications

RSVP FRR LSP with detour backup or facility backup applies the same algorithm for backup path computation. The algorithm is designed to follow RFCs to avoid various issues intrinsic to detour backup type such as early session merging and bandwidth sharing. Because bypass LSP provide virtual tunnel to isolate the visibility of backup sessions and protected session, most of those issues are not present in the facility backup mode. Applying the same algorithm in the facility backup as in detour backup computation is to be very conservative. The only benefit of using the current conservative algorithm is to prevent a single failure triggering both the protected session and backup session failures at the same time. The restrictive algorithm can run into situations that no backup path can be established due to bypass LSPs cannot qualify under those restrictions, especially under certain less meshed topology such as single ring topology.

> **NOTE**
> Multi-Service IronWare devices support FRR failover for RSVP LSPs. If the point of failure is a transit router with respect to the LSP path, then failover can be performed (LSP is repaired, and traffic flow restored) very quickly, in less than 50 milliseconds. When the point of failure is ingress router with respect to the LSP path, FRR failover is not quick enough and it may take more than 50 milliseconds to repair the LSP. Traffic loss will be more than 50 milliseconds.

The steps of facility backup path computation, which involves selecting best qualified bypass LSP are:

Merge point selection: PLR backup query process first selects in the order of preferred merge point, based on ingress signaled property. Once a merge point is selected in the order of preference, it selects from the available bypass LSPs reaching this merge point. If no bypass LSP qualify to serve, it moves on to the next preferred merge point. The merge point preference order depends on if ingress signal with the node protection has the desired flag.

A: When the node protection desired flag is present, PLR goes through the merge point in the order of next-next-hop (if present, to achieve node protection), next-hop (link protection), hops after next-next-hop in sequence of traverse, if any are present.

B: When Node protection desired flag is not set, it simply selects the downstream next hops in sequence of traverse (example: next-hop, next-next-hop and so on).

Bypass LSP qualification:

A: Bypass LSP cannot traverse any link (forward directional) traversed by protected session from ingress of LSP to PLR.

B: Bypass LSP cannot traverses any link attached to nodes traversed by protected session between PLR and egress of LSP.

When there are more than the one bypass LSP qualified to serve for backup path, select the ones with lowest LSP cost, if the metric is considered. When there are more than one bypass LSP available with the lowest cost, it selects the one with the lowest number of riding backup sessions.

RSVP per-session statistics

RSVP per-session statistics

Resource Reservation Protocol (RSVP) statistics are a useful troubleshooting tool. At the global and interface levels, statistics are already available. In some cases, especially scaled scenarios, it is helpful to also count these statistics on a per-session level. The same statistics that are available at the global and interface levels are also available at the session level.

RSVP per-session statistics and their applicability

The table below illustrates the collection of RSVP protocol statistics to the session available at the global, interface, and session levels.

TABLE 8 RSVP statistics and their applicability

Statistics Name	Global	Interface	Session
Protocol Packet (stats-Path, Resv, PathErr, ResvErr, PathTear, ResvTear, pathconf)	Yes	Yes	Yes
Path Expired	Yes	No	Yes
Resv Expired	Yes	No	Yes
Unknown Message Type	Yes	Yes	No
Bundle, Ack, Hello, Sumrefresh Packets	Yes	Yes	No
Bad Authorization	Yes	Yes	No
Message Id Error	Yes	Yes	No
Summary Refresh Error	Yes	Yes	No
Nack	Yes	Yes	No
Errored Protocol Packets	Yes	No	Yes
Errored length	Not Applicable	Not Applicable	Not Applicable
Errored version	Not Applicable	Not Applicable	Not Applicable
Errored chksum	Not Applicable	Not Applicable	Not Applicable
Errored malloc	Not Applicable	Not Applicable	Not Applicable

Liberal bypass selection and liberal dynamic bypass

New facility backup computation mode applies an algorithm between an extremely conservative approach and an extremely liberal approach. The changes affect how the bypass LSP is qualified. There are no changes regarding the merge point selection or capability to disable providing node protection or tie breaker from multiple qualified bypass LSPs and how a dynamic bypass LSP path is chosen while creating a dynamic bypass.

For a bypass LSP to qualify for the backup path, it must pass following tests:

- Bypass LSP cannot traverse any forward direction links used by protected session
- While the Liberal Bypass option is enabled, the bypass LSP cannot traverse any nodes between PLR and MP
- Bypass LSP cannot reach MP where protected LSP is already in a repaired state
- Bypass can traverse the nodes between MP and Egress

- Bypass LSP cannot traverse the locally protected link (this is implicit because the bypass LSP is already avoided using protected interface)

In summary, all Bypass selection constraints remain same as per existing bypass selection except that the Bypass can traverse the nodes between the Protected LSP Merge Point and Egress.

At present by default, Dynamic Bypass path computation excludes the nodes between MP and Egress. Now when Liberal Bypass is enabled, Dynamic Bypass creation path computation can compute Dynamic Bypass path including nodes between Merge Point and Egress of protected session.

FIGURE 36 Bypass LSP selection: cannot traverse any node between PLR and MP
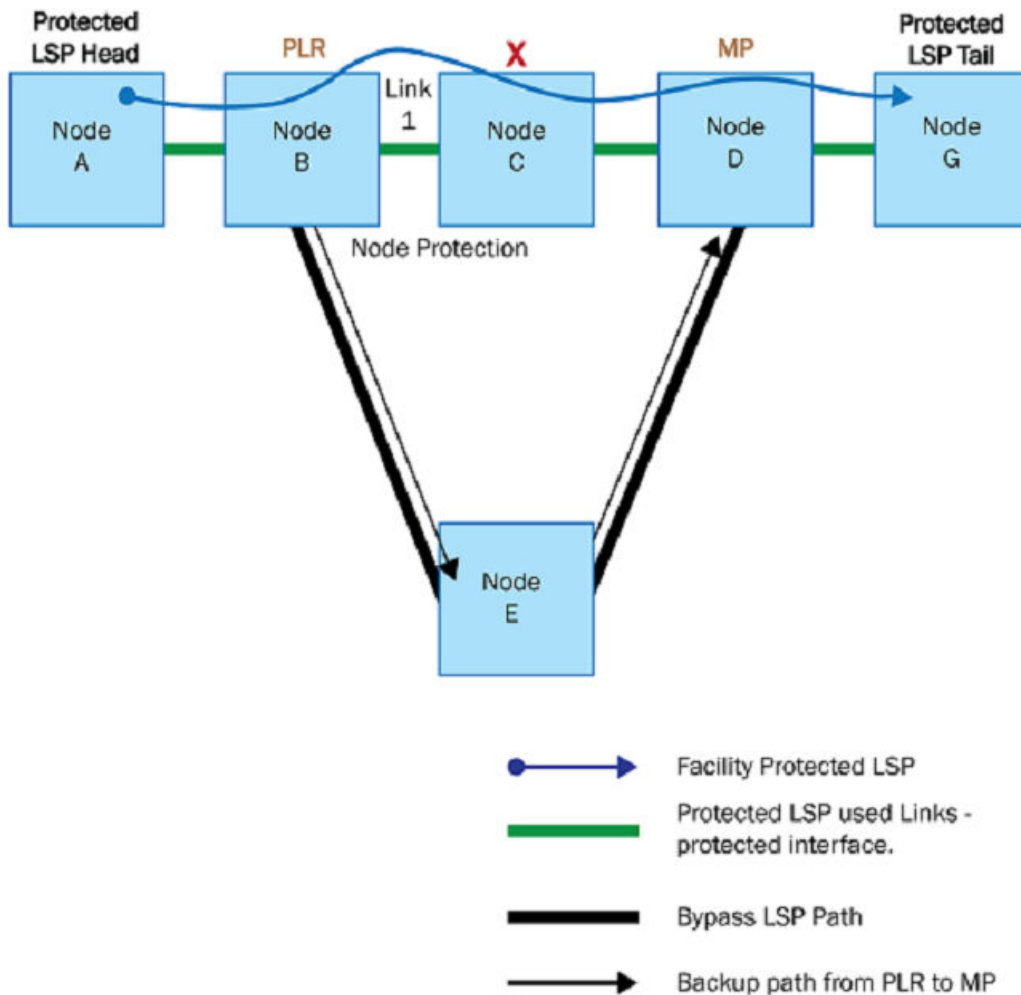
FIGURE 37 Bypass LSP selection: traversing the downstream node
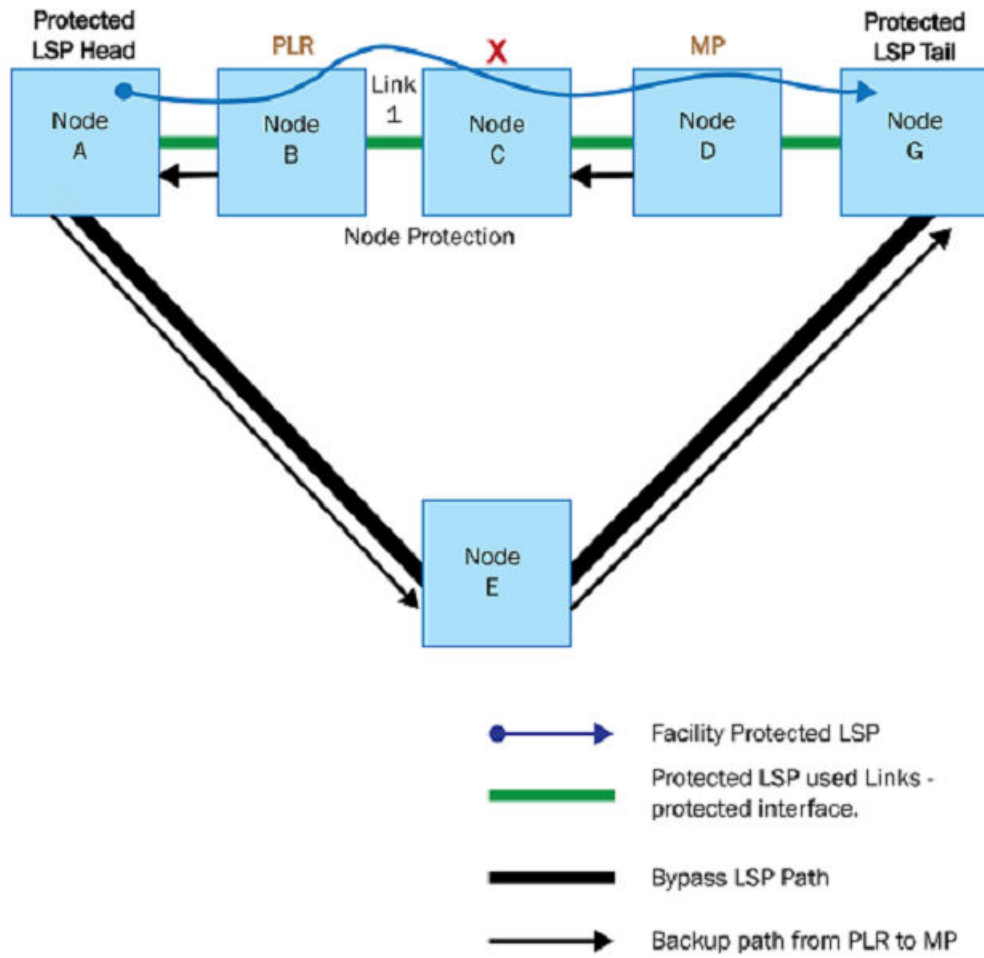
Bypass LSP Selection: traversing thr downstream node

The following cases are NOT supported:

FIGURE 38 Cannot traverse any link between PLR and MP
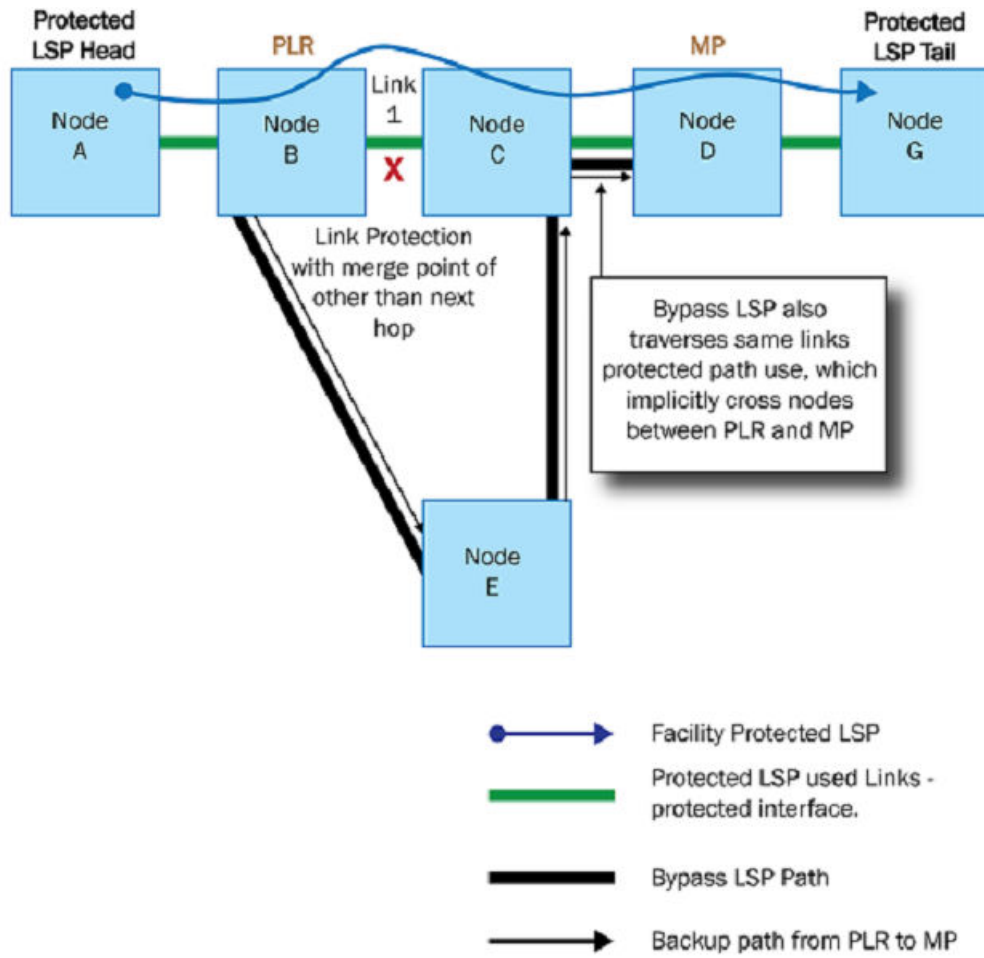
FIGURE 39 Cannot traverse any node between PLR and MP
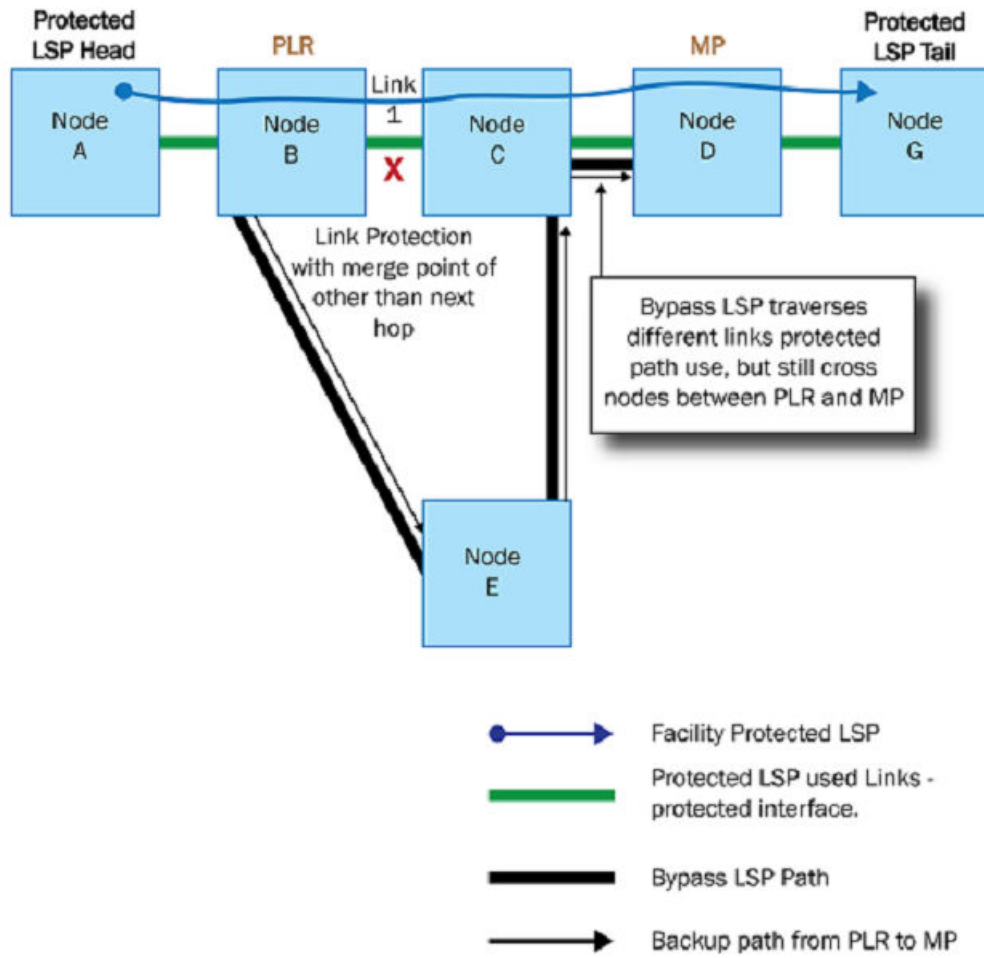
Bypass LSP Selection: cannot traverse any link between PLR and MP

The following diagram illustrates the difference between node protection and link protection with merge point other than next hop:

FIGURE 40 True node protection
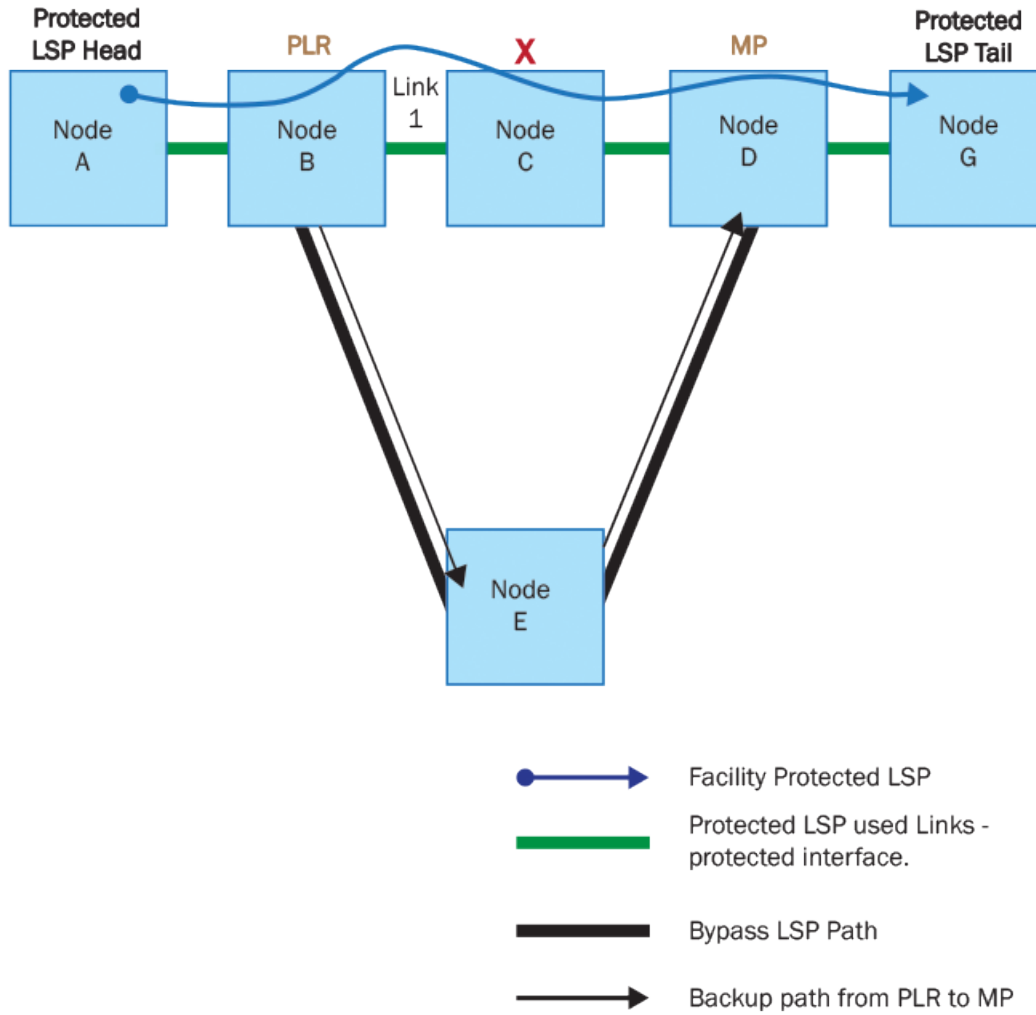
True node protection

The bypass LSP provides link protection which does not need to bypass node C:

FIGURE 41 Potential link protection with traversing link or node between PLR and MP
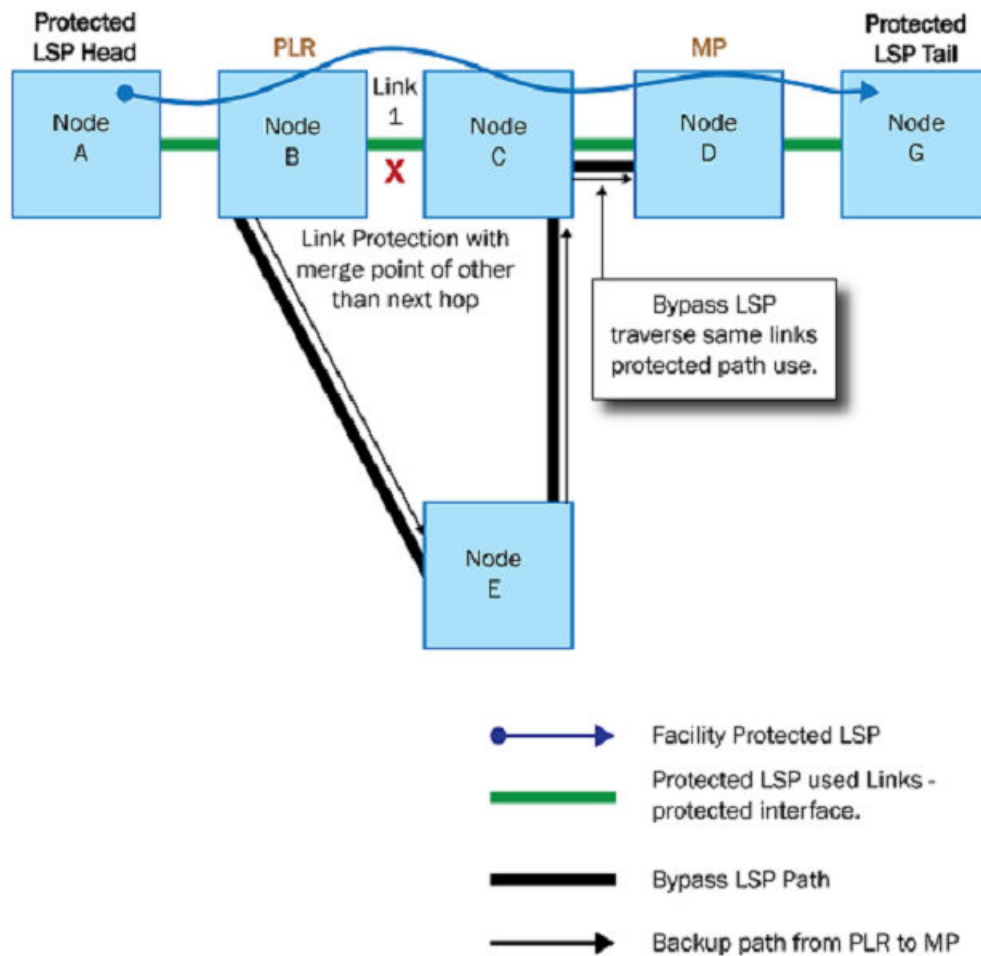
Bypass LSP Selection: Potential link protection with traversing link/node between PLR and MP

Backward compatibility

This is a software only feature and does not require special support from hardware. There are no backward compatibility issues as this feature is local to RSVP module regarding if the bypass LSP can serve as backup protection. With the new mode turned on, there can be additional bypass LSPs qualifying for the backup path.

Limitations

New Dynamic bypass creation at upstream after a failure at any downstream of the FRR LSP might fail, do to router ID resolution.

This is applicable only if the dynamic bypass creation is triggered from the same FRR LSP with any of the downstream failed Link. The reasoning behind this is that a failed link IP in the avoid Node List is not reachable and its node ID cannot be resolved. In such a scenario, the path calculation to a new Dynamic bypass fails.

Upgrade and downgrade considerations

The CLI changes are longer seen on a downgrade and the feature can lead to backup sessions not established due to more restricted qualification criteria in older releases.

IP Traceroute over MPLS

RFC Support

The Extreme*Internet Protocol (IP)* Traceroute over *Multi-Protocol Label Switching (MPLS)* implementation complies with the following RFC Internet Drafts:

- *RFC 3032*, MPLS Label Stack Encoding
- *RFC 4884*, Extended ICMP to Support Multi-Part Messages
- *RFC 4950*, ICMP Extensions for MPLS

Standard traceroute

Traceroute is a diagnostic utility that allows the user to troubleshoot a network path by interactively sending *Internet Control Message Protocol (ICMP)* packets through an IP network from a source to a destination. Packets have a defined TTL and sent to a port that is known to be invalid on the destination device (usually above 3300). At each hop, the transit device decrements the *Time-to-Live (TTL)* value contained in the IP header portion of the datagram by one. Based on the remaining TTL value, the device performs one of the following actions:

- TTL > 0: The device passes the packet to the next device using standard IP routing protocols.
- TTL = 0: The device drops the packet, which is now expired, and returns an ICMP response of type 11, ttl-exceeded, along with a portion of the original datagram to the source device that originated the traceroute probe.

With each traceroute iteration, the source device increments the TTL value of the packets by one. This causes the packets to be forwarded another hop to the next transit device until the TTL is large enough for the probe to reach the destination.

When the destination device receives the packets, it attempts to connect to the port specified in the ICMP. The attempt fails because the port is invalid. The recipient generates an ICMP message of type 3, destination unreachable, and sends it back to the source device that originated the probe.

Based on the ICMP messages received during the traceroute operation, the source device obtains the following information:

- The number of hops traversed by the traceroute probe until it reaches its destination.
- The IP addresses for each of the devices the probes encounter along their path from the source to the destination.
- The *round-trip time (RTT)* for each successive probe. The RTT value is the sum of the time a packet travels until it expires, plus the time the ICMP message takes to return to the source.

The following example illustrates the output of the **traceroute** command traversing seven hops in a standard IP network.

```
device# traceroute 10.125.31.70
Type Control-c to abort
Tracing the route to IP node (10.157.22.199) from 1 to 30 hops
  1     4 ms    <1 ms    <1 ms 10.31.20.25
  2    <1 ms    <1 ms    <1 ms 10.16.200.121
  3    <1 ms    <1 ms    <1 ms 10.110.111.102
  4    <1 ms    <1 ms    <1 ms 10.49.131.1
  5    <1 ms    <1 ms    <1 ms 10.49.130.18
  6    <1 ms    <1 ms     1 ms 10.125.199.61
  7     1 ms     3 ms     2 ms 10.125.31.70T
```

Each line of output represents a hop along the IP network path. For each packet sent, the **traceroute** command records the RTT in milliseconds and the IP address of the device that returned the ICMP ttl-exceeded message. An asterisk (*) indicates that no information could be obtained for the specified hop or the **traceroute** command timed out.

Traceroute in an MPLS-enabled network

The standard traceroute implementation is insufficient for diagnosing Layer 3 routing problems in an MPLS environment, such as a provider network configured to tunnel a customer's *Virtual Private Network (VPN)* traffic through a public backbone.

Standard traceroute relies on IP forwarding based on routing table lookup. It requires that the IP addresses of the source and the destination are transparent to the transit *Label Switch Routers (LSRs)* so they can route ICMP error responses back to the source. When traffic is tunneled through an MPLS domain, IP addresses outside the MPLS domain may not be available to provider transit nodes. In a typical Layer 3 VPN, only the *Provider Edge (PE)* routers have IP routes to *Customer Edge (CE)* devices and can use IP addresses to send traffic to and from the MPLS domain. Once a packet enters the MPLS domain, transit LSRs must rely on label information to move traffic along pre-configured *Label-Switched Paths (LSPs)*.

Because of its dependence on IP routing protocols, the user cannot use the standard **traceroute** command to troubleshoot and identify problems within an MPLS domain, such as a faulty LSP.

Enhanced traceroute using ICMP label extensions

To troubleshoot MPLS-based routing problems, the user can configure traceroute to use LSP forwarding when sending ICMP packets through an MPLS domain and to report MPLS label information. The user controls the traceroute configuration with the **ip icmp mpls-response** command.

The IP traceroute over MPLS enhancement implements extensions to ICMP error control messages as described in *RFC 4884* and *RFC 4950* . The basic idea is to allow the label stack of an expired IP packet to be appended to the ICMP ttl-exceeded message that is generated when a router drops a traceroute packet due to TTL expiration. Recall that for any transit LSR, the source and destination IP addresses embedded in the IP packet header may have no meaning, and the LSR may find itself in a situation where it is unable to route the ICMP message back to the source.

When the user configures traceroute to use ICMP message extensions, an LSR copies the label stack that encapsulated the original datagram when it arrived at the LSR to the ICMP ttl-exceeded message generated by that LSR when it drops a packet due to TTL expiration. Before discarding an expired packet, the LSR strips the label stack from the datagram and attaches it to the ICMP message. The extended message contains the label stack--an inner VPN label identifying the outgoing VRF interface and an outer label that specifies the next hop, each one with its embedded TTL value--plus the IP addresses of the packet source and destination (embedded in the IP header attached to the ICMP message).

With the label stack of the discarded datagram appended to the ICMP message, the forwarding LSR has two options, depending on the MPLS response configuration:

- When a Layer 3 VPN is used to tunnel customer traffic through the MPLS domain, the transit LSRs have no knowledge of the source and destination IP addresses of the traceroute probe. In this case, the user has the option to configure the LSRs so they route the ICMP messages in the direction of the traceroute destination and back to the source using label switching along predetermined LSPs.

- When IP forwarding is enabled in the MPLS core, the user can configure the LSRs to use IP forwarding to route the ICMP ttl-exceeded messages back to the source. The enhanced **traceroute** command offers the added benefit of reporting MPLS label stack information for each hop along with the traceroute output in addition to the IP source addresses of the transit LSRs.

Tracing a route through an MPLS domain

Figure 42 shows an MPLS-enabled provider network consisting of four LSRs. R1 is the *Provider Edge (PE)* ingress *Label Edge Router (LER)*, R2 and R3 are transit LSRs, and R4 is the PE egress LER. CE1 is a *Customer Edge (CE)* device in San Jose, and CE2 is the destination CE on another customer site in Broomfield.

FIGURE 42 Traceroute in a Layer 3 VPN MPLS cloud
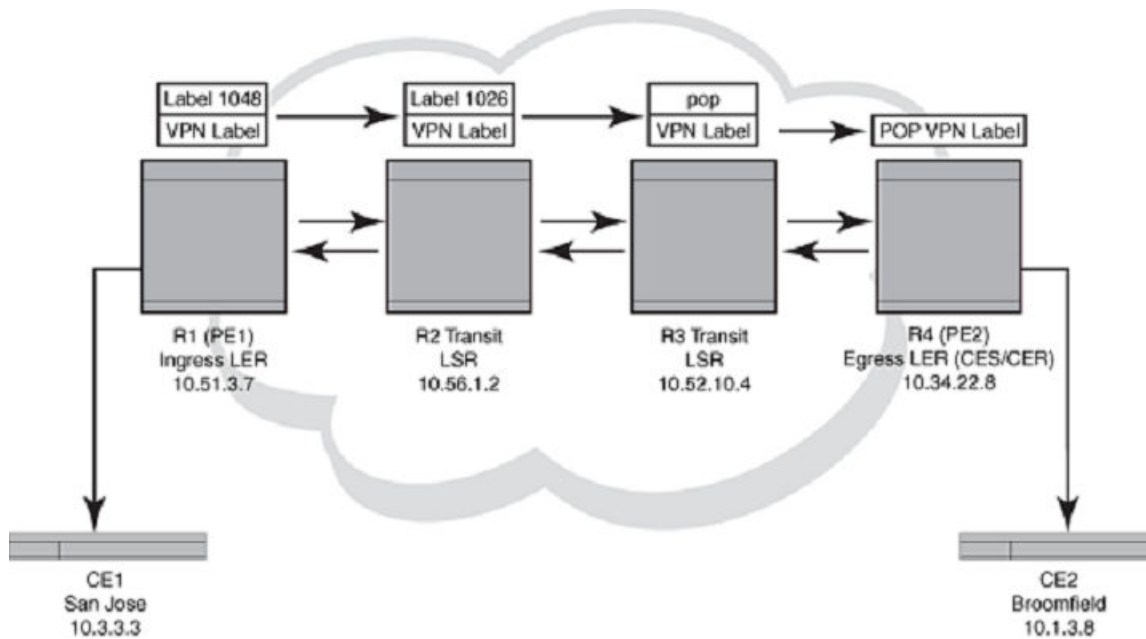
For the purpose of exemplifying the traceroute behavior in an MPLS domain, assume the following:

- Customer traffic is tunneled through a Layer 3 VPN, and traffic within the MPLS core is forwarded by label-switching only.

- Traceroute is configured to generate ICMP responses per ICMP extensions and to use LSPs to forward these messages.

- The PE routers have knowledge of the IP address space on the customer side, whereas the transit LSRs have no such knowledge.

- The **traceroute** command is issued from CE1 to CE2 and reports the following information:

```
device# traceroute 10.1.3.8
Type Control-c to abort
Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
1     <1 ms    <1 ms    <1 ms 10.51.3.7
2     <1 ms    <1 ms    <1 ms 10.56.1.2
      MPLS Label=1048    Exp=7 TTL=1 S=0
      MPLS Label=500000 Exp=7 TTL=1 S=1
3     <1 ms    <1 ms    <1 ms 10.52.10.4
      MPLS Label=1026    Exp=7 TTL=1 S=0
      MPLS Label=500000 Exp=7 TTL=1 S=1
4     <1 ms    <1 ms    <1 ms 10.34.22.8
5     <1 ms    <1 ms    <1 ms 10.1.3.8
```

> **NOTE**
> The traceroute output reports information on a traceroute packet only when its TTL equals one. Label stack information associated with subsequent routing of the ICMP message along the LSPs to the destination and back to the source is not displayed.

In the Figure 42 scenario, the traceroute operation can be described as follows:

1. CE1 in San Jose sends a traceroute probe with a TTL of 1 to its peer in Broomfield, CE2, with the destination IP address of 10.1.3.8. PE1 decrements the packet's TTL by one and drops the expired packet. It then generates an ICMP ttl-exceeded message, and sends it back to the source IP address embedded in the IP header of the discarded datagram. Traceroute reports the PE1 IP address at hop 1, but there is no label information.

```
1     <1 ms    <1 ms    <1 ms 10.51.3.7
```

2. CE1 sends a second traceroute probe to CE2, this time with a TTL value of 2. PE1 decrements the TTL to 1 and pushes an inner VPN label of 500000 and an outer label of 1048 onto the packet to route it to CE2 by way of R2. PE1 also copies the TTL value from the IP header into the TTL field of the labels (recall that TTL propagation must be enabled on the ingress PE).

   R2 decrements the TTL, drops the expired packet, and generates an ICMP ttl-exceeded message. Before dropping the packet, and using the ICMP extension mechanism, R2 copies the packet's label stack plus its IP header and appends both to the ICMP message. The message destination is CE1, the device that emitted the traceroute probe. But because R2 cannot return the ICMP message directly to CE1, R2 uses label-switching to forward the encapsulated ICMP response in the direction of the original traceroute probe along the configured LSPs and back to CE1. R2 sets the TTL in the topmost label to the maximum supported value of 225 to ensure that the message can reach its destination before it times out.

   Traceroute reports the IP address of R2, plus the label stack that was pushed onto the traceroute packet by PE1 and received by R2 when the packet's TTL was 1. Note PE1s inner VPN label of 50000 at the bottom of the stack and the outer label of 1048 at the top.

```
2     <1 ms    <1 ms    <1 ms 10.56.1.2
        MPLS Label=1048 Exp=7 TTL=1 S=0
        MPLS Label=500000 Exp=7 TTL=1 S=1
```

3. The third traceroute probe (TTL=3) is label-switched until it expires at R3. R3 (the *Penultimate Hop Popping (PHP)* LSR) generates the ICMP message, appends the label stack from the expired traceroute packet, and passes it on to PE2 without imposing a label. PE2 forwards the ICMP message back to CE1 along the return LSP.

   Traceroute reports the IP address of R3, plus the label stack which R3 received with the traceroute packet from R2 when the packet's TTL was 1.The packet's label stack includes the inner VPN label at the bottom of the stack, and the outer label 1026, which R2 imposed when it swapped R1s 1048 label.

```
3     <1 ms    <1 ms    <1 ms 10.52.10.4
        MPLS Label=1026 Exp=7 TTL=1 S=0
        MPLS Label=500000 Exp=7 TTL=1 S=1
```

4. CE1 sends a forth traceroute probe with a TTL value of 4. The packet is label-switched until it arrives at PE2 with a TTL value of 1. PE2 drops the packet and generates an ICMP ttl-exceeded message without label stack extension.

   Traceroute reports only the IP address of PE2. There is no label stack to report. R3 popped the outer label before passing the traceroute packet on to PE2, and the Extreme egress router pops the VPN label before sending the ICMP message back to CE1.

```
4     <1 ms    <1 ms    <1 ms 10.34.22.8
5     <1 ms    <1 ms    <1 ms 10.1.3.8
```

5. The fifth traceroute probe has a TTL large enough (TTL=5) for the packets to reach CE2. CE2 generates an ICMP destination unreachable message, which CE2 sends back to CE1.

   Traceroute reports only the IP address of the destination device CE2. No label extension is added because the received packet is not labeled. The destination unreachable message is label-switched back as a normal data packet.

```
5     <1 ms    <1 ms    <1 ms 10.1.3.8
```

Configuring Traceroute over MPLS

The **ip icmp mpls-response** command configures the behavior of the traceroute operation by controlling both the ICMP message format (use ICMP label stack extensions or not) and the manner in which the ICMP messages are forwarded through an MPLS domain (by way of IP routing table lookup or through label-switching using LSPs).

The command is accessible in global configuration mode.

MPLS response is enabled by default. To enable the MPLS response after it was disabled, enter the following command:

```
device(config)# ip icmp mpls-response
```

To specify using LSP to forward the ICMP messages with MPLS label extensions, enter the following command:

```
device(config)# ip icmp mpls-response use-lsp
```

To specify generating ICMP messages without MPLS label extensions, enter the following command:

```
device(config)# ip icmp mpls-response no-label-extensions
```

To disable the IP Traceroute over MPLS feature, enter the following command:

```
device(config)# no ip icmp mpls-response
```

Syntax: **[no] ip icmp mpls-response** [**use-lsp**] [**no-label-extension**]

The **mpls-response** parameter enables the ICMP MPLS response in default mode. The feature is enabled by default and configured to use IP routing to forward ICMP messages.

The **use-lsp** parameter, specifies to forward ICMP error messages based on information encoded in the label stack along the LSPs configured for the MPLS domain. LSP use is disabled by default.

The **no-label-extension** parameter specifies not to use label extensions in the ICMP error messages.

The [**no**] option disables the ICMP MPLS response configuration. When the feature is disabled, standard **traceroute** is used to trace a traffic path through an MPLS domain.

The output of the **show ip traffic** command displays counts for ICMP messages that have been generated by an MPLS-enabled LSR with label extensions and returned to the source of the traceroute probe. Refer to " Displaying IP traffic statistics" for a description of the **show ip traffic** command and associated output.

Use the **show ip traffic** command to verify the ICMP message count the user receives from the traceroute output.

The **show running configuration** command has been modified to include the MPLS response configuration.

> NOTE
> That the default configuration (**mpls-response** enabled and using IP routes) is not displayed in the running configuration.

Prerequisites

- MPLS must be enabled on each of the provider LSRs. Refer to the chapter *"Configuring MPLS Traffic Engineering"* of the *Multi-Service IronWare Multiprotocol Label Switch (MPLS) Configuration Guide* for more information.
- The user must enable TTL propagation on the ingress and egress LERs (PEs) before the user can use the enhanced traceroute feature. TTL propagation ensures that the TTL value of the IP packets received by the ingress LER is copied onto the outgoing MPLS label. Refer to IP-over-MPLS TTL propagation control of the *Multi-Service IronWare Multiprotocol Label Switch (MPLS) Configuration Guide* for more information.

Limitations

- The only ICMP messages supported with this feature are ttl-exceeded messages (Type 11). Destination unreachable messages are not supported.
- IP Traceroute over MPLS requires *Time-To-Live (TTL)* propagation at the ingress and egress *Provider Edge (PE)* routers. The feature is not applicable in contexts where TTL values are not propagated (for example, in a Layer 2 VPN).
- IP Traceroute over MPLS supports a label stack size of up to five entries when displaying MPLS label information in the traceroute output. For label stack sizes greater than five the traceroute output only displays the first five label entries.
- The user cannot use the **traceroute** command in a Layer 3 VPN network when the ICMP response is configured with the **no use-lsp** option. The transit LSRs have no *Virtual Routing and Forwarding (VRF)* concept (only the PE routers do), and consequently no VPN routes when the label-switching mechanism is blocked. Even when the responding LSR may have a route to the traceroute source, the ICMP ttl-exceeded response packet generated by the LSR router cannot be properly routed by the ingress router to the CE router because the response is a normal IP packet and does not have a VPN label. Refer to Scenario B - Layer 3 VPN over MPLS on page 185 for an illustration.
- IP Traceroute over MPLS is not backwards-compatible.
- IP Traceroute over MPLS supports IPv4 traceroute only.

Configuration examples

The MPLS response feature supports four different configuration options for controlling the traceroute behavior in an MPLS domain. The command behavior varies depending on what types of routes are configured for a given MPLS domain.

The following examples assume the same MPLS response configuration for each of the LSRs in the MPLS domain. The user configures the MPLS response parameters per LSR, and to get consistent traceroute behavior, it is desirable for all the LSRs in the MPLS domain to have the same settings. Use the **show running configuration** command to verify the configuration.

The configuration examples refer to the topology described in Figure 1 configured with either IP over MPLS or Layer 3 VPN over MPLS. All Layer 3 VPN examples use a CER 2000 Series or a CES 2000 Series as the egress router.

Using IP traceroute over MPLS with the default configuration

Assumptions: The default MPLS response configuration was disabled on all LSRs in the MPLS domain and the user re-enables the default configuration. In default mode, ICMP messages are generated with label stack extensions and forwarded per IP routing table lookup. ExtremeCE1 is tracing the traffic path to ExtremeCE2 with a destination IP address of 10.1.3.8.

Scenario A - IP over MPLS

IP over MPLS (IPoMPLS) is enabled in the provider core.

1. Re-enable the MPLS response default configuration on each LSR (R1, R2, R3, and R4

    ```
    device# configure
    device(config)# ip icmp mpls-response
    ```

2. On ExtremeCE1 (IP address 10.3.3.3), issue the **traceroute** command with the destination address of ExtremeCE2 (IP address 10.1.3.8).

```
device# traceroute 10.1.3.8
Type Control-c to abort
Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
1     <1 ms   <1 ms   <1 ms 10.51.3.7
2     <1 ms   <1 ms   <1 ms 10.56.1.2
       MPLS Label=1048 Exp=7 TTL=1 S=1
3     <1 ms   <1 ms   <1 ms 10.52.10.4
       MPLS Label=1026 Exp=7 TTL=1 S=1
4     <1 ms   <1 ms   <1 ms 10.34.22.8
5     <1 ms   <1 ms   <1 ms 10.1.3.8
```

The default IP over MPLS traceroute output shows a single label at hop 2 and at hop 3. Traceroute operates like the standard implementation except that the output is enhanced with label information. VPN labels are not used in this scenario. All ICMP error responses are routed as normal IP packets.

Scenario B – Layer 3 VPN over MPLS

MPLS is enabled in the provider core. Customer traffic is routed through the provider network using a Layer 3 VPN.

1. Re-enable the MPLS response default configuration on each LSR (R1, R2, R3, and R4).

```
device# configure
device(config)# ip icmp mpls-response
```

2. On ExtremeCE 1 (IP address 10.3.3.3), issue the **traceroute** command with the destination address of ExtremeCE2 (IP address 10.1.3.8).

```
device# traceroute 10.1.3.8
Type Control-c to abort
Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
1     <1 ms   <1 ms   <1 ms 10.51.3.7
2     *       *       *      ?
3     *       *       *      ?
4     *       *       *      ?
5     <1 ms   <1 ms   <1 ms 10.1.3.8
```

In this scenario, only PE1 and E2 return traceroute information. ICMP error messages generated at R2, R3, and PE2 with subsequent probes are dropped, because these LSRs can only use label-switching to transport the ICMP response and **use-lsp** is disabled in the ICMP response default configuration. Traceroute prints an asterisk (*) for each dropped package. The fifth traceroute probe makes it to ExtremeCE2, which routes an ICMP destination unreachable message back to ExtremeCE1 using IP routing table lookup. Refer to Limitations on page 184 for more information.

Tracing a route across an MPLS domain with label extensions using LSP

Assumptions: The default MPLS response feature is enabled on all provider routers. The user intends to trace the LSPs in the MPLS core and the user configures the routers to use LSPs. ExtremeCE1 is sending a traceroute to ExtremeCE2 with a destination IP address of 10.1.3.8. ICMP messages are generated with label stack extensions (always enabled unless explicitly disabled) and are propagated along configured LSPs. Only the PE routers have specific IP routes to the ExtremeCE routers.

Scenario A – IP over MPLS

IP over MPLS (IPoMPLS) is enabled in the provider core.

1. Issue the **ip icmp mpls-response** command with the **use-lsp** option on each LSR (R1, R2, R3, and R4).

    ```
    device# configure terminal
    device(config)# ip icmp mpls-response use-lsp
    ```

2. On the ExtremeCE 1 (IP address 10.3.3.3), issue the **traceroute** command with the destination address of ExtremeCE2 (IP address 10.1.3.8).

    ```
    deviceCE1# traceroute 10.1.3.8
    Type Control-c to abort
    Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
    1     <1 ms   <1 ms   <1 ms 10.51.3.7
    2     <1 ms   <1 ms   <1 ms 10.56.1.2
           MPLS Label=1048 Exp=7 TTL=1 S=1
    3     <1 ms   <1 ms   <1 ms 10.52.10.4
           MPLS Label=1026 Exp=7 TTL=1 S=1
    4     <1 ms   <1 ms   <1 ms 10.34.22.8
    5     <1 ms   <1 ms   <1 ms 10.1.3.8
    ```

    IP forwarding is enabled in the MPLS domain and the ICMP responses are forwarded using IP routes. Traceroute reports MPLS labels at hop 2 and 3 but no VPN label.

Scenario B – Layer 3 VPN over MPLS

MPLS is enabled in the provider core. Customer traffic is routed through the provider network using a Layer 3 VPN. The egress PE is a CER 2000 Series or a CES 2000 Series.

1. Issue the **ip icmp mpls-response** command with the **use-lsp** option on each LSR (R1, R2, R3, and R4).

    ```
    device# configure terminal
    device(config)# ip icmp mpls-response use-lsp
    ```

2. On the deviceCE 1 (IP address 10.3.3.3), issue the **traceroute** command with the destination address of deviceCE2 (IP address 10.1.3.8).

    ```
    device# traceroute 10.1.3.8
    Type Control-c to abort
    Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
    1     <1 ms   <1 ms   <1 ms 10.51.3.7
    2     <1 ms   <1 ms   <1 ms 10.56.1.2
          MPLS Label=1048 Exp=7 TTL=1 S=0
          MPLS Label=500000 Exp=7 TTL=1 S=1
    3     <1 ms   <1 ms   <1 ms 10.52.10.4
          MPLS Label=1026 Exp=7 TTL=1 S=0
          MPLS Label=500000 Exp=7 TTL=1 S=1
    4     <1 ms   <1 ms   <1 ms 10.34.22.8
    5     <1 ms   <1 ms   <1 ms 10.1.3.8
    ```

    In a Layer 3 VPN network, ICMP responses can only be forwarded as labeled MPLS packets along configured LSPs. Traceroute operates as described in section "IP Traceroute over MPLS on page 179, reporting both the outer path label and the VPN label used to route the packets to the appropriate VRF interface.

Tracing a route across an MPLS domain without label stack extensions and use-lsp disabled

Assumptions: The MPLS response default configuration is enabled on all LSRs in the provider network and the user wants to troubleshoot the IP routes. the user does not care about labels and the user configures traceroute to suppress label stack information. ExtremeCE1 is sending a traceroute to ExtremeCE2 with a destination IP address of 10.1.3.8. ICMP messages are generated without label stack extensions and forwarded per IP routing table lookup.

Scenario A - IP over MPLS

IP over MPLS (IPoMPLS) is enabled in the provider core.

1. Issue the **ip icmp mpls-response** command with the **no-label-extension** option on each LSR (R1, R2, R3, and R4).

   ```
   device# configure terminal
   device(config)#ip icmp mpls-response no-label-extension
   ```

2. On the deviceCE 1 (IP address 10.3.3.3), issue the **traceroute** command with the destination address of deviceCE2 (IP address 10.1.3.8).

   ```
   device# traceroute 10.1.3.8
   Type Control-c to abort
   Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
   1     <1 ms   <1 ms   <1 ms 10.51.3.7
   2     <1 ms   <1 ms   <1 ms 10.56.1.2
   3     <1 ms   <1 ms   <1 ms 10.52.10.4
   4     <1 ms   <1 ms   <1 ms 10.34.22.8
   5     <1 ms   <1 ms   <1 ms 10.1.3.8
   ```

   In this scenario, IP traceroute over MPLS behaves just like the standard **traceroute** command. At each hop, ICMP messages are generated and returned to the destination (source deviceCE1) as regular IP packets through standard IP routing protocols. The user gets the same traceroute behavior when the user disables the IP Traceroute over MPLS feature with the **no ip icmp mpls-response** command before sending a traceroute probe. When the ICMP response is disabled, the standard traceroute implementation is used.

Scenario B - Layer 3 VPN over MPLS

MPLS is enabled in the provider core. Customer traffic is routed through the provider network using a Layer 3 VPN. The egress PE is a CER 2000 Series or a CES 2000 Series.

1. Issue the **ip icmp mpls-response** command with the **no-label-extension** option on each LSR (R1, R2, R3, and R4).

   ```
   device# configure terminal
   device(config)# ip icmp mpls-response no-label-extension
   ```

2. On CE 1 (IP address 10.3.3.3), issue the **traceroute** command with the destination address of CE2 (IP address 10.1.3.8).

```
device# traceroute 10.1.3.8
Type Control-c to abort
Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
1    <1 ms   <1 ms   <1 ms 10.51.3.7
2     *       *       *      ?
3     *       *       *      ?
4     *       *       *      ?
5    <1 ms   <1 ms   <1 ms 10.1.3.8
```

The outcome is the same as if the user were using the default configuration (Scenario B – Layer 3 VPN over MPLS on page 185). Only PE1 and CE2 return traceroute information. ICMP error messages generated at R2, R3, and PE2 with subsequent probes are dropped, because these routers can only use LSPs to transport traffic and using LSP is blocked. Traceroute prints an asterisk (*) for each dropped package. The fifth traceroute probe makes it to C2, which routes a destination unreachable message back to CE1 using IP routing lookup. Refer to the Limitations on page 184 for more information.

Tracing a route across an MPLS domain with label extensions disabled and use-lsp enabled

Assumptions: MPLS is enabled in the provider core. Only the PEs have IP routes to the CEs. The MPLS response default configuration is enabled on all routers in the MPLS domain. The example specifies LSP forwarding but suppresses label stack information. CE1 is sending a traceroute to CE2 with a destination IP address of 10.1.3.8.

This example is included only to illustrate the CLI behavior. It is not useful for diagnosing LSP routing problems. Regardless of whether the user has IP over MPLS or a Layer 3 VPN configured, the provider transit router cannot propagate ICMP errors without label extensions when **use-lsp** is specified. For this reason, traceroute returns information only for the PE1 and CE2.

1. Issue the **ip icmp mpls-response** command with the **use-lsp** and the **no-label-extension** options on each LSR (R1, R2, R3, and R4).

```
device# configure terminal
device(config)# ip icmp mpls-response use-lsp no-label-extension
```

2. On the CE 1 (IP address 10.3.3.3), issue the **traceroute** command with the destination address of CE2 (IP address 10.1.3.8).

```
device# traceroute 10.1.3.8
Type Control-c to abort
Tracing the route to IP node (10.1.3.8) from 1 to 30 hops
1    <1 ms   <1 ms   <1 ms 10.51.3.7
2     *       *       *      ?
3     *       *       *      ?
4     *       *       *      ?
5    <1 ms   <1 ms   <1 ms 10.1.3.8
```

Understanding traceroute label information

When IP traceroute over MPLS is enabled on a provider LSR and configured to use label extensions, the **traceroute** command displays label information in its output.

Table 9 explains the output from the enhanced **traceroute** command.

TABLE 9 Output from the enhanced **traceroute** command

This Field…	Displays…
MPLS Label	The label appended to the ICMP ttl-exceeded message by the LSR that generated the message at the specified hop. The label is a 20-bit value.
Exp	A field in the MPLS label for experimental use. The field is a 3-bit value

TABLE 9 Output from the enhanced **traceroute** command (continued)

This Field...	Displays...
TTL	Time-to-live value of the datagram when it arrived at the LSR that generated the ICMP message. The TTL field is an 8-bit value.
S	Indicates the position of the label in the label stack The S field is a 1-bit value. A label with S=1 is located at the bottom of the stack. A label with a value of S=0 is not at the bottom of the stack.

MPLS LDP-IGP synchronization

Packet loss can occur because the actions of the IGP and LDP are not synchronized.

The MPLS LDP-IGP Synchronization feature provides the following benefits:

- Provides a means to synchronize LDP and IGPs to minimize MPLS packet loss
- MPLS LDP-IGP Synchronization may be enabled per interface, or globally
- OSPF and IS-IS are supported for the IGP; each operates independently
- LDP determines convergence (receipt of all labels) for a link by one of two methods
  - Receive Label silence mechanism
  - End Of Lib mechanism (*RFC 5919*)
- Provides a means to disable LDP-IGP Synchronization on interfaces that the user does not want enabled
- Enables the user to globally enable LDP-IGP synchronization on each interface associated with an IGP *Open Shortest Path First (OSPF)* or IS-IS process

FIGURE 43 Example with LDP IGP synchronization
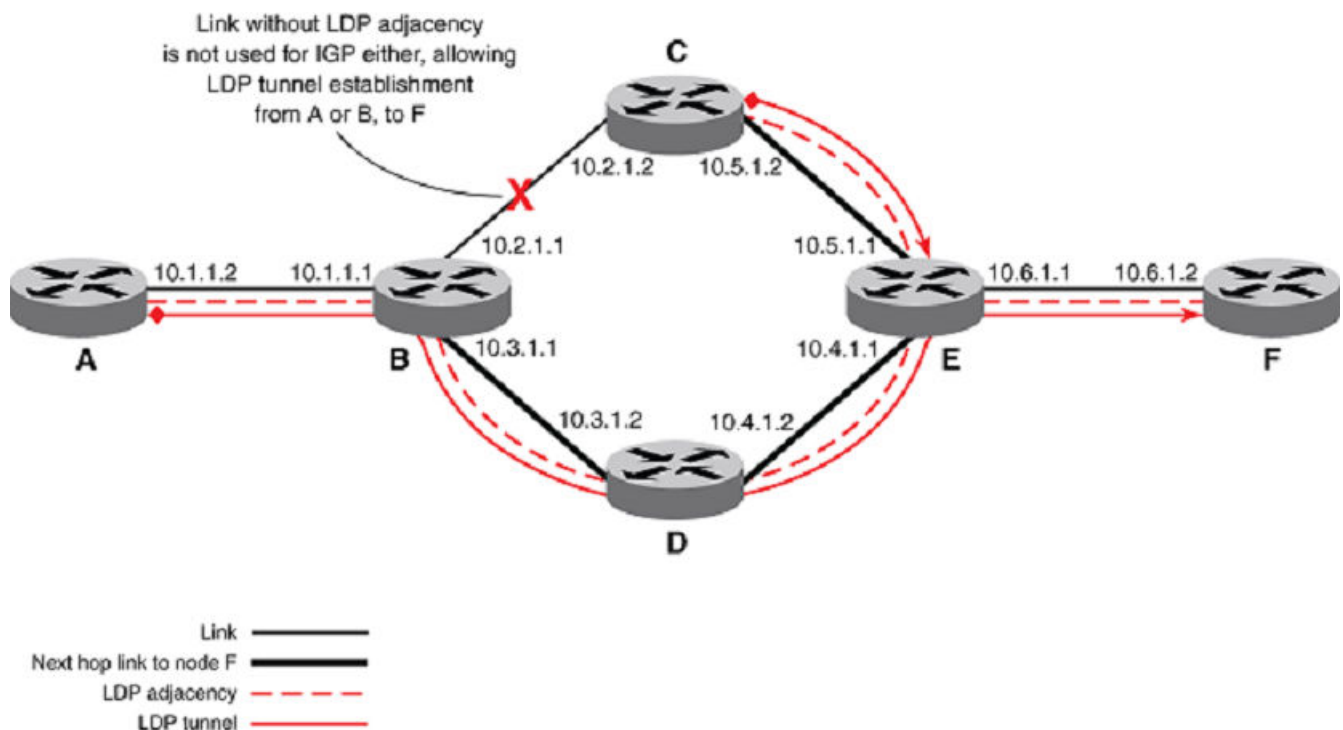
To enable LDP-IGP synchronization on each interface that belongs to an OSPF or IS-IS process, enter the **isis ldp sync** or the **ip ospf ldp sync** command under the corresponding mode. When the user does not want all of the interfaces to have LDP-IGP synchronization enabled, issue the [**no**] form of the command on the specific interfaces.

When the LDP peer is reachable, the IGP waits indefinitely (by default) for synchronization to be achieved. To limit the length of time the IGP session must wait, enter the **mpls ldp igp sync holddown** command. When the LDP peer is not reachable, the IGP establishes the adjacency to enable the LDP session to be established.

When an IGP adjacency is established on a link but LDP-IGP synchronization is not yet achieved or is lost, the IGP advertises the max-metric on that link.

Configuration considerations

- Supports only point-to-point interfaces.
- On IS-IS, wide metric-style is required.
- When enabled on IS-IS, the feature applies to both level-1 and level-2 metrics.
- Affects IPv4 metrics only.

Configuring MPLS LDP-IGP synchronization

This section contains the following tasks:

- Configuring MPLS LDP-IGP synchronization with OSPF interfaces (required).
- Selectively Disabling MPLS LDP-IGP synchronization from some OSPF interfaces (optional).
- Verifying MPLS LDP-IGP synchronization with OSPF (optional).

   NOTE
   Extreme recommends configuring the hold-down timer to more than 60 seconds to avoid traffic loss.

Configuring MPLS LDP-IGP synchronization globally

MPLS LDP-IGP synchronization is disabled by default. To globally enable MPLS LDP-IGP synchronization with IS-IS, enter the following commands:

```
device# configure
device(config)# router isis
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-router-ipv4u)# metric-style wide
device(config-isis-router-router-ipv4u)# ldp-sync
device(config-isis-router-router-ipv4u)# exit
device(config-isis-router)#
```

MPLS LDP-IGP synchronization is disabled by default. To globally enable MPLS LDP-IGP synchronization with OSPF, enter the following commands.

```
device# configure
device(config)# router ospf
device(config-router-ospf-vrf-default-vrf)# ldp-sync
device(config-router-ospf-vrf-default-vrf)#
```

Syntax: [no] ldp-sync

Setting the LDP IGP sync hold down time

The **ldp-sync hold-down** command sets the LDP-IGP sync hold down time. The hold down time (in router OSPF and the router IS-IS modes) is the interval which the IGP advertises the maximum IP metric, while waiting for an update from LDP.

The hold down interval starts whenever the IGP initially is enabled with LDP-IGP sync. It is also started whenever LDP updates the IGP with an update indicating the interface status, from LDP's perspective, is not-in-sync. When the hold down time expires, the IGP resumes advertising the normal metric for the link.

When hold down time is configured (from no hold down time), the router starts the hold-down-timer on every interface that is not-in-sync at the time.

When hold down time is un-configured, the router stops the hold-down-timer on every interface that has hold-down-timer running at the time as if there is no hold down time configured. As a result, these interfaces have infinite hold down time. For those not-in-sync interfaces with hold-down time already expired, IGP will continues to advertise normal metric.

By default, hold-down time is disabled. IGP waits until LDP gives an in-sync indication for the link before it advertised the normal metric.

By default, **ldp-sync hold-down** is disabled. To enable the **ldp-sync hold-down** timer with IS-IS, enter the following commands:

```
device# configure
device(config)# router isis
device(config-router-isis)# address-family ipv4 unicast
device(config-router-isis-ipv4u)# metric-style wide
device(config-router-isis-ipv4u)# ldp-sync
device(config-router-isis-ipv4u)# ldp-sync hold-down 100
```

By default, **ldp-sync hold-down** is disabled. To enable the **ldp-sync hold-down** timer with OSPF, enter the following commands:

```
device# configure
device(config)# router ospf
device(config-router-ospf-vrf-default-vrf)# ldp-sync
device(config-router-ospf-vrf-default-vrf)# ldp-sync hold-down 100
```

Syntax: ldp-sync *hold-down seconds*

The *seconds* parameter range is 1 to 65535 seconds.

Enabling LDP sync on an interface

Use the **isis ldp-sync** command under the **config-if-eth-1/1** policy to enable the LDP sync feature on a specific IS-IS interface. This overrides the global setting from the MPLS LDP-sync feature. By default, the **isis ldp-sync** is not enabled individually on an interface.

```
device# configure
device(config)# interface e 1/1
device(config-if-eth-1/1)# ip router isis
device(config-if-eth-1/1)# isis ldp-sync enable
```

Syntax: isis ldp-sync [**enable** | **disable**]

Use the **ip ospf ldp-sync** command under the **conf-if-e-1/1** policy to enable the LDP sync feature individually on an OSPF interface. By default, the **ip ospf ldp-sync** is not enabled individually on an OSPF interface.

```
device# configure
device(config)# interface ethernet 1/1
device(config-if-eth-1/1)# ip ospf area 0.0.0.0
device(config-if-eth-1/1)# ip ospf ldp-sync enable
```

Syntax: ip ospf ldp-sync [**enable** | **disable**]

Setting the receive label silence timer

When labels are not received from the peer for a short period of time, the session is declared 'In Sync'. When a label is received from a peer, then the 'receive label silence timer' is reset.

Use the **rx-label-silence-time** command under **config-mpls-ldp** policy to define the length of the receive label silence timer.

```
device(conf)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# rx-label-silence-time 30000
```

Syntax: rx-label-silence-time *value*

The *value* parameter specifies the length of time of the receive label silence timer in milliseconds. Possible values are from 100 to 60000 milliseconds. The default value is 1000.

Enabling the end-of-lib submode

Configure the **end-of-lib** sub-mode under LDP to contain all the attributes of the end of lib capability and notification.

```
device(conf)# router mpls
device(conf-router-mpls)# ldp
device(conf-router-mpls-ldp)# end-of-lib
device(conf-router-mpls-ldp-eol)#
Disabling the end-of-lib submode
```

Enabling the **end-of-lib** sub-mode determines whether the two RFCs, *RFC 5561* and *RFC 5919* are enabled by the LSR. The user can turn this feature off either by:

- Removing the **end-of-lib** sub-mode.
- Issuing the **disable** command under the end-of-lib sub-mode.

```
device(conf)# router mpls
device(conf-router-mpls)# ldp
device(conf-router-mpls-ldp)# end-of-lib
device(conf-router-mpls-ldp-eol)# disable
```

Syntax: [no] end-of-lib

The [**no**] form of "disable" enables the feature.

Setting the EOL notification timer

Use the **EOL notification timer** command under the **conf-router-mpls-ldp-eol** policy to set the length of the EOL notification timer. This command is LDP global.

```
device(conf)# router mpls
device(conf-router-mpls)# ldp
device(conf-router-mpls-ldp)# end-of-lib
device(conf-router-mpls-ldp-eol)# notification-timer value
```

Syntax: EOL notification timer *value*

The *value* parameter specifies the length of the EOL notification timer in milliseconds. Possible values are from 100 to 120000 milliseconds. The default value is 60000.

Setting the EOL transmit label silence timer

Use the **tx-label-silence-timer** command under **conf-router-mpls-ldp-eol** policy to sets the length of the EOL transmit label silence timer. This command is LDP global.

```
device(config)# router mpls
device(config-router-mpls)# ldp
device(config-router-mpls-ldp)# end-of-lib
device(config-router-mpls-ldp-eol)# tx-label-silence-timer 2000
```

Syntax: tx-label-silence-timer *value*

The *value* parameter specifies the length of the EOL transmit label silence timer in milliseconds. Possible values are from 100 to 60000 milliseconds. The default value is 1000.

Displaying MPLS and RSVP information

The user can display the following information about the MPLS configuration on the device:

- Information about MPLS-enabled interfaces on the device
- Statistics about the MPLS-enabled interfaces
- MPLS summary information
- Contents of the *Traffic Engineering Database (TED)*
- Status information about signaled LSPs configured on the device
- Information about paths configured on the device
- The label applied at each hop in an LSP
- Contents of the MPLS routing table
- RSVP information, including the status of RSVP-enabled interfaces, session information, and statistics
- MPLS Fast Reroute Information

Transit LSP statistics

Transit LSP statistics extends the ability to collect and display traffic statistics on a transit router for both LDP and RSVP.

MLX Series and XMR Series limitations

> **NOTE**
> Not all interface modules support transit LSP statistics. Gen-2 interface modules that do support packet count, byte count and rate include the 24x10G, 2x100G, and 8x10G modules.

Refer to Table 10 for additional interface module and packet count support.

TABLE 10 Transit LSP module support

Interface module	Feature support		
Packet count	Byte count	Rate (kbps)	
NI-MLX-1GX20-GC	X		
NI-XMR-1Gx20-GC	X		

TABLE 10 Transit LSP module support (continued)

Interface module	Feature support		
NI-XMR-10Gx4	X		
NI-MLX-10GX4	X		
BR-MLX-10GX4-X	X		
BR-MLX-10Gx4-X-ML	X		
NI-MLX-10GX8-M	X	X	X
NI-MLX-10GX8-D	X	X	X
BR-MLX-10GX8-X	X	X	X
BR-MLX-10Gx24-DM	X	X	X
BR-MLX-100GX-1	X	X	X
BR-MLX-100GX-2	X	X	X

CES 2000 Series and CER 2000 Series limitations

LDP LSP statistic collection is not supported; only RSVP LSP transit statistics is supported.

Either Packet or (Byte and Rate) are supported, but not both simultaneously.

The number of counters available to collect the statistics is limited to 2K even though the CES 2000 Series and CER 2000 Series support 4000 transit sessions.

Clearing MPLS statistics

TABLE 11 Output from the **mpls statistics ldp transit** command

This field...	Displays...
FEC	The specified FEC for MPLS LDP transit statistics.
Packets	Specifies the number of packets received.
Bytes	Specifies the number of bytes received.
Rate-kbps	Rate is in kilobits per second.

The following sections describe the commands used to clear MPLS statistics.

Clearing MPLS RSVP sessions

To clear only the RSVP statistics transit counters, enter the following command:

```
device# clear mpls statistics rsvp session 10.2.2.2 10.1.1.1 5
```

Syntax: **clear mpls statistics rsvp session** [*destination ip address* | *source ip address* | *tunnel id*]

Where the *destination ip address* specifies the destination IP address of session object.

Where the *source ip address* specifies the source IP address of session object.

Where the *tunnel id* specifies the tunnel ID of session object.

Clearing MPLS LDP transit statistics

To clear the transit traffic statistics for the specified LDP FEC, enter the following command.

```
device# clear mpls statistics ldp transit fec 10.3.3.3/32
```

Syntax: **clear mpls statistics ldp transit fec** [*ipaddress/subnet mask length*]

Where the *ip address* is the IP address configured on this interface.

Where the *subnet mask length* specifies the network prefix mask length.

Clearing MPLS label statistics

To clear the traffic statistics for the specified incoming label, enter a command such as the following.

```
device# clear mpls statistics label 2032
```

Syntax: **clear mpls statistics label** [*in-label*]

The *in-label* variable allows the user to specify the label value.

Sample configurations

To set the counter-mode to packet on the CES 2000 Series or CER 2000 Series, enter the following commands.

```
device# configure
device(config)# router mpls
device(config-mpls)# counter-mode
device(config-mpls)# counter-mode packet
```

To verify the configuration, enter the following command.

```
device(config-mpls)# show mpls configuration
router mpls
 counter-mode packet
mpls-interface e1/11
end of MPLS configuration
```

To reset counter-mode to byte, enter the following command.

```
device(config-mpls)# no counter-mode packet
```

To verify the configuration, enter the following command.

```
device(config-mpls)# show mpls conf
router mpls
mpls-interface e1/11
end of MPLS configuration
```

MPLS sample configurations

This section presents examples of typical MPLS configurations.

LSP with redundant paths

Figure 44 shows a signaled LSP configuration that has a primary and a secondary path. The destination for this LSP is 10.1.1.1. The primary path to this destination is through interface e 2/2, which has a direct link to interface e 2/2 on R3. When this link fails, the secondary path is established. The secondary path goes through R2.

FIGURE 44 LSP configuration with primary and secondary paths
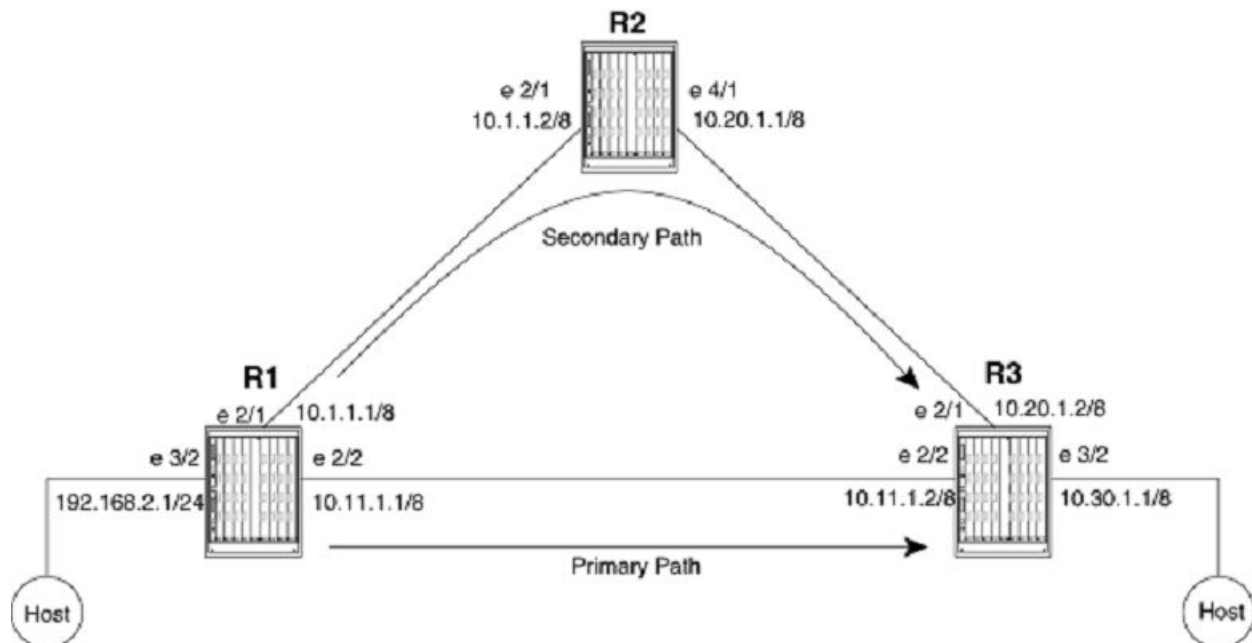
Router R1 is the ingress LER for signaled LSP t3. Packets whose destination is 10.1.1.1 are assigned to this LSP. Two paths are configured, direct_conn and via_r2. Path direct_conn consists of a single strict node, 10.1.1.2, which is a directly connected interface on the destination LSR, R3.

Path via_r2 also consists of a single strict node, 10.1.1.2, a directly connected interface on R2. Since path via_r2 does not specify a node for R3, the hop between R2 and R3 is treated as a hop to a loose node. This means standard hop-by-hop routing is used to determine the path between R2 and R3.

Path direct_conn is the primary path for LSP t3, and path via_r2 is the secondary path. When the LSP is enabled, RSVP signaling messages set up path direct_conn. Packets assigned to this LSP use this path to reach the destination.

When path direct_conn fails, path via_r2 is set up, and packets assigned to LSP t3 then use path via_r2 to reach the destination. By default, the secondary path is not set up until the primary path fails. When the user employs the **standby** parameter in the configuration of the secondary path, both the primary and secondary paths are set up at the same time, although packets assigned to the LSP travel on the primary path only. When the primary path fails, the secondary path immediately carries the traffic.

Router device1

The following commands configure Router device1 in

```
device1# configure
device1(config)# router mpls
device1(config-router-mpls)# mpls-interface e 2/1 e 2/2
device1(config-router-mpls)# path direct_conn
device1(config-router-mpls-path)# strict 10.11.1.2
device1(config-router-mpls-path)# exit
device1(config-router-mpls)# path via_r2
device1(config-router-mpls-path)# strict 10.1.1.2
device1(config-router-mpls-path)# exit
device1(config-router-mpls)# lsp t3
device1(config-router-mpls-lsp-t3)# to 10.30.1.1
device1(config-router-mpls-lsp-t3)# primary direct
device1(config-router-mpls-lsp-t3)# secondary via_r2
device1(config-router-mpls-lsp-t3)# enable
```

```
device1(config-router-mpls-lsp-t3)# exit
device1(config-router-mpls)# interface ethernet 2/1
device1(config-router-mpls-eth-2/1)# ip address 10.1.1.1 255.0.0.0
device1(config-router-mpls-eth-2/1)# exit
device1(config-router-mpls)# interface ethernet 2/2
device1(config-router-mpls-eth-2/2)# ip address 10.11.1.1 255.0.0.0
device1(config-router-mpls-eth-2/2)# exit
device1(config-router-mpls)# interface ethernet 3/2
device1(config-router-mpls-if-eth-3/2)# ip address 192.168.2.1 255.255.255.0
device1(config-router-mpls-if-eth-3/2)# exit
```

Router device2

In the configuration in LSP with redundant paths on page 195, Router device2 serves as a transit LSR for path via_r2. Since path via_r2 is the secondary path for the LSP, it is used only when the primary path fails.

```
device2# configure
device2(config)# router mpls
device2(config-router-mpls)# mpls-interface ethernet 2/1 ethernet 4/1
device2(config-router-mpls)# interface ethernet 2/1
device2(config-router-mpls-eth-2/1)# ip address 10.1.1.2 255.0.0.0
device2(config-router-mpls-eth-2/1)# exit
device2(config-router-mpls)# interface ethernet 4/1
device2(config-router-mpls-eth-4/1)# ip address 10.20.1.1 255.0.0.0
device2(config-router-mpls-eth-4/1)# exit
```

Router device3

In the configuration in LSP with redundant paths on page 195, Router device3 is the egress LER for LSP t3.

```
device3# configure
device3(config)# router mpls
device3(config-router-mpls)# mpls-interface ethernet 2/1 ethernet 2/2
device3(config-router-mpls)# interface pos 2/1
device3(config-router-mpls-eth-2/1)# ip address 10.20.1.2 255.0.0.0
device3(config-router-mpls-eth-2/1)# exit
device3(config-router-mpls)# interface ethernet 2/2
device3(config-router-mpls-eth-2/2)# ip address 10.11.1.2 255.0.0.0
device3(config-router-mpls-eth-2/2)# exit
device3(config-router-mpls)# interface ethernet 3/2
device3(config-router-mpls-if-eth-3/2)# ip address 10.30.1.1 255.0.0.0
device3(config-router-mpls-if-eth-3/2)# exit
```

Example of MPLS Fast Reroute configuration

This example describes an MPLS Fast Reroute Loop Configuration. It provides the configuration required on Ingress Router 4 and examples of the **show mpls rsvp session** displays for all of the routers in the configuration. These examples show how the MPLS Fast Reroute configuration of an LSP affects the RSVP session on each of the routers in the configuration.

As illustrated in Figure 45 and described in the configuration example that follows, Ingress Router 4 is configured with a strict Label Switch Path A to Egress Router 5. In this configuration, when the path is broken between Ingress Router 4 and Egress Router 5, Transit Routers 2 or 6 take a detour path back through Ingress Router 4 and continue through Transit Routers 3 and 1 to reach Egress Router 5.

FIGURE 45 MPLS Fast Reroute Loop configuration
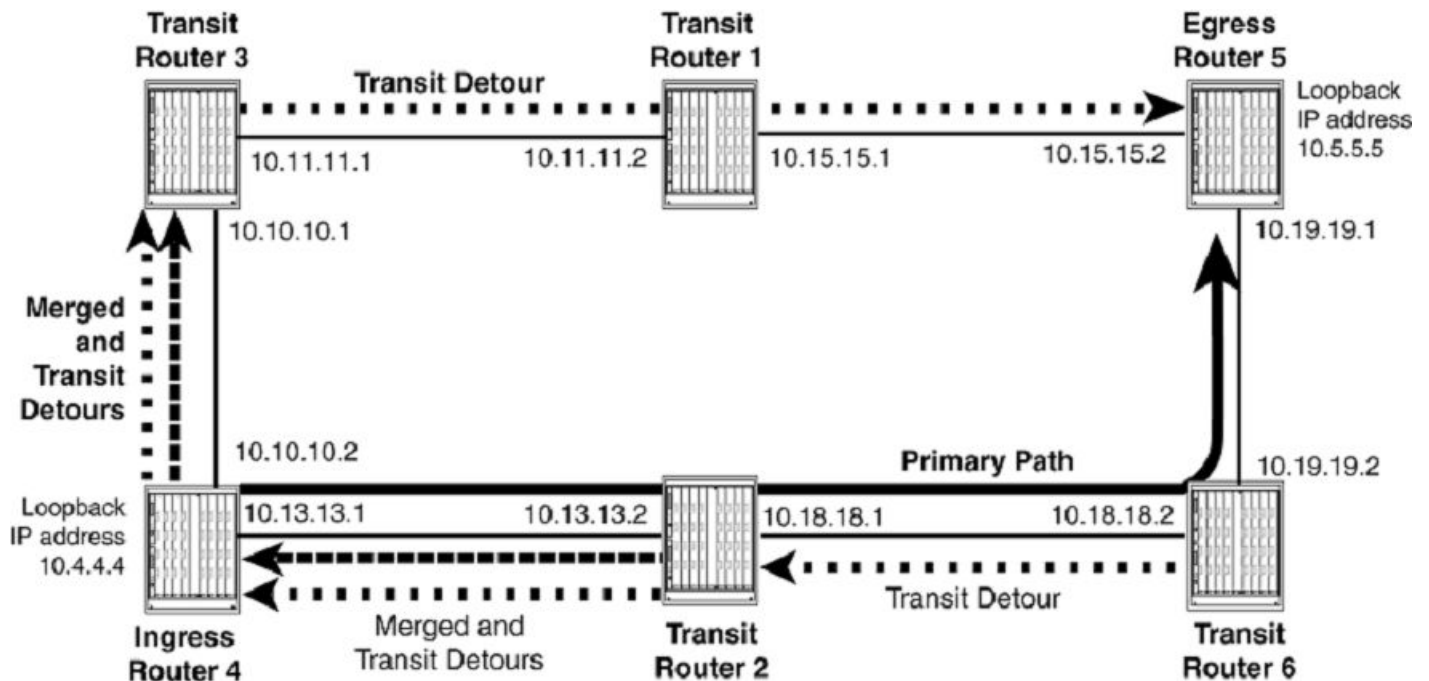
The following is the MPLS Fast Reroute configuration for Ingress Router 4.

```
device(config)# interface loopback 1
device4(config-Loopback-1)# ip address 10.4.4.4/24
device4(config)# interface ethernet 2/1
device4(config-if-eth-2/1)# ip address 10.10.10.2/24
device4(config)# interface ethernet 2/9
device4(config-if-eth-2/9)# ip address 10.13.13.1/24
device4(config)# router mpls
device4(config-router-mpls)# mpls-interface ethernet 2/1 ethernet 2/9
device4(config-router-mpls-if-eth-2/1-eth-2/9)# path a
device4(config-router-mpls-path-a)#hop 10.2.2.2 strict
device4(config-router-mpls-path-a)# hop 10.6.6.6 strict
device4(config-router-mpls)# lsp 1
device4(config-router-mpls-lsp-1)# to 10.5.5.5
device4(config-router-mpls-lsp-1)# primary-path a
device4(config-router-mpls-lsp-1)# frr
```

Displaying RSVP session information for example network

The **show mpls rsvp session** command, provides information regarding the primary and detour routes in an MPLS RSVP Fast Reroute enabled network. Display examples are provided for the following routers in the configuration shown in Example of MPLS Fast Reroute configuration on page 197:

- Transit Router 6
- Transit Router 2
- Ingress Router 4
- Transit Router 3
- Transit Router 1
- Egress Router 5

The following examples include displays for the **show mpls rsvp session** and **show mpls rsvp session detail** commands. For a general description of the command and its output refer to

The Transit Router 1 display

The following display examples are from Transit Router 1. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

The display for Transit Router 1 shows a *Transit Detour (DT)* path. The DT path is the detour path that is an alternative to the primary path configured on Ingress Router 4 from Ingress Router 4 to Egress Router 5. This detour path shown from Ingress Router 4 at Loopback IP address 10.4.4.4 to Egress Router 5 at Loopback IP address 10.5.5.5 is only used when there is a failed link or router between the source and destination of the primary path.

```
device1# show mpls rsvp session
Codes: DI:Ingress Detour   DT:Transit Detour   DM:Merged Detour
       DE:Egress Detour   RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To               From             St     Style Lbl_in Lbl_out LSPname
10.5.5.5         10.4.4.4(DT)     Up     SE    1028   3       1
Egress RSVP:      0 session(s)
```

The following example displays the output from Transit Router 1 using the **show mpls rsvp session detail** command. This option provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DT path, the "Explicit path hop count" field indicates that there is one hop from this router to the egress of the path at the router at IP address 10.15.15.2 (Egress Router 5).

```
device3# show mpls rsvp session detail
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To               From             St     Style Lbl_in Lbl_out LSPname
10.5.5.5         10.4.4.4(DT)     Up     SE    1028   3       1
  Time left in seconds (PATH refresh: 18, ttd: 146
                        RESV refresh: 15, ttd: 154)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Explicit path hop count: 1
   10.15.15.2 (S)
  Received RRO count: 1
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.15.15.2
  PATH rcvfrom: 10.11.11.1      (e7/16         ) (MD5 OFF)
  PATH sentto:  10.15.15.2      (e6/2          ) (MD5 OFF)
  RESV rcvfrom: 10.15.15.2      (e6/2          ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

The Transit Router 2 display

The following display examples are from Transit Router 2. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

As for Transit Router 2, the displays show an *Ingress Detour (DI)* path, and a path without a code which identifies a protected path. In addition, a *Merged Detour (DM)* path is shown. The DM path is the detour path merged from Transit Router 6. All three paths are shown from Ingress Router 4 at Loopback IP address 10.4.4.4 to Egress Router 5 at Loopback IP address 10.5.5.5.

```
device1# show mpls rsvp session
Codes: DI:Ingress Detour   DT:Transit Detour   DM:Merged Detour
       DE:Egress Detour   RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To               From             St   Style Lbl_in Lbl_out LSPname
10.5.5.5         10.4.4.4(DI)     Up   SE    1024   1024    1
10.5.5.5         10.4.4.4         Up   SE    1024   1024    1
```

```
10.5.5.5          10.4.4.4(DM)      Up  SE    1025   1024    1
Egress RSVP:       0 session(s)
```

The following example displays the output from Transit Router 2 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DI path, two PLR and Avoid Node ID pairs are shown labeled [1] and [2]. In [1] the *Point of Local Repair (PLR)* is at IP Address 10.18.18.1 which is the interface to the primary path on Transit Router 2 and the Avoid Node is IP address 10.18.18.2 on Transit Router 6. In [2] the *Point of Local Repair (PLR)* is at IP Address 10.19.19.2 on Transit Router 6 and the Avoid Node is IP address 0.0.0.0. The "Explicit path hop count" field indicates that there are four hops on this path from this router to the egress of the path at routers with the following IP addresses 10.13.13.1 (Ingress Router 4), 10.10.10.1 (Transit Router 3), 10.11.11.2 (Transit Router 1) and 10.15.15.2 (Egress Router 5).

For the DM path, one PLR and Avoid Node ID pair is shown labeled [1]. In [1] the *Point of Local Repair (PLR)* is at IP Address 10.19.19.2 which is an interface on Transit Router 6 and the Avoid Node is IP address 0.0.0.0.

For the primary path, the "Explicit path hop count" field indicates that the path has two hops from this router to the egress to the path at routers with the following IP addresses 10.18.18.2 (Transit Router 6) and 10.19.19.1 (Egress Router 5). The 'Fast Reroute' field indicates that the primary path has been configured for one-to-one backup.

```
device2# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From         St    Style Lbl_in Lbl_out LSPname
10.5.5.5        10.4.4.4(DI)    Up   SE    1024   1024    1
  Time left in seconds (PATH refresh: 1, ttd: 4293570
                      RESV refresh: 13, ttd: 141)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Explicit path hop count: 4
   10.13.13.1 (S) -> 10.10.10.1 (S) -> 10.11.11.2 (S) -> 10.15.15.2 (S)
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.13.13.1 -> 10.10.10.1 -> 10.11.11.2 ->
    10.15.15.2
  Detour Sent: Number of PLR and Avoid Node ID pair(s): 2
   [1]: PLR: 10.18.18.1  Avoid Node: 10.18.18.2
   [2]: PLR: 10.19.19.2  Avoid Node: 0.0.0.0
  PATH sentto: 10.13.13.1       (e5/10          ) (MD5 OFF)
  RESV rcvfrom: 10.13.13.1      (e5/10          ) (MD5 OFF)
To              From         St    Style Lbl_in Lbl_out LSPname
10.5.5.5        10.4.4.4       Up   SE    1024   1024    1
  Time left in seconds (PATH refresh: 26, ttd: 151
                      RESV refresh: 13, ttd: 151)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Fast Reroute: one-to-one backup desired
   Setup priority: 7, hold priority: 0
   Bandwidth: 0 kbps, hop limit: 255
   Detour LSP: UP.  Nexthop (node) protection available.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 2
   10.18.18.2 (S) -> 10.19.19.1 (S)
  Received RRO count: 2
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.18.18.2 (PN) -> 10.19.19.1
  PATH rcvfrom: 10.13.13.1      (e5/10          ) (MD5 OFF)
  PATH sentto:  10.18.18.2      (e2/1           ) (MD5 OFF)
  RESV rcvfrom: 10.18.18.2      (e2/1           ) (MD5 OFF)
To              From              St    Style Lbl_in  Lbl_out  LSPname
10.5.5.5          10.4.4.4(DM)     Up   SE    1025    1024     1
  Time left in seconds (PATH refresh: 31, ttd: 133
                      RESV refresh: 13, ttd: 141)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.13.13.1 -> 10.10.10.1 -> 10.11.11.2 ->
```

```
   10.15.15.2
 Detour Rcvd: Number of PLR and Avoid Node ID pair(s): 1
  [1]: PLR: 10.19.19.2  Avoid Node: 0.0.0.0
 PATH rcvfrom: 10.18.18.2      (e2/1           ) (MD5 OFF)
 RESV rcvfrom: 10.13.13.1      (e5/10          ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

The Transit Router 3 display

The following display examples come from Transit Router 3. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

The display for Transit Router 3 shows a *Transit Detour (DT)* path. The DT path is the detour path that is an alternative to the primary path configured on Ingress Router 4 from itself to Egress Router 5. This detour path, shown from Ingress Router 4 at Loopback IP address 10.4.4.4 to Egress Router 5 at Loopback IP address 10.5.5.5, is only used when a link or router fails between the source and destination of the primary path.

```
device3# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From             St    Style Lbl_in Lbl_out LSPname
10.5.5.5        10.4.4.4(DT)     Up    SE    1028   1028    1
Egress RSVP:      0 session(s)
```

The following example displays the output from Transit Router 3 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DT path, the "Explicit path hop count:" field shows that two hops exist from this router to the egress of the path at routers with the following at IP addresses: 10.11.11.2 (Transit Router 1) and 10.15.15.2 (Egress Router 5).

```
device3# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To              From             St    Style Lbl_in Lbl_out LSPname
10.5.5.5        10.4.4.4(DT)     Up    SE    1028   1028    1
  Time left in seconds (PATH refresh: 2, ttd: 141
                        RESV refresh: 25, ttd: 154)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Explicit path hop count: 2
   10.11.11.2 (S) -> 10.15.15.2 (S)
  Received RRO count: 2
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.11.11.2 -> 10.15.15.2
  PATH rcvfrom: 10.10.10.2      (e12/1          ) (MD5 OFF)
  PATH sentto:  10.11.11.2      (e11/18         ) (MD5 OFF)
  RESV rcvfrom: 10.11.11.2      (e11/18         ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

The Ingress Router 4 display

The following display examples are from Ingress Router 4. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

Like the display for Transit Router 2, the displays show an *Ingress Detour (DI)* path, a path without a code which identifies a protected path and *Merged Detour (DM)* path. The DM path is the detour path merged from Transit Routers 2 and 6. All three paths are shown from Ingress Router 4 at IP address Loopback 10.4.4.4 to Egress Router 5 at IP address Loopback 10.5.5.5. In the case of the DM

path, a reroute at either Transit Router 2 or 6 sends traffic that had begun at Ingress Router 4 back through it, and forward through Transit Routers 3 and 1 to the ultimate destination at Egress Router 5.

```
device4# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     1 session(s)
To                 From          St   Style Lbl_in Lbl_out LSPname
10.5.5.5           10.4.4.4(DI)  Up   SE    -      1028    1
10.5.5.5           10.4.4.4(DM)  Up   SE    1024   1028    1
10.5.5.5           10.4.4.4      Up   SE    -      1024    1
Transit RSVP:     0 session(s)
Egress RSVP:      0 session(s)
```

The following example displays the **show mpls rsvp session detail** output for Ingress Router 4. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DI path, three PLR and Avoid Node ID pairs are shown labeled [1], [2] and [3]. In [1] and [2] the *Point of Local Repair (PLR)* is at IP Address 10.13.13.1 which is the interface to the primary path on Ingress Router 4. The Avoid Node for pair [1] is IP address 10.13.13.2 on Transit Router 2 and the Avoid Node for pair [2] is IP address 10.18.18.2 on Transit Router 6. In [3] the *Point of Local Repair (PLR)* is at IP Address 10.18.18.1 which is the interface to the primary path on Ingress Router 2 and the Avoid Node for pair [3] is IP address 10.18.18.2 on Transit Router 6. The "Explicit path hop count" field indicates that there are three hops on the path from this router to the egress of the path at routers with the following IP addresses 10.10.10.1 (Transit Router 3), 10.11.11.2 (Transit Router 1) and 10.15.15.2 (Egress Router 5).

For the DM path, one PLR and Avoid Node ID pair is shown labeled [1]. In [1] the *Point of Local Repair (PLR)* is at IP Address 10.18.18.1 which is the interface to the primary path on Ingress Router 2 and the Avoid Node is IP address 10.18.18.2 on Transit Router 6.

For the primary path, the "Explicit path hop count" field indicates that there are three hops on this path from Ingress Router 4 to the egress to the path at routers with the following IP addresses 10.13.13.2 (Transit Router 2), 10.18.18.2 (Transit Router 6) and 10.19.19.1 (Egress Router 5). The "Fast Reroute" field indicates that the primary path has been configured for one-to-one backup.

```
device4# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     1 session(s)
To                 From          St   Style Lbl_in Lbl_out LSPname
10.5.5.5           10.4.4.4(DI)  Up   SE    -      1028n   1
  Time left in seconds (PATH refresh: 16, ttd: 4293608
                        RESV refresh: 27, ttd: 133)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 3
   10.10.10.1 (S) -> 10.11.11.2 (S) -> 10.15.15.2 (S)
  Received RRO count: 3
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.10.10.1 -> 10.11.11.2 -> 10.15.15.2
  Detour Sent: Number of PLR and Avoid Node ID pair(s): 3
   [1]: PLR: 10.13.13.1  Avoid Node: 10.13.13.2
   [2]: PLR: 10.13.13.1  Avoid Node: 10.18.18.2
   [3]: PLR: 10.18.18.1  Avoid Node: 10.18.18.2
  PATH sentto:  10.10.10.1      (e2/1          ) (MD5 OFF)
  RESV rcvfrom: 10.10.10.1      (e2/1          ) (MD5 OFF)
To                 From          St   Style Lbl_in Lbl_out LSPname
10.5.5.5           10.4.4.4(DM)  Up   SE    1024   1028    1
  Time left in seconds (PATH refresh: 6, ttd: 134
                        RESV refresh: 27, ttd: 133)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Received RRO count: 3
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.10.10.1 -> 10.11.11.2 -> 10.15.15.2
  Detour Rcvd: Number of PLR and Avoid Node ID pair(s): 1
   [1]: PLR: 10.18.18.1  Avoid Node: 10.18.18.2
  PATH rcvfrom: 10.13.13.2      (e2/20         ) (MD5 OFF)
  RESV rcvfrom: 10.10.10.1      (e2/1          ) (MD5 OFF)
```

```
To                From            St    Style Lbl_in Lbl_out LSPname
10.5.5.5          10.4.4.4        Up    SE    -      1024    1
  Time left in seconds (PATH refresh: 37, ttd: 148
                       RESV refresh: 27, ttd: 152)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Fast Reroute: one-to-one backup desired
   Setup priority: 7, hold priority: 0
   Bandwidth: 0 kbps, hop limit: 255
   Detour LSP: UP.  Nexthop (node) protection available.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 3
   10.13.13.2 (S) -> 10.18.18.2 (S) -> 10.19.19.1 (S)
  Received RRO count: 3
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.13.13.2 (PN) -> 10.18.18.2 (PN) -> 10.19.19.1
  PATH sentto:  10.13.13.2      (e2/20        ) (MD5 OFF)
  RESV rcvfrom: 10.13.13.2      (e2/20        ) (MD5 OFF)
Transit RSVP:     0 session(s)
Egress RSVP:      0 session(s)
```

The Egress Router 5 display

The following display examples are from Egress Router 5. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

The display for Egress Router 5, shows an *Egress Detour (DE)* path and a path without a code that identifies a protected path. Both paths are shown from Ingress Router 4 at Loopback IP address 10.4.4.4 to Egress Router 5 at Loopback IP address 10.5.5.5. The primary path traverses Transit Routers 2 and 6. In the case of the DE path, a reroute sends traffic through Transit Routers 3 and 1.

```
device5# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     0 session(s)
Egress RSVP:      1 session(s)
To                From            St    Style Lbl_in Lbl_out LSPname
10.5.5.5          10.4.4.4(DE)    Up    SE    3      0       1
10.5.5.5          10.4.4.4        Up    SE    3      0       1
```

The following example displays the output from Egress Router 5 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DE path, two PLR and Avoid Node ID pairs are shown labeled [1] and [2]. In both the *Point of Local Repair (PLR)* is at IP Address 10.13.13.1 which is the interface to the primary path on Ingress Router 4. The Avoid Node for pair [1] is IP address 10.13.13.2 on Transit Router 2 and the Avoid Node for pair [2] is IP address 10.18.18.2 on Transit Router 6.

The "Fast Reroute" field indicates that the primary path has been configured for one-to-one backup.

There is no "Explicit path hop count" field for either route because Egress Router 5 is the destination of the path.

```
device5# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     0 session(s)
Egress RSVP:      1 session(s)
To                From            St    Style Lbl_in Lbl_out LSPname
10.5.5.5          10.4.4.4(DE)    Up    SE    3      0       1
  Time left in seconds (PATH refresh: 18, ttd: 149
                       RESV refresh: 7, ttd: 152)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
  Detour Rcvd: Number of PLR and Avoid Node ID pair(s): 2
   [1]: PLR: 10.13.13.1  Avoid Node: 10.13.13.2
   [2]: PLR: 10.13.13.1  Avoid Node: 10.18.18.2
  PATH rcvfrom: 10.15.15.1      (e8/2         ) (MD5 OFF)
To                From            St    Styl Lbl_in Lbl_outm LSPname
10.5.5.5          10.4.4.4        Up    SE    3      0       1
```

```
      Time left in seconds (PATH refresh: 30, ttd: 152
                           RESV refresh: 7, ttd: 152)
      Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
      Fast Reroute: one-to-one backup desired
       Setup priority: 7, hold priority: 0
       Bandwidth: 0 kbps, hop limit: 255

   PATH rcvfrom: 10.19.19.2      (e8/1) (MD5 OFF)
```

The Transit Router 6 display

The following display examples are from Transit Router 6. Displays are shown for the **show mpls rsvp session** and **show mpls rsvp session detail** commands.

Both displays show two paths from Ingress Router 4 at Loopback IP address 10.4.4.4. to Egress Router 5 at Loopback IP address 10.5.5.5. The (DI) path is an Ingress Detour path, and the path without a code is a protected path. The DI path is the detour path that is taken when Transit Router 6 is unable to use the primary path to Egress Router 5.

```
device6# show mpls rsvp session
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To               From            St   Style Lbl_in Lbl_out LSPname
10.5.5.5         10.4.4.4(DI)    Up   SE    1024   1025    1
10.5.5.5         10.4.4.4        Up   SE    1024   3       1
Egress RSVP:      0 session(s)
```

The following example displays the output from Transit Router 6 using the **show mpls rsvp session detail** command. This command provides additional details about the paths described in the output from the **show mpls rsvp session** command.

For the DI path, one PLR and Avoid Node ID pair is shown labeled [1]. In [1], the Point of Local Repair (PLR) is at IP Address 10.19.19.2 which is an interface on Transit Router 6 and the Avoid Node is IP address 0.0.0.0. The "Explicit path hop count" field indicates that there are five hops on this path from this router to the egress to the path at routers with the following IP addresses 10.18.18.1 (Transit Router 2), 10.13.13.1 (Ingress Router 4), 10.10.10.1 (Transit Router 3), 10.11.11.2 (Transit Router 1) and 10.15.15.2 (Egress Router 5).

For the primary path, the "Explicit path hop count" field indicates that there is one hop on this path from this router to the egress to the path to the router at IP address 10.19.19.1 (Egress Router 5). The "Fast Reroute" field indicates that the primary path has been configured for one-to-one backup.

```
device6# show mpls rsvp session detail
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  RP:Repaired Session
Ingress RSVP:     0 session(s)
Transit RSVP:     1 session(s)
To               From            St   Style Lbl_in Lbl_out LSPname
10.5.5.5         10.4.4.4(DI)    Up   SE    1024   1025    1
   Time left in seconds (PATH refresh: 6, ttd: 4293497
                        RESV refresh: 24, ttd: 131)
   Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
   Explicit path hop count: 5
    10.18.18.1 (S) -> 10.13.13.1 (S) -> 10.10.10.1 (S) -> 10.11.11.2 (S) ->
     10.15.15.2 (S)
   Received RRO count: 5
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.18.18.1 -> 10.13.13.1 -> 10.10.10.1 ->
     10.11.11.2 -> 10.15.15.2
   Detour Sent: Number of PLR and Avoid Node ID pair(s): 1
    [1]: PLR: 10.19.19.2  Avoid Node: 0.0.0.0
   PATH sentto:  10.18.18.1      (e5/1          ) (MD5 OFF)
   RESV rcvfrom: 10.18.18.1      (e5/1          ) (MD5 OFF)
To               From            St   Style Lbl_in Lbl_out LSPname
10.5.5.5         10.4.4.4        Up   SE    1024   3       1
   Time left in seconds (PATH refresh: 28, ttd: 150
```

```
                          RESV refresh: 24, ttd: 128)
   Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 1500
   Fast Reroute: one-to-one backup desired
    Setup priority: 7, hold priority: 0
    Bandwidth: 0 kbps, hop limit: 255
    Detour LSP: UP.  Nexthop (node) protection available.
     Up/Down times: 1, num retries: 0
   Explicit path hop count: 1
    10.19.19.1 (S)
   Received RRO count: 1
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.19.19.1
   PATH rcvfrom: 10.18.18.1      (e5/1            ) (MD5 OFF)
   PATH sentto:  10.19.19.1      (e5/2            ) (MD5 OFF)
   RESV rcvfrom: 10.19.19.1      (e5/2            ) (MD5 OFF)
Egress RSVP:      0 session(s)
```

Examples of MPLS bypass LSP

This section contains **show** command output for protected LSPs and bypass LSPs. The section for bypass LSP shows information for a bypass configured on an Ethernet interface and a bypass configured on a LAG.

A **show mpls lsp** or **show mpls bypass-lsp type** of command must be entered at the ingress node of the LSP or bypass LSP, and the **show mpls rsvp session name type** of command must used at transit nodes.

The subsections are separated into the following components:

- An interface with a bypass LSP protecting the interface, Displaying an interface with bypass protection on page 205
- Protected LSP configuration in an RSVP session, Protected LSP shown in RSVP session on page 206
- Bypass LSP shown in an RSVP session, Bypass LSP in an RSVP session on page 209
- An LSP that is requesting facility backup, Displaying an LSP configured for bypass protection on page 209
- Information when the bypass LSP is active, A protected LSP while the bypass LSP is active on page 210

Displaying an interface with bypass protection

This example shows that Ethernet interface 4/15 has one bypass LSP xmr4-by. Bypass LSP xmr4-by has, therefore, recorded at least this interface (and likely others) in its list of exclude interfaces. (Similarly, an interface can have multiple bypass LSPs protecting it. For example, the LSPs that traverse an interface might have destinations that make a single merge point impossible, so multiple bypass LSPs would be needed in this case to support different LSPs.).

```
device# show mpls interface ethernet 4/15
e4/15
  Admin: Up  Oper: Up
  Maximum BW: 1000000 kbps, maximum reservable BW: 1000000 kbps
  Admin group: 0x00000000
  Reservable BW [priority] kbps:
    [0] 780000     [1] 780000     [2] 780000     [3] 760000
    [4] 760000     [5] 760000     [6] 760000     [7] 760000
  Last sent reservable BW [priority] kbps:
    [0] 780000     [1] 780000     [2] 780000     [3] 760000
    [4] 760000     [5] 760000     [6] 760000     [7] 760000
  Configured Protecting bypass lsps:
    xmr4-by(UP)
```

Syntax: **show mpls interface ethernet** *name*

In the example that follows, interface e1/11 is on a LAG named Trunk3. One bypass LSP (xmr2) is protecting the interface.

```
device# show mpls interface
e1/11(Trunk3)
  Admin: Up  Oper: Up
  Maximum BW: 13000000 kbps, maximum reservable BW: 13000000 kbps
```

```
Admin group: 0x00000000
Reservable BW [priority] kbps:
  [0] 12981000    [1] 12981000    [2] 12981000    [3] 12981000
  [4] 12981000    [5] 12981000    [6] 12981000    [7] 12981000
Last sent reservable BW [priority] kbps:
  [0] 12981000    [1] 12981000    [2] 12981000    [3] 12981000
  [4] 12981000    [5] 12981000    [6] 12981000    [7] 12981000
Configured Protecting bypass lsps:
  xmr2(UP)
```

Syntax: show mpls interface *name*

Protected LSP shown in RSVP session

Show the MPLS RSVP session for protected LSP xmr3-199. The line "Backup LSP UP. Nexthop (node) protection available" shows that protection is available for xmr-199. If this LSP were actually riding the bypass LSP, this status would change from "protection available" to "in use."

```
deviceXMR5(config-mpls-bypasslsp-123)# show mpls rsvp sess name xmr3-199
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup  BM: Merged Backup BE:Egress Backup
       RP:Repaired Session  BYI: Bypass Ingress
To          From         State Style Lbl_in Lbl_out LSP name
10.33.33.33 10.55.55.55 Up    SE    -      2399    xmr3-199
  Tunnel ID: 121, LSP ID: 1
  Time left in seconds (PATH refresh: 11, ttd: 137
                        RESV refresh: 8, ttd: 138)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Fast Reroute: Facility backup desired
   Setup priority: 4, hold priority: 3
   Bandwidth: 0 kbps, hop limit: 255
   Backup LSP: UP. Nexthop (node) protection available.
     Up/Down times: 1, num retries: 0
  Explicit path hop count: 4
   10.0.0.1 (S) -> 10.0.0.38 (S) -> 10.0.0.2 (S) -> 10.0.0.9 (S)
  Received RRO count: 4
   Protection codes: P: Local N: Node B: Bandwidth I: InUse
   10.0.0.1 -> 10.0.0.38 -> 10.0.0.2 -> 10.0.0.9
  PATH sentto: 10.0.0.1 (e3/1 ) (MD5 OFF)
  RESV rcvfrom: 10.0.0.1 (e3/1 ) (MD5 OFF)
To          From         State Style Lbl_in Lbl_out LSPname
10.33.33.33 10.0.0.2(BI) Up    SE    -      3       xmr3-199
  Tunnel ID: 121, LSP ID: 1
  Time left in seconds (PATH refresh: 0, ttd: 4196607
                        RESV refresh: 8, ttd: 4196765)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 1
   10.0.0.21 (S)
  Backup Sent
  PATH sentto: 10.0.0.21 (e2/13 ) (MD5 OFF)
  RESV rcvfrom: 10.0.0.21 (e2/13 ) (MD5 OFF)
  Riding bypass lsp: xmr3-1
```

Syntax:show mpls rsvp sessionname *name*

Displaying bypass LSPs

This section has a variety of bypass LSP displays. The first example shows the running configuration. This output shows the name of the bypass LSP, its destination interface, the exclude interface e1/1 (of the protected LSP), and that the bypass LSP is enabled.

Syntax: show mpls bypass-lsp

To display any bypass LSPs that exist on the router, use the following command.

```
device(config-if-e1000-2/15)# show mpls bypass-lsp
Note: LSPs marked with * are taking a Secondary Path
```

```
                            Admin Oper Tunnel   Up/Dn Retry Active
Name              To        State State Intf    Times No.   Path
xmr1-2            10.11.11.11 UP    UP    tnl4    2     0     --
xmr1-1            10.11.11.11 UP    UP    tnl3    2     0     --
xmr4              10.44.44.44 UP    UP    tnl7    2     0     xmr4-1
xmr1              10.11.11.11 UP    UP    tnl2    2     0     --
xmr1-5            10.11.11.11 UP    UP    tnl5    2     0     --
xmr3              10.33.33.33 UP    UP    tnl6    2     0     --
```

Syntax: show mpls bypass-lsp

The following example displays details for the bypass LSP named xmr1-2.

```
device(config-if-e1000-2/15)# show mpls bypass-lsp xmr1-2
LSP xmr1-2, to 10.11.11.11
  From: 10.22.22.22, admin: UP, status: UP, tunnel interface (primary path): tnl4
  Times primary LSP goes up since enabled: 2
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: NONE, up: yes, active: yes
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 22000 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl4, outbound interface: e2/15
   Tunnel index: 5, Tunnel instance: 1 outbound label: 3
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: yes
   Explicit path hop count: 1
    10.0.0.37 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.0.0.37
  exclude interface(s): e1/1

  Tunnel bandwidth
  Maximum BW: 22000 kbps
  Reservable BW [priority] kbps:
    [0] 22000      [1] 22000   [2] 22000   [3] 2000
    [4] 2000    [5] 2000    [6] 2000    [7] 2000
```

Syntax: show mpls bypass-lsp *lsp_name*

The *lsp_name* variable specifies the name of the LSP the user wants to display.

To display the detailed bypass LSP configuration under the router MPLS mode, enter the **show mpls bypass-lsp detail** command. The following example shows the command output for an adaptive bypass LSP. In addition to the active path information, it shows instance-specific data for each bypass instance. With this feature, the exclude interface list is included as part of the instance-specific data, and not as part of the active path data.

```
device(config-mpls-bypasslsp-b1)# show mpls bypass-lsp detail
LSP b1, to 10.6.6.6
  From: 10.7.7.7, admin: UP, status: UP, tunnel interface(primary path): tnl0
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0  Adaptive
  Maximum retries: NONE, no. of retries: 0
  Pri. path: NONE, up: yes, active: yes
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: no
   Path cspf-group computation-mode: disabled, cost: 1
  Tie breaking: random, hop limit: 0
  exclude interface(s): e3/1
  OTHER INSTANCE PRIMARY: NEW_INSTANCE admin: DOWN, status: DOWN
  Maximum retries: NONE, no. of retries: 0
  Pri. path: NONE, up: no, active: no
```

```
Setup priority: 7, hold priority: 0
Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
Constraint-based routing enabled: yes
 Path calculated using constraint-based routing: no
 Path calculated using interface constraint: no
Path cspf-group computation-mode: disabled, cost: 0
Tie breaking: random, hop limit: 0
exclude interface(s): e3/2
Active Path attributes:
 Tunnel interface: tnl0, outbound interface: e3/2
 Tunnel index: 3, Tunnel instance: 1 outbound label: 3
 Recorded routes:
  Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
  10.2.2.1
Tunnel bandwidth
Maximum BW: 0 kbps
Reservable BW      [priority] kbps:
[0] 0      [1] 0     [2] 0       [3] 0
[4] 0      [5] 0     [6] 0       [7] 0
```

Syntax: show mpls bypass-lsp detail

The **show mpls bypass-lsp wide** command allows the user to display the full bypass LSP name in a single line. Previously, a long LSP name (greater than 12 characters) was text-wrapped in multiple lines. Now, the full LSP name can be displayed in a single line as shown in the following example.

```
device(config)# show mpls bypass-lsp wide
```

> **NOTE**
> Note: LSPs marked with * are taking a Secondary Path Admin Oper Tunnel Up/Dn Retry Active Name To State State Intf Times No. Path by1 10.3.3.3 UP UP tnl1 1 0 -- by2 10.3.3.3 UP UP tnl2 1 0 -- bypasstunnelfromsanfranciscotonewyork 10.3.3.3 UP UP tnl5 1 0 pathfromsanfrancisotonewyourk

Syntax: show mpls bypass-lsp wide

The **include** option can be used with the **show mpls bypass-lsp wide** command to filter and display specific bypass LSP name.

```
device# show mpls bypass-lsp wide | include bypasstunnelfromsanfranciscotonewyork
        Admin      Oper  Tunnel  Up/Dn  Retry  Active
Name    To         State State  Intf   Times  No.   Path bypasstunnelfromsanfranciscotonewyork
        10.3.3.3   UP    UP     tnl5   1      0     pathfromsanfrancisotonewyourk
```

Syntax: show mpls bypass-lsp [wide |[include *lsp_name* **]]**

The *lsp_name* variable specifies the name of the LSP the user wants to display.

The following example shows details of an adaptive bypass LSP named t100:

```
device# show mpls bypass-lsp name t100
LSP t100, to 10.1.1.1
  From: 10.2.2.2, admin: UP, status: UP
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0   Adaptive
  Maximum retries: NONE, no. of retries: 0
  Pri. path: NONE, up: no, active: no
  Setup priority: 7, hold priority: 0, ReoptimizeTimer 300
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
   Path calculated using constraint-based routing: no
   Path calculated using interface constraint: no
  Tie breaking: random, hop limit: 0
  Active Path attributes:
```

Bypass LSP in an RSVP session

Use the **show mpls rsvp session** command to display bypass LSP xmr1-by. This example shows bypass LSP traversing a LAG, and the BYI field shows this is the bypass ingress.

```
device# show mpls rsvp sess name xmr1-by
Codes: DI:Ingress Detour DT:Transit Detour DM:Merged Detour
       DE:Egress Detour BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress

To                From                 State Style Lbl_in  Lbl_out  LSPname
10.11.11.11       10.55.55.55(BYI)     Up    SE    -       1267     xmr1-by
  Tunnel ID: 512, LSP ID: 1
  Time left in seconds (PATH refresh: 8, ttd: 148
                        RESV refresh: 10, ttd: 140)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 2
   10.0.0.13 (S) -> 10.0.0.9 (S)
  Received RRO count: 2
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.0.0.13 -> 10.0.0.9
  PATH sentto:  10.0.0.13      (p4/3(Trunk1)  ) (MD5 OFF)
  RESV rcvfrom: 10.0.0.13      (p4/3(Trunk1)  ) (MD5 OFF)
```

Syntax: show mpls rsvp sess name *name*

Displaying an LSP configured for bypass protection

This example shows that:

- LSP xmr3-120 is a candidate a bypass LSP. (The line "Fast Reroute facility backup desired" shows that LSP xmr3-120 has requested facility backup.)

- The subsequent line shows that xmr3-120 has selected bypass LSP xmr1-by for protection.

- Bypass LSP xmr1-by is up.

- The interface is on LAG p4/3.

```
device# show mpls lsp xmr3-120
LSP xmr3-120, to 10.33.33.33
  From: 10.55.55.55, admin: UP, status: UP, tunnel interface(primary path): tnl35
  revert timer: 10 seconds
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: xmr3-100, up: yes, active: yes
   Setup priority: 4, hold priority: 3
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   Tie breaking: random, hop limit: 0
  Sec. path: xmr3-101, active: no
   Hot-standby: yes, status: up
   Setup priority: 7, hold priority: 0
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   hop limit: 0
  Active Path attributes:
   Tunnel interface: tnl35, outbound interface: e3/1
   Tunnel index: 36, Tunnel instance: 1 outbound label: 1032
   Path calculated using constraint-based routing: yes
   Path calculated using interface constraint: no
   Explicit path hop count: 4
    10.0.0.1 (S) -> 10.0.0.38 (S) -> 10.0.0.2 (S) -> 10.0.0.5 (S)
   Recorded routes:
    Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
    10.0.0.1 (PN) -> 10.0.0.38 (P) -> 10.0.0.2 ->
    10.0.0.5
  Fast Reroute: facility backup desired
```

```
      Backup LSP: UP, out-label: 1032, outbound interface: p4/3(Trunk1) bypass_lsp: xmr1-by
      FRR Forwarding State: Pri(active), Sec(up), Backup(up)
```

Syntax: show mpls lsp *name*

A protected LSP while the bypass LSP is active

This section shows two views of a protected LSP while the bypass LSP is active.

To show a protected LSP while its bypass LSP is active, display the RSVP session for the LSP named xmr3-120. This bypass LSP is on LAG p4/3, and two lines in the output show that xmr3-120 is riding xmr1-by.

```
device# show mpls rsvp sess name xmr3-120
Codes: DI:Ingress Detour  DT:Transit Detour  DM:Merged Detour
       DE:Egress Detour  BI:Ingress Backup BM: Merged Backup BE:Egress Backup
       RP:Repaired Session BYI: Bypass Ingress
To              From              St     Style Lbl_in Lbl_out LSPname
10.33.33.33     10.55.55.55(RP)   Up     SE    -      1032    xmr3-120
  Tunnel ID: 36, LSP ID: 1
  Time left in seconds (PATH refresh: 14, ttd: 141
                        RESV refresh: 36, ttd: 155)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Fast Reroute: Facility backup desired
   Setup priority: 4, hold priority: 3
   Bandwidth: 0 kbps, hop limit: 255
   Detour LSP: UP.  Nexthop (link) protection available and is in use.
    Up/Down times: 1, num retries: 0
  Explicit path hop count: 4
   10.0.0.1 (S) -> 10.0.0.38 (S) -> 10.0.0.2 (S) -> 10.0.0.5 (S)
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.0.0.9 -> 10.0.0.38 (P) -> 10.0.0.2 ->
    10.0.0.5
  RESV rcvfrom: 10.0.0.9        (p4/3(Trunk1)  ) (MD5 OFF)
 Riding bypass lsp: xmr1-by
To              From              St     Style Lbl_in Lbl_out LSP name
10.33.33.33     10.0.0.2(BI)      Up     SE    -      1032    xmr3-120
  Tunnel ID: 36, LSP ID: 1
  Time left in seconds (PATH refresh: 37, ttd: 155
                        RESV refresh: 36, ttd: 155)
  Tspec: peak 0 kbps rate 0 kbps size 0 bytes m 20 M 65535
  Explicit path hop count: 4
   10.0.0.9 (S) -> 10.0.0.38 (S) -> 10.0.0.2 (S) -> 10.0.0.5 (S)
  Received RRO count: 4
   Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
   10.0.0.9 -> 10.0.0.38 (P) -> 10.0.0.2 ->
    10.0.0.5
  Backup Sent
  PATH sentto:  10.0.0.9        (p4/3(Trunk1)  ) (MD5 OFF)
  RESV rcvfrom: 10.0.0.9        (p4/3(Trunk1)  ) (MD5 OFF)
  Riding bypass lsp: xmr1-by
```

Syntax: show mpls rsvp session name *name*

The following command shows an LSP that is using its bypass. Note the last lines of output.

```
device# show mpls lsp xmr3-120
LSP xmr3-120, to 10.33.33.33
  From: 10.55.55.55, admin: UP, status: UP, tunnel interface(primary path): tnl35
  revert timer: 10 seconds
  Times primary LSP goes up since enabled: 1
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: xmr3-100, up: yes (backup), active: yes
   Setup priority: 4, hold priority: 3
   Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
   Constraint-based routing enabled: yes
   Tie breaking: random, hop limit: 0
```

```
     Sec. path: xmr3-101, active: no
      Hot-standby: yes, status: up
      Setup priority: 7, hold priority: 0
      Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
      Constraint-based routing enabled: yes
      hop limit: 0
     Active Path attributes:
      Tunnel interface: tnl35, outbound interface: p4/3(Trunk1)
      Tunnel index: 36, Tunnel instance: 1 outbound label: 1032
      Path calculated using constraint-based routing: yes
      Path calculated using interface constraint: no
      Explicit path hop count: 4
       10.0.0.9 (S) -> 10.0.0.38 (S) -> 10.0.0.2 (S) -> 10.0.0.5 (S)
      Recorded routes:
       Protection codes: P: Local  N: Node  B: Bandwidth  I: InUse
       10.0.0.9 -> 10.0.0.38 (P) -> 10.0.0.2 ->
       10.0.0.5
     Fast Reroute: facility backup desired
      Backup LSP: UP, out-label: 1032, outbound interface: p4/3(Trunk1) bypass_lsp: xmr1-by
      FRR Forwarding State: Pri(down), Sec(up), Backup(active)
```

Syntax: show mpls lsp *name*

Bypass LSP statistics

When the traffic is rerouted over a bypass LSP by one or more regular facility-protected FRR LSPs, the ingress node of bypass LSP collects the statistics for the incoming traffic. This applies for both dynamic and static LSPs.

To collect statistics of a bypass LSP we need to collect statistics for two types of Fast Reroute (FRR) protected LSP. Those which originate at same node and bypass LSP and those which originate at some node upstream.

In the following figure, node R1 is the ingress node for both the regular LSP (in blue) and bypass LSP (in red). There can be a few regular LSPs (blue) that originates at node R1 and use the bypass LSPs (red) for Fast Reroute (FRR) protection. To collect the bypass LSP (red) statistics, you must collect statistics of all regular LSPs (blue) that takes protection of the bypass LSPs (red) for FRR protection.

FIGURE 46 LSP ingress, bypass, and transit nodes
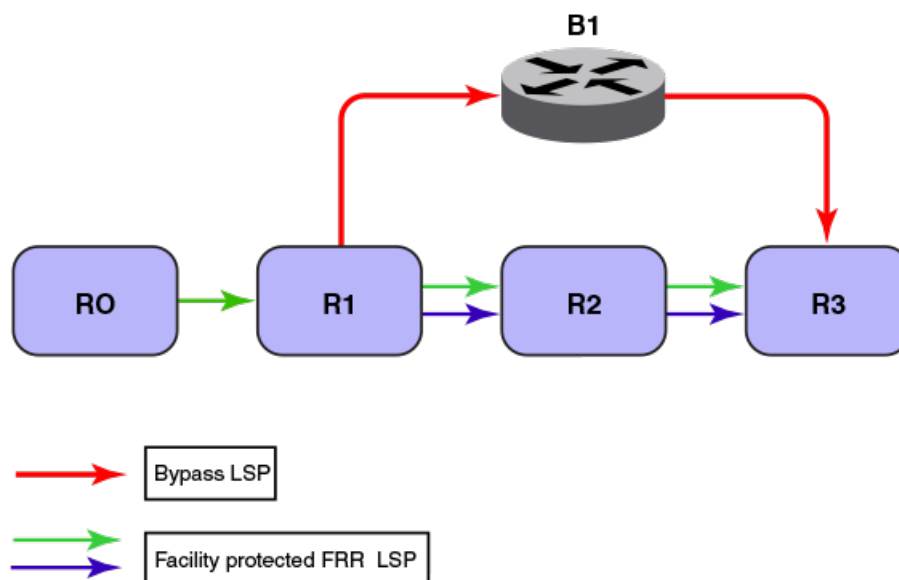
In the figure node R1 is a transit node for all facility protected FRR LSPs (in green). We need to collect statistics of all facility protected FRR LSPs (green) which uses bypass LSPs (red) for FRR protection.

> NOTE
> To collect the statistics of the regular facility protected FRR LSPs, it is always necessary to configure the ingress tunnel accounting at ingress node.

Configuring ingress tunnel accounting

This section explains how to configure ingress tunnel accounting at Link Switch Routers (LSR).

1. Enter **router mpls** command to configure MPLS in global configuration mode.

2. Enter **policy** command to set the MPLS policy.

3. Enter **ingress-tunnel-accounting** command to configure ingress tunnel accounting at Link Switch Routers (LSR).

The following example shows how to configure ingress tunnel accounting at Link Switch Routers (LSR).

```
device# configure
device(config)# router mpls
device(config-router-mpls)# policy
device(config-router-mpls-policy)# ingress-tunnel-accounting
```

Limitations

1. To accurately collect the statistics for the bypass LSP, it is necessary to configure lsp-out-acl CAM to its maximum value using the **system-max lsp-out-acl-cam** command in global configuration mode.

2. For CES 2000 Series and CER 2000 Series devices, either packets or bytes will be collected based on the configuration.

3. 24X10G cards do not support single hop tunnel accounting.

4. Ping packets sent using **ping mpls rsvp lsp***lsp_name* command will not be accounted for bypass tunnel.

5. Ingress tunnel accounting is not supported on CES 2000 Series and CER 2000 Series devices.

LSP Tunnel Persistent Index

LSP tunnel persistent index

Gives the user control over the value of the interface index.

An MPLS LSP tunnel has an interface index which can be polled using SNMP etc to get information like stats. If there is a reboot and the LSP comes up after that, if the interface index is persisted with, we would not need to get the interface index again to check if it has changed or not. Giving the user control over the value of the interface index of the tunnel means more control for the user to manage the LSP better.

When a user creates an RSVP LSP, the tunnel for this LSP gets an interface index that falls within a particular range within the system. The user, however, has no control over the interface index, and when the configuration is saved and the system reloaded, the interface index after the reload could be different from what it was before the reload.

The user could be polling the LSP using SNMP or other management mechanisms using the interface index of the LSP tunnel. After a reboot, however, the LSP's interface index would change. This would mean that the user has to query the LSP for the interface index again and then start the polling again with the new interface index. With this feature, the user has control of what the tunnel interface index is, and will also be guaranteed persistence of that index across reloads.

This interface index value will be for the tunnel and is shared between the paths - primary or secondary - and instances.

Requirements

This feature 'LSP Tunnel Persistent Index' aims to provide the following capabilities:

1. Allow the user to configure the interface index of LSPs and bypass LSPs.
2. Persist with the same interface index of the LSP and bypass LSP across reloads.
3. Expand the interface-index range for MPLS tunnels and split the expanded range between LDP and RSVP tunnels.

Compatibility considerations

Considerations and compatibility issues.

Upgrade and downgrade considerations

When a system is downgraded, the configuration that is saved with the release still works, however, the interface-index command fails in that configuration and the interface index does not persist across re-boots.

Backward compatibility

A configuration saved from this release throws errors on older releases when encountering the command for interface index of an LSP.

Customer configuration

LSP tunnel persistent index customer configurations.

The user may configure an ingress tunnel from A to C. An *Simple Network Management Protocol (SNMP)* server would be polling statistics on the LSP using the tunnel-interface index 100.
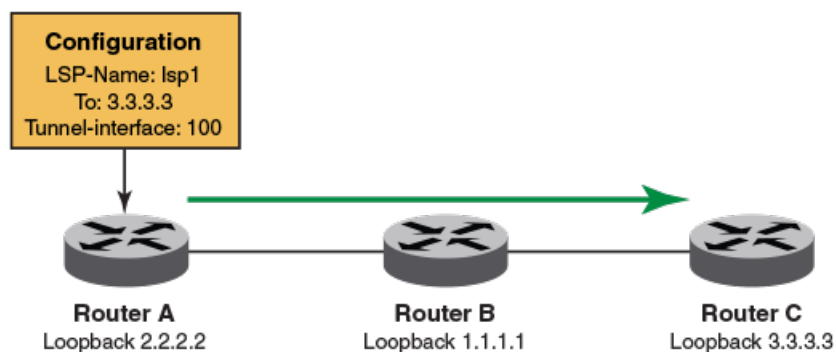
1. The user configures the following attributes for the *Label Switching Path (LSP)*:
   a. LSP Name.
   b. "TO" Address.
   c. Interface Index.
2. The user saves the configuration and reloads the router,

3. After reload, the LSP comes up with the same interface index as configured a 1c above.

4. The polling from the SNMP server would continue as before with the same interface index.

Configuration example

Configuring the LSP tunnel interface index:

```
device# configure terminal
device(config)#router mpls
device(config)#lsp lsp1
device(config)#tunnel-interface 100
device(config)#to 3.3.3.3
device(config)#enable

device# configure terminal
device(config)#router mpls
device(config)#bypass-lsp byp1
device(config)#tunnel-interface 102
device(config)#to 3.3.3.3
device(config)#exclude-interface eth 2/1
device(config)#enable
```

PCEP

Path Computation Element Protocol

Path Computation Element Protocol (PCEP) provides support for MPLS, or any other system, to send a path computation request to compute a Traffic Engineering (TE) constrained path, to an external node or a Path Computation Element (PCE), and obtains the computed path. The LSP is established based on the path returned.

The computed path may be to a destination that is within the same Autonomous System (AS) as the requesting node, or in a different AS. The device making the request is called the Path Computation Client (PCC). When the PCE has the capability to compute the path across Autonomous Systems, the path is computed. When path computation is unsuccessful, the PCE returns a failure message.

> **NOTE**
> Only the Path Computation Client (PCC) applies for the NetIron 6.0 release. The MLX/CES/CER functions as the PCC only and does not support the PCE function. The device running PCEP can make a request for path computation to an external device, but is not able to service a request for path computation made by another device.

PCE requirements

To acquire a computed path from an external PCE, complete the following steps.

- Configure the IPv4 address of one or more PCE nodes.
- Configure the parameters of the PCE node – for example, the keepalive timer and the dead timer.
- Establish a TCP connection to the PCE node.
- Allow an LSP to specify the PCE node that computes the path. –OR–
- Allow an LSP to specify that a PCE computes the path.
- Send a path computation request using a PCReq message.
- Obtain a path computation response and PCRep message.
- Close the TCP connection with the PCE node when not needed.

PCEP considerations

Review these reliability, availability, and serviceability issues before upgrading to PCEP.

PCEP limitations and prerequisites

At least one external PCE is expected to compute a path. When the configured PCE does not exist, the path is not computed and the LSP does not come up.

Upgrade and downgrade considerations for PCEP

When upgrading to PCEP, an LSP path can be computed using an external PCE node.

When disabling PCEP, the LSP path can be computed only using the local Constrained Shortest path First (CSPF) with a local computer TE database.

Backward compatibility for PCEP

An LSP configured to use a PCE node can come up in releases prior to NetIron 6.0, but returns an error when executing a command to specify the PCE computation. The path of the LSP is computed by the local CSPF, which may or may not be able to compute the path. In the case of LSPs with destinations in a different Autonomous System, such a computation fails.

Performance considerations for PCEP

The path computation must communicate with an external node. Communication with an external node introduces a delay in the computation of the path of the LSP and thus, the LSP bring-up time may be slower. Because path computation involves messages sent to an external node, in a scaled scenario there may be several messages being sent and received to compute a path.

PCEP architecture

The following figure is a sample PCEP architecture for an external PCE showing the sequence of events.

FIGURE 47 External PCE
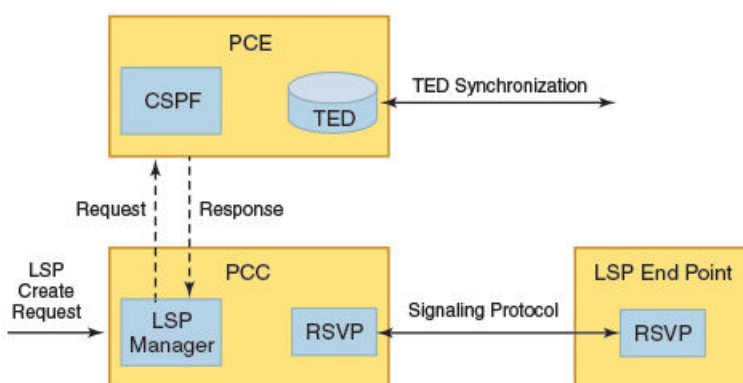
1. The LSP Manager receives an LSP create request through a CLI.
2. To compute the path to the destination, the LSP Manager sends a request to the PCE.

3. The PCE node computes the path using the CSPF and Traffic Engineering Database (TED).

4. The PCE node returns the computed path to the head end node.

5. The head end node passes on the computed path to the RSVP to signal the LSP.

FIGURE 48 Inter-AS route calculation
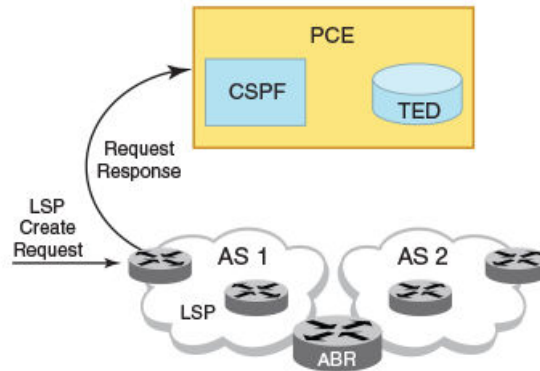
The preceding figure displays the calculation of an Inter-AS route.

1. The head end node receives a request to establish an LSP to a destination that lies in a different area (Autonomous System).

2. The head end node consults the PCE mode that has knowledge of both Autonomous Systems.

3. The PCE node computes the path based on its TEDs and uses other PCEs if needed.

4. The PCE node returns the computed path based on the established LSP.

PCEP configuration steps

How to configure parameters for the PCE server.

1. How to configure PCE server specific parameters. All commands are under the **config-pcep** configuration.

```
device> enable
device# configure terminal
device(config)# router pcep
device(config-pcep)# ?

router pcep
  keepalive 25
  min-keepalive 12
  dead-timer 100
  request-timer 30
  max-unknown-messages 7
  max-unknown-requests 4
```

2. How to configure an LSP to use an external PCE server to compute the path.

```
device> enable
device# configure terminal
device(config)# pce NY_sever ?

  pce NY_server
    address 135.40.22.9
    key session1745NY
    preference 4
    message-bundle-support
    enable
```

3. Optionally, an LSP can choose to specify a particular PCE server for its path computation.

```
device> enable
device# configure terminal
device(config)# router mpls
device(config-mpls)#bypass-lsp byp_to_DC
device(config-mpls-bypass-lsp-byp_to_DC)# ?

 bypass byp_to_DC
 to 152.23.12.8
 exclude-interface eth 2/5
 pce compute
 enable

  lsp to_DC
    to 152.25.14.8
    pce compute NY_server
    enable
```

PCEP route computation types

PCEP supports a number of route computation types when a route computation is requested. The following table gives a depiction of the supported route computations when requesting a route from an external PCE server.

TABLE 12 Types of route computations

Route computation type	Supported (Yes/No)	Comments
Regular LSP	Yes	
Re-optimization requests	Yes	Re-optimization is driven by the timer, auto-bandwidth LSP, topology change, global revertiveness, soft preemption.
Static Bypass LSP	Yes	
Dynamic Bypass LSP	No	Node can make use of local CSPF and TE database to calculate the dynamic LSP.
FRR detour	No	Point of Local Repair (PLR) router can make use of local CSPF and TE database to calculate its detour path.
FRR Facility backup	No	Stateless PCE does not have a database of existing bypass LSPs on the Path Computation Client (PCC). Therefore, the choice of the bypass LSP is made locally. In this case, even when the LSP is configured to use the PCE server for the route, it consults the local database for backup route selection. When triggering a new dynamic bypass LSP, the route may be requested from a PCE server.
Secondary LSP	Yes	Hot-standby and non-hot-standby.
CSPF Fate sharing groups	No	This can be implemented as a policy on the PCE.

TABLE 12 Types of route computations (continued)

Route computation type	Supported (Yes/No)	Comments
CSPF Interface constraints	No	This can be implemented as a policy on the PCE.
CSPF TE/IGP metric	Yes	
Liberal Bypass	No	This is a PCE specific feature. There is no connection to PCEP because the bypass selection algorithm is purely local to the PCE. There is no consulting of the PCE server for selecting an existing bypass LSP tunnel as the backup.
Route expansion on transit nodes (due to loose hop)	No	

Configuring Label Distribution Protocol (LDP)

LDP overview

When used to create tunnel LSPs, LDP allows a set of destination IP prefixes (known as a *Forwarding Equivalence Class* or FEC) to be associated with an LSP. Each LSR establishes a peer relationship with its neighboring LDP-enabled routers and exchanges label mapping information. This label mapping information is stored in an LDP database on each LSR. When an LSR determines that one of its peers is the next hop for a FEC, it uses the label mapping information from the peer to set up an LSP that is associated with the FEC. It then sends label mapping information to its upstream peers, allowing the LSP to extend across the MPLS network.

The devices advertise their loopback addresses to their LDP peers as a 32-bit prefix-type FEC. When an LSR installs a label for a FEC, it also creates an MPLS tunnel route, which is then made available to routing applications. This allows each router to potentially be an ingress LER for an LSP whose destination is the device's loopback address.

The result of an LDP configuration is a full mesh of LSPs in an MPLS network, with each LDP-enabled router a potential ingress, transit, or egress LSR, depending on the destination.

The system supports *Label Distribution Protocol (LDP)* for the configuration of non-traffic-engineered tunnel LSPs in an MPLS network. LDP is described in *RFC 3036*.

The implementation supports the following aspects of LDP:

Liberal label retention - Each LSR sends its peers Label Mapping messages, which map a label to a FEC. Peer LSR receiving these messages retain all of the mappings, even though they may not actually be used for data forwarding.

Unsolicited label advertisement - The LSR sends Label Mapping messages to its LDP peers even though they did not explicitly request them.

Ordered label distribution - The LSR sends a Label Mapping message to its peers only when it knows the next hop for a FEC, or is itself an egress LER for the FEC. When an LSR does not know the next hop for a FEC, and is not an egress LER for the FEC, it waits until a downstream LSR sends it a Label Mapping message for the FEC. At this point, the LSR can send Label Mapping messages for the FEC to its peers. This allows label mappings to be distributed, in an orderly fashion, starting from the egress LER and progressing upstream.

The Multi-Service IronWare software the LDP label space ID has a default value of zero which improves interoperability with routers from other vendors.

Configuring LDP on an interface

To use LDP, a loopback address (with a 32-bit mask) must be configured on the LSR. The first loopback address configured on the device is used in its LDP identifier. When the loopback address used in the LDP identifier is removed, all LDP functions on the LSR is shut down. LDP sessions between the LSR and its peers are terminated, and LDP-created tunnels are removed. When other loopback interfaces are configured on the device, the lowest-numbered loopback address is then used as a new LDP identifier. LDP sessions and tunnels are set up using this new LDP identifier.

To configure LDP on an interface, enter commands such as the following.

```
device(config)# router mpls
device(config-mpls)# mpls-interface e 1/2
device(config-mpls)# ldp-enable
```

Syntax: [no] ldp-enable

> NOTE
> The user must enable LDP on the same set of interfaces that IGP routing protocols such as OSPF and IS-IS are enabled.

Configuring an option of FEC type for auto-discovered VPLS peers

By default, Extreme devices use FEC 129 to send the VC label binding for auto-discovered VPLS peers. There are mixed environments where VPLS static configured peers and auto-discovered peers exist. In these environments, the following VPLS command allows the user to configure FEC 128 for all VPLS. Enter a command such as the following.

```
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# fec-128-for-auto-discovered-peers
```

Syntax: [no] fec-128-for-auto-discovered-peers

The default value is FEC 129.

> NOTE
> The user must reload the system for this command to take
> effect.

LDP Inbound-FEC filtering

MPLS LDP inbound-FEC filtering filters inbound label bindings on a MPLS router. The user can control the amount of memory and CPU processing involved in installing and advertising label bindings not used for forwarding.

MPLS LDP inbound-FEC filtering also serves as a tool to avoid DOS attack. By creating a prefix-list, and specifying prefixes label mappings, the forwarding plane accepts and installs the label bindings.

The prefix-list is applied to an individual LDP session or globally to all the LDP sessions.

Configuration Considerations

- The FECs filtered by LDP inbound-FEC filter do not install in the forwarding plane or advertise to the upstream neighbors. The FEC remains in the retained state.
- The LDP inbound-FEC filter are changed directly without deleting the one previously configured. The change automatically applies and triggers the filtering of inbound FECs.
- Changes to a referenced prefix-list automatically applies to LDP inbound-FEC filtering. This triggers filtering by way of the new configuration, filtering any existing FECs which violate the filter.
- In order to allow multiple route filter updates, the device waits for default 10 seconds before notifying the application of the filter change. The time for notification is configurable.
- When the LDP inbound-FEC filter is not configured, LDP does not filter any inbound FECs.
- By default, when the prefix-list referenced by the LDP inbound-FEC filter has no configuration, it is an implicit deny. All inbound FECs are filtered out and retained. The behavior is the same when the prefix list is deleted after setting it in the inbound FEC filter configuration. This behavior is consistent with other protocols which use device filters and also with the use of the **advertise-labels** command for LDP route injection.
- Inbound FEC filtering is applicable only for L3 FECs and not for VC FECs. Inbound FEC filtering is not applicable for L2VPNs.

Configuring LDP inbound FEC filtering

Enabling LDP inbound FEC filtering

To enable LDP inbound FEC filtering, enter commands such as the following:

```
device(config)# router
device(config-mpls)# ldp
device(config-mpls-ldp)# filter-fec list-abc in
```

To set LDP to accept inbound FEC 10.20.20.0/24 and filter out all others FECs, enter commands such as the following:

```
device(config)# ip prefix-list list-abc permit 10.20.20.0/24
device(config)# router mpls
device(config-mpls)# ldp
```

Syntax: [no] **filter-fec** *prefix-list* **in**

The *prefix list* parameter specifies the prefixes.

The **in** keyword specifies inbound-fec-filter configuration.

Modifying prefix-list after setting it in the filter-inbound-FEC

When the prefix-list referenced by the LDP inbound-FEC filter is configured or changes, all the existing in-bound FECs and received later are subject to the changed prefix-list.

> **NOTE**
> There is a configurable delay between changing the prefix-list and the changed prefix-list taking effect on LDP FEC-filter configuration.

```
device(config)# ip prefix-list list-abc permit 10.20.20.0/24
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# filter-fec  list-abc  in
device(config-mpls)# exit
device(config)# exit
device(config)# ip prefix-list list-abc permit 10.21.21.21/24
```

Syntax: [no] filter-fec *prefixlist* **in**

The *prefix list* parameter specifies the prefixes.

The **in** keyword specifies inbound-fec-filter configuration.

Sample Configurations

The following examples use the FEC filtering parameter:

Consider three MPLS router system devices with an ID 10.66 with the transit device between them.

FIGURE 49 Inbound FEC filtering example

1. Use the following command to configure the prefix list to allow all /32 addresses:

   ```
   device(config)# ip prefix filter172_24 permit 172.16.09.0/16 ge 24 le 24
   ```

2. Configure the prefix list to allow 172.16.0.0/16 ge 24 le 24:

   ```
   device(config)# ip prefix filter172_24 permit 172.16.09.0/16 ge 24 le 24
   ```

3. Configure the prefix list to allow 172.16.0.0/16 ge 24 le 28:

   ```
   device(config)# ip prefix-list filter172_28 permit 172.16.0.0/16 ge 24 le 28
   ```

4. Configure the prefix list to allow all of the above FECs:

```
device(config)# ip prefix-list filterAll permit 0.0.0.0/0 ge 32
device(config)# ip prefix-list filterAll permit 172.16.0.0/16 ge 24 le 28
```

Verify the configuration by using the **show mpls ldp** command.

```
device(config)# show mpls ldp
Label Distribution Protocol version 1
  LSR ID: 10.44.44.44, using Loopback 1 (deleting stops LDP)
  Hello interval: Link 5 sec, Targeted 15 sec
  Hold time value sent in Hellos: Link 15 sec, Targeted 45 sec
  Keepalive interval: 6 sec, Hold time multiple: 6 intervals
  Load sharing: 1
  Tunnel metric: 0
  FEC used for auto discovered peers: current 129, configured 129
  Graceful restart: disabled
device(config)# show mpls ldp database
Session 10.44.44.44:0 - 10.14.14.14:0
 Downstream label database:
   Label        Prefix          State
   3            10.14.14.14/32   Installed
   1024         172.16.8.0/24    Installed
   1025         172.16.16.0/24   Installed
   1026         172.16.32.0/24   Installed
   1027         172.16.64.0/24   Installed
   1028         172.16.8.0/28    Installed
   1029         172.16.8.16/28   Installed
   1030         172.16.8.32/28   Installed
   1031         172.16.8.64/28   Installed
 Upstream label database:
   Label        Prefix
   3            10.44.44.44/32
   1033         10.66.66.66/32
Session 10.44.44.44:0 - 10.66.66.66:0
 Downstream label database:
   Label        Prefix          State
   3            10.66.66.66/32   Installed
 Upstream label database:
   Label        Prefix
   3            10.44.44.44/32
   1024         172.16.8.0/24
   1025         172.16.16.0/24
   1026         172.16.32.0/24
   1027         172.16.64.0/24
   1028         172.16.8.0/28
   1029         172.16.8.16/28
   1030         172.16.8.32/28
   1031         172.16.8.64/28
   1032         10.14.14.14/32
```

LDP outbound FEC filtering

LDP performs a hop-by-hop or dynamic path setup in an MPLS network by assigning and distributing labels to routes learned from the underlying IGP routing protocols. By default, LDP distributes all FECs learned locally or from LDP neighbors, to all other LDP neighbors. When this is not desired, you can configure LDP to perform outbound filtering for label advertisement using the **outbound fec filtering** feature. This is achieved by creating a prefix-list specifying prefixes whose label mappings can be distributed. The prefix-list is then applied to an individual LDP neighbor, or globally to all the LDP neighbors. The FECs permitted by the prefix-list only are accordingly distributed to the specified LDP neighbor or to all LDP neighbors.

This feature gives you the ability to control which FECs can be advertised and to which LDP neighbors.

• This feature reduces the number of Labels distributed to neighbors and the number of messages exchanged with peers.

- • Improves LDP scalability and convergence.
- • Improves security and performance.

Limitations and pre-requisites

MPLS and LDP protocol must be enabled on the router to use this feature.

Upgrade and downgrade considerations

On Upgrade, because there is no outbound fec filter configured, all FEC are allowed by default. For downgrade, value of outbound "fec-filter" configured, if any, is lost.

Configuration steps

Follow the listed steps to configure LDP outbound FEC filter.

1. Create a prefix-list to permit or deny required set of FECs.
2. Go to the **ldp config** mode available on the **router mpls config** mode.
3. Set the above created prefix-list in the global or per neighbor outbound fec filter configuration.

Configuration example

For Global outbound FEC filter configuration

Example: To set LDP to prevent advertisement of FEC 10.44.44.44/32 while allowing all other FECs to all neighbors follow the listed configuration steps.

```
device(config)# ip prefix-list list-abc deny 10.44.44.44/32
device(config)# ip prefix-list list-abc permit 0.0.0.0/0 ge 32
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# filter-fec list-abc out
```

Syntax: [no] **filter-fec** *prefix-list* **out**

Parameters:

prefix-list

Prefix-list specifying the prefixes.

out

Specifies outbound-fec-filter configuration.

For per neighbor outbound FEC filter configuration

Example : To set LDP to prevent advertisement of FEC 10.44.44.44/32 while allowing all other FECs to neighbor 10.12.12.12 follow the listed configuration steps.

```
device(config)# ip prefix-list list-abc deny 10.44.44.44/32
device(config)# ip prefix-list list-abc permit 0.0.0.0/0 ge 32
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# session 10.12.12.12 filter-fec list-abc out
```

Syntax: [no] **session** *peer-ip* **filter-fec** *prefix-list* **out**

Parameters:

session

The **session** keyword specifies the per session configuration.

peer-ip

The *peer-ip* parameter is the peer IP of the LDP to which the filter needs to be applied.

out

The **out** keyword specifies the outbound-fec-filter configuration.

MPLS LDP FEC display enhancement

Glossary

Glossary of terms used for MPLS LDP FEC display enhancement.

Term	Meaning
LDP	Label Distribution Protocol
LSP	Label Switched Path
FEC	Forwarding Equivalence Class
ACL	Access Control List
GR	Graceful Restart
RSVP	Resource ReSerVation Protocol

Introduction

Describes the improvements implemented for the LDP CLI commands for display of prefix FECs.

This document describes the design details for MPLS LDP FEC display enhancement.

Specifications

LDP, by default, does not display the prefix FECs in hierarchical order of the FEC definition. This feature implements changes to the command **show mpls ldp fec prefix** to display the FECs in hierarchical order.

Customer configuration examples

Show command enhancement to display the LDP prefixes FECs in order of the prefix FEC destination address and length.

The current output of **show mpls ldp fec prefix** command:

```
device# show mpls ldp fec prefix
Total number of prefix FECs: 4
Total number of prefix FECs installed: 1
Total number of prefix FECs filtered(in/out): 1/0
Total number of prefix FECs with LWD timer running: 0

Destination     State       Out-intf        Next-hop        Ingress  Egress  Filtered  LWD
144.144.1.1/32  current     e1/5            5.5.5.6         Yes      No      -         No
```

```
                                 e1/6           6.6.6.6
77.77.77.77/32   current   --             --              No       Yes     -          No
155.0.0.0/8      current   e1/3           3.3.3.5         Yes      No      -
No
                                 e1/4           4.4.4.5
144.144.1.64/32 current    e1/5           5.5.5.6         Yes      No      In         No
                                 e1/6           6.6.6.6
```

New output with changes implemented for this feature:

```
device# show mpls ldp fec prefix
Total number of prefix FECs: 4
Total number of prefix FECs installed: 1
Total number of prefix FECs filtered(in/out): 1/0
Total number of prefix FECs with LWD timer running: 0

Destination      State      Out-intf       Next-hop        Ingress  Egress  Filtered   LWD
77.77.77.77/32   current    --             --              No       Yes     -          No
144.144.1.1/32   current    e1/5           5.5.5.6         Yes      No      -          No
                            e1/6           6.6.6.6
144.144.1.64/32  current    e1/5           5.5.5.6         Yes      No      In         No
                            e1/6           6.6.6.6
155.0.0.0/8      current    e1/3           3.3.3.5         Yes      No      -          No
                            e1/4           4.4.4.5
```

Label withdrawal delay

The label withdrawal timer delays sending a label withdraw message for a FEC to a neighbor.

When an LDP session fails, the label associated with a FEC is withdrawn from all upstream peers. In addition, if the IGP adjusts the route for the FEC such that the current neighbor is no longer the next hop for the FEC, then the associated FEC label is withdrawn from all upstream peers.

The label withdrawal delay timer introduces a configurable delay to allow the IGP and LDP to converge after these events. The delay helps avoid sending the label withdraw message to the upstream peers. For example, after a link failure, instead of immediately sending a label withdraw to all upstream peers for the FEC; the delay allows the IGP to install a route which may match another existing downstream session. Label withdrawal from all upstream peers can be avoided if the FEC achieves a downstream label mapping which is consistent with the IGP routing table.

If the timer expires, the FEC label is then withdrawn from all upstream peers.

Upgrading to this feature

If a system is upgraded to a release which does not support the label withdrawal delay feature, then the newly upgraded system will exhibit new behavior as label withdrawal delay is enabled by default.

If a system is upgraded from a release which supports label withdrawal delay, then the newly upgraded system may exhibit new behavior; as follows:

- If there is no label withdrawal delay configuration, the system will exhibit the default behavior after the upgrade. This means that the system will change to having label withdrawal delay enabled.
- If a label withdrawal configuration exists, the label withdrawal delay feature will be enabled. However, if the value for the timer matches the default (example: 60 seconds) then the configuration line is removed from the running configuration.

Downgrade information

If a system running is downgraded to a release which does not support this feature, the feature is not available.

If the system is downgraded to a release which does support this featureless example the behavior is as follows:

- If the configuration has a line for disabling label withdrawal delay (example, the value is set to zero) then the feature is disabled after the downgrade.
- If there is no label withdrawal delay configuration then the feature is disabled after the downgrade.
- If the configuration has a line for enabling label withdrawal delay because, for example, it is set to a non-default value then the feature is enabled after the downgrade.

Configuring the label withdrawal delay timer

The label withdrawal delay timer delays sending a label withdrawal message for a FEC to a neighbor. This feature is enabled by default.

Examples

To set the label withdrawal delay timer to 30 seconds, enter the following command:

```
device(config-mpls-ldp)# label-withdrawal-delay 30
```

To restore the label withdrawal delay timer default behavior, when the delay period is already configured as 30 seconds, enter the following command:

```
device(config-mpls-ldp)# no label-withdrawal-delay 30
```

To disable the label withdrawal delay timer, enter the following command:

```
device(config-mpls-ldp)# label-withdrawal-delay 0
```

[no] label-withdrawal-delay *secs*

The *secs* variable specifies the delay period (in seconds) for the label withdrawal delay timer. The range is 0 - 300. The default value is 60.

Setting the *secs* variable to a value in the range 1 - 300, updates the configured value.

Setting the *secs* variable to zero disables the label withdrawal delay feature for subsequent events. Any FEC which has already started the label withdrawal delay timer continues to run the timer and to delay sending its label withdrawal messages upstream.

The [**no**] form of the command restores the default behavior. The value specified for the *secs* variable must match the configured value at the time that the [**no**] form of the command is executed.

> NOTE
> A typical convergence period in a heavily scaled network (50 LDP sessions and 4k FECs) on the XMR platform is less than four seconds for a single link up or session down event.

Label withdrawal delay and LDP *Graceful Restart (GR)*

Helper node

Graceful restart helper procedures are initiated when a session to a peer goes down. During the session down processing, the label mappings exchanged with a peer are preserved while the peer is reconnecting. If the peer session is reconnected within a configurable time limit then the label mappings previously exchanged with the peer are refreshed with new label mappings. Any mappings that are not refreshed are released.

When both label withdrawal delay and GR are enabled, the label withdrawal delay timer is not initiated when the session goes down because the session is considered to be in a special restarting state and not actually down. If the session is not re-established within the reconnect time for GR then the session is considered to be down and the label withdrawal delay timer may be started for any FECs which meet the criteria for label withdrawal delay. If the session is re-established within the reconnect time for GR then the label withdrawal delay timer is not started.

Individual FECs may experience some transition as the label mappings from the peer are refreshed. For example, if a label mapping is not refreshed during the restart window, then that label mapping will be removed. This will affect the FEC, especially if it was associated with an installed downstream mapping. If an alternate route exists, the FEC will re-converge on the alternate route much earlier and again the label withdrawal delay timer is not started for the FEC

Restarting node

The GR restarting procedures are executed by a node which is starting the LDP process and has retained the forwarding state for LDP connections from an earlier instance of LDP. The restarting procedures are executed during a management module switchover. During the forwarding state hold period, connections in the forwarding state are marked stale until they are refreshed by an additional label mapping. Stale connections are removed after this period.

This scenario does not affect label withdrawal delay since it involves a complete restart of the LDP control plane. In other words, there are no FECs or sessions to apply the label withdrawal delay timer to at restart. Other procedures for label withdrawal delay as described above may occur in the same way during the forwarding state hold period.

Label withdrawal delay and LDP-IGP synchronization

LDP-IGP synchronization aims to prevent traffic loss due to the introduction of a new link into the network. Appropriate label mappings for a FEC may not be available for some time after the route for the FEC has been established on the new link.

When LDP-IGP synchronization is enabled, the IGP metric for the new link is temporarily advertised at a maximum value to force traffic to use an alternate route, if one is available. After all label mappings are received on the link, the IGP metric is adjusted on the link to the normal value and route updates may occur as the cost of the link has been reduced.

When both label withdrawal delay and LDP-IGP synchronization are enabled, the label withdrawal delay timer will not be started if there are alternate routes for the FEC. For example, the following sequence of events is possible:

1. A FEC has an installed downstream mapping over link 1.
2. Link 2 is introduced to the network. The IGP metric for link 2 is advertised at maximum value so there is no route update for any FEC with an already established route.
3. Label mappings are received from the new peer and a new retained downstream mapping is established for the FEC.
4. When all label mappings for the session on link 2 have been received, the IGP metric is adjusted to the normal value. By default, LDP-IGP sync hold down time is disabled and IGP will wait until LDP gives and "in-sync" indication for the link before advertising it with the normal metric. If LDP-IGP sync hold-down time is enabled and label mappings are not received from the new peer within the configured sync hold-down time period, then the label withdrawal delay timer will start for the FEC. In this case, it is the time at which the label withdrawal delay timer starts that is affected: instead of starting almost immediately after the new link becomes operational, it is delayed by the time configured to allow for LDP-IGP synchronization.
5. This may result in a route change at the FEC. The FEC may install the downstream mapping associated with link 2 and transition the downstream mapping associated with link 1 to retained.

LDP label withdrawal delay at ingress

Glossary

Term	Meaning
Apply Current Route	Compare the current route with received Label Mappings and install any downstream mappings; as appropriate.
Current Route	Current next hop info for a FEC. The current route may not be applied immediately to the current Label Mapping as a result of this feature.
Downstream Mapping	This represents the Label Mapping received from a downstream peer for a FEC (Abbreviated DM).
FEC	Forwarding Equivalency Class. Each FEC is a destination (IP address) for an LDP tunnel.
LDP	Label Distribution Protocol
Label Mapping	LDP message which indicates the label which is to be used for a FEC from the peer.
LSP	Label Switched Path
LWD	Label Withdrawal Delay
Route Event	An update from the routing table to LDP.
Upstream Mapping	This represents the Label Mapping sent to an upstream peer for a FEC.

Introduction

Use the Label Withdraw Delay Enhancement to improve network convergence for an LDP network by avoiding sending Label Withdraw protocol messages after specific events in the network.

Specifications

The label withdrawal delay at ingress enhancement introduces a timer which defers the reaction of LDP to session down and routing events in order to allow the network to stabilize.

This functionality is helpful for performance at ingress nodes also; that is; nodes which have not advertised a label to an upstream peer.

The benefits of LWD at ingress are:

- Reduces outage time for traffic originating from ingress tunnels.
- Reduces message generation within internal MPLS modules and line cards.

The behavior of LWD at ingress is the same as the original LWD feature; minus the actions regarding upstream labels.

Requirements

The feature requirements for LWD are the same, except as noted below:

This feature starts the LWD timer even for purely ingress FECs. The behavior is the same as for the FEC with upstream labels other than procedures for cleaning up the upstream labels on LWD expiration.

The LWD Ingress feature is enabled along with the LWD (transit) feature automatically. No new configuration command is required.

Customer configuration examples

The use cases for LWD remain intact with the adjustment that the timer is started even when there are no upstream labels.

Session down

When a session becomes inactive; each DM for the session is deleted. This may cause LW to be sent upstream if the last installed DM is removed for a FEC.

The modified behavior is as follows:

- If a DM is a candidate to be removed; the system evaluates whether or not its FEC is a candidate for delaying the LW upstream; based on the following criteria:
  - There are no other installed DMs.
- If these criteria are met, the system proceeds as follows:
  - A timer is started on the FEC for the LW delay period.
  - The DM remains installed until either a new DM becomes installed for the FEC; the route for the FEC is deleted; or the FEC timer expires.
- If these criteria are not met, then the DM is removed as per normal; resulting in the current behavior.

Route update occurs

When a route update for the FEC occurs; the installed DM may transition to retained if the next hop address for the FEC has changed.

The modified behavior is as follows:

- If a route update would result in a FEC with the following criteria:
  - The LW delay timer is not already running for the FEC.
  - The FEC would reference only retained DMs after processing the route update. Note there must be at least one DM.
- If these criteria are met, the system proceeds as follows:
  - A timer is started on the FEC for the LW delay period.
  - The current installed DM remains installed.
  - The current route information is updated. The route is not applied until there is a LM which matches the route; or the LW delay timer expires.
- If these criteria are not met, then the LW delay feature is not activated and normal procedures for the FEC are done.
- The existing behavior for link update / route update is not changed by this feature if the LW delay feature is not activated for the FEC.

LDP ECMP for transit LSR

> NOTE
> LDP ECMP for transit LSR is supported only on XMR Series and MLX Series devices.

LDP *Equal-Cost Multi-Path (ECMP)* for transit LSR provides ECMP support for transit routers on an LDP LSP. The LDP LSP tunnel at ingress continues to be a single ECMP path.

ECMP programming for LDP transit LSP creates a set of ECMP paths on the forwarding plane at any transit router. LDP LSPs transit traffic is load balanced using programmed ECMP. The number of ECMP paths that are used depends on the number of eligible paths that are available, and the maximum number of LDP ECMP paths configured by the user. The number of available paths sent to LDP are

controlled by the *Routing Table Manager (RTM)* which is limited by IP load sharing. LDP also enables its own load sharing limit. The lesser of the two load sharing limits form the maximum number of ECMP paths that can be programmed on forwarding plane.

When new ECMP paths are added, or existing paths are deleted from a set of eligible ECMP paths, MPLS forwarding decides when these changes lead to a different set of paths to be used for LDP LSP, ingress tunnel, or transit LSP. When a different set of paths are used, updates are sent to the forwarding plane. MPLS only sends an update to the forwarding plane when there is a change to the set of programmed paths. MPLS always sends the complete set of ECMP paths to the forwarding plane. When the user changes the load sharing configuration, updates are also sent to the forwarding plane. FEC updates are only generated when the new load sharing value is different from the set of ECMP paths programmed in the forwarding plane.

> **NOTE**
> LDP ECMP is not supported at the ingress router.

The ingress LDP LSP can be different from the transit LSP for the same FEC. When all ECMP paths provided by the RTM are using LDP tunneling enabled for RSVP shortcut LSP(s), then the ingress LDP tunnel is not created.

> **NOTE**
> MP switchover event may not be handled properly by MPLS/RSVP module. This may result in inconsistent state for RSVP LSPs/Sessions. This could be fixed by adding support for RSVP Hello feature.

MPLS OAM support for LDP ECMP

MPLS OAM support for traceroute at any transit router returns the list of labels used at that transit router. However, traceroute is not able to exercise all ECMP paths. The forwarding plane selects one ECMP path to forward OAM packets. All traversed labels that were returned at each transit router are displayed at the Extreme router originating the traceroute.

LDP ECMP LER

TABLE 13 Glossary

Term	Meaning
CAM	Hardware Routing Table
CAM2PRAM	Indirection pointer to PRAM table, also has no ECMP paths
ECMP	Equal Cost Multi Path
IPoMPLS	IPv4 (shortcuts) over MPLS tunnels
L3VPN	Layer-3 VPN routes
LER	Label Edge Router
PRAM	Next-hop information table
RTM	RTM module on MP

Overview

The LDP ECMP LER feature provides the capability to create LDP tunnels with up to eight paths. These tunnels can be used by applications like IP over MPLS and L3VPN to transport IPv4 and IPv6 traffic. The traffic sent over the tunnel is load-balanced across all the paths based on a hashing algorithm. The algorithm takes into account the information from the packets such as MAC address, IP address, TCP and UDP ports.

Configuration considerations

1. By default, LDP ECMP is disabled on an LER or LSR.

2. It is possible to configure LDP ECMP through the command line. The moment the configuration is done, for all tunnels for which the router is acting as ingress, changes are made to compute ECMP paths. The same command is already used to compute the ECMP paths for LSPs transiting the router (already existing functionality works without change). Information from RTM is used to get the set of equal cost paths.

3. The maximum number of paths used by LDP ECMP ingress is same as transit. The default is one.

4. When MPLS OAM ping or traceroute is done on a LDP tunnel, MPLS echo request is sent out on the first path of the tunnel. Optionally, you can choose which path to send the request by specifying the next-hop IP address.

5. When LDP graceful restart is enabled and VPLS uses LDP tunnel for its transport, MPLS makes sure that the path used to carry VPLS traffic remains unchanged after MP failover and there is no traffic loss observed during the failover.

6. L2VPN application does not have support for LDP ECMP at LER. L2VPN can still use the LDP tunnel; however, all L2VPN traffic is sent out on a single path.

7. ECMP routes cannot have a combination of IP nexthops and MPLS nexthops.

8. Individual paths of a LDP ECMP tunnel can ONLY have outgoing interfaces of type physical, VE or LAG. LDP over GRE or LDP over RSVP ingress functionality is not supported.

9. Ingress tunnel accounting is supported for LDP ECMP tunnels. Statistics for LDP ECMP tunnel aggregate traffic from all the individual paths.

10. LDP ECMP LER tunnels cannot use GRE or RSVP tunnel interfaces.

Interactions with other features

LDP ECMP tunnel accounting

The **show mpls statistics ldp tunnel** command is used to retrieve statistics for particular LDP tunnel. MPLS module on MP collects packet counts for all the paths in the ECMP from all line cards and adds them together before displaying the result to the user.

There is an existing configurable option **exclude-ethernet-overhead** available for tunnel ingress accounting which is used to exclude the Ethernet overhead bytes from the total byte statistics retrieved for an LDP tunnel. With ECMP paths for LDP tunnel, some paths can be over tagged interfaces and some can be over untagged interface. To keep consistency in the counting of overhead bytes, it is mandatory for the tunnel statistics be cleared after changing **exclude-ethernet-overhead** mode for ingress tunnel accounting. To enforce this, a confirmation message is added to the command to change the **exclude-ethernet-overhead** and the command now clears the counters automatically.

Sflow support for MPLS LER

Sflow sample for traffic sent over LDP tunnel includes the tunnel ID and not the tunnel's out segment information. Adding the support for LDP ECMP tunnel does not have any impact on Sflow sample.

Setting the LDP Hello Interval and Hold Timeout values

The LDP Hello interval and Hello Hold Timeout timers are used to establish Hello Adjacency between peers. The Hello Interval is the time period between which the LSR sends out Hello messages and the Hello Hold Timeout value is the amount of time that the sending LSR maintains its record of Hellos from the receiving LSR without receipt of another Hello message.

The Hello interval and Hello Hold Timeout timer values can be obtained from the global default values, configured globally on a router, or in the case of the Hello Hold Timeout timer configured per-interface. When configuring these values the following constraints must be followed:

- The Hello Interval value must be < 32767
- The Hello Hold Timeout value must be <65535
- The Hello Hold Timeout value must be >= 2 * Hello Interval value

As described in the following sections, values can be set that determine the values used on the configured router and values sent to adjacent peers for their configuration:

- Setting the LDP Hello Interval and Hold Timeout Values
- Setting the LDP Hold Time Sent to Adjacent LSRs
- Determining the LDP Hold Time on an MPLS Interface

Setting the LDP Hello interval values

The LDP hello interval controls how often the device sends out LDP Hello messages. Hello messages are used to maintain LDP sessions between the device and its LDP peers. The user can set the interval for LDP Link Hello messages (LDP Hello messages multicast to all routers on the sub-net), as well as for LDP Targeted Hello messages (LDP Hello messages unicast to a specific address, such as a VLL peer):

- **For targeted LDP sessions –** the LDP Hello Interval can only be set globally. This configuration is described in Setting the LDP Hello Interval globally for targeted LDP sessions on page 234. When a Hello Interval is not set for Targeted LDP sessions, then the global default value is used.
- **For link LDP sessions –** the LDP Hello Interval can be set globally which applies to all LDP interfaces or on a per-interface basis. The LDP Hello Interval values in Link LDP sessions are determined by the following procedure in the order described below.

1. When the Hello Interval is set per-interface, that value is used. This configuration is described in Setting the LDP Hello Interval per-Interface (link only) on page 234.
2. When the Hello Interval is not set per-interface, then the value set for LDPs globally is used. This configuration is described in Setting the LDP Hello Interval globally for link LDP sessions on page 234.
3. When the Hello Interval is not set either globally or per-interface, the global default value is used.

   When a Hello Adjacency already exists, the adjacency remains up and any new configured interval takes effect upon the expiration of the current Hello Interval timer. Consequently, the next and subsequent hello messages are sent at the new interval.

Setting the LDP Hello Interval globally

The user can set a global LDP Hello Interval that applies to all LDP sessions, regardless of interface. This is performed separately for Link and Targeted LDP sessions as described in the following sections:

- Setting the LDP Hello Interval Globally for Link LDP Sessions

- Setting the LDP Hello Interval Globally for Targeted LDP Sessions

Setting the LDP Hello Interval globally for link LDP sessions

To set the interval for LDP Link Hello messages to 10 seconds, enter the following command.

```
device(config-mpls)# ldp
device(config-mpls-ldp)# hello-interval 10
```

Syntax: [no] hello-interval *seconds*

The *seconds* variable specifies the value in seconds of the Hello Interval that the user is globally configuring for LDP Link Hello messages. The LDP hello interval can be from 1 - 32767 seconds. The default value for LDP Link Hello messages is five seconds.

The value set here can be overridden on a per-interface basis as described in Setting the LDP Hello Interval per-Interface (link only) on page 234.

The [no] option removes a previously configured LDP Link Hello Interval.

Setting the LDP Hello Interval globally for targeted LDP sessions

To modify the hello message interval for targeted LDP sessions to 20 seconds, enter the **hello-interval target** command.

```
device(config-mpls)# ldp
device(config-mpls-ldp)# hello-interval target 20
```

Syntax: [no] hello-interval target *seconds*

The *seconds* variable specifies the value in seconds of the hello interval that the user is globally configuring for LDP Targeted messages. The LDP hello interval can be from 1 - 32767 seconds. When the user sets a new LDP hello interval, it takes effect immediately. The default value for LDP Targeted Hello messages is 15 seconds.

The [no] option removes a previously configured LDP Targeted Hello Interval.

> NOTE
> This value can only be set globally for all Targeted LDP sessions on the router. Per-interface configuration is only available for Link LDP sessions.

Setting the LDP Hello Interval per-Interface (link only)

The user can set the LDP Hello Interval on a per-Interface basis. This option is only available for Link LDP sessions. The following example configures the MPLS Interface at Ethernet port 1/1 with a **hello-interval** of 10 seconds.

```
device(config)# mpls
device(config-mpls)# mpls-interface ethernet 1/1
device(config-mpls-if-e100-1/1)# ldp-params
device(config-mpls-if-e100-1/1-ldp-params)# hello interval 10
```

Syntax: [no] hello-interval *seconds*

The *seconds* variable specifies the value in seconds of the Hello Interval that the user is configuring on this MPLS interface for LDP Link Hello messages.

No default value exists for this parameter. When a value is set here, it overrides any LDP Hello Interval that was globally configured. However, when no value is set for this parameter, it defaults either to the LDP Hello Interval that was configured globally or, when no value was configured globally, to the default global value. For information about the global configuration, refer to Setting the LDP Hello Interval globally for link LDP sessions on page 234.

The [no] option removes a previously configured LDP Hello Interval.

Setting the LDP hold time sent to adjacent LSRs

The LDP hold time specifies how long the device waits for its LDP peers to send a Hello message. When the device does not receive a Hello message within this time, the LDP session with the peer can be terminated. The device includes the hold time in the Hello messages it sends out to its LDP peers.

The LDP Hold Time sent in Hello messages to adjacent LSRs can be configured globally for either Link or Targeted LDP sessions, as described in the following sections:

- Setting the LDP Hello hold time sent to adjacent LSRs for link LDP sessions
- Setting the LDP Hello hold time sent to adjacent LSRs for targeted LDP sessions

Setting the LDP Hello hold time sent to adjacent LSRs for link LDP sessions

To set the hold time included in LDP Link Hello messages to 20 seconds, enter the **hello-timeout** command.

```
device(config-mpls)# ldp
device(config-mpls-ldp)# hello-timeout 20
```

Syntax: [no] hello-timeout *seconds*

The *seconds* variable specifies the value in seconds of the LDP hello timeout that is sent in Hello messages to Link LDP peers. The range for this value is 1 - 65535 seconds. The default value is 15 seconds.

When the user globally sets a LDP hold time, the new time takes effect immediately and goes in the next Hello message sent. This hold time applies to only the hold time that the device sends to its peers; it does not affect the hold time the device uses to time out those peers. The latter is determined from the hold time that peers send to the device.

The [**no**] option removes a previously configured LDP Hello Timeout value and returns the value to the default.

Setting the LDP Hello hold time sent to adjacent LSRs for targeted LDP sessions

To set the hold time included in LDP targeted Hello messages to 60 seconds, enter the following command.

```
device(config-mpls)# ldp
device(config-mpls-ldp)# hello-timeout target 60
```

Syntax: [no] hello-timeout target *seconds*

The *seconds* variable specifies the value in seconds of the LDP hello timeout that is sent in Hello messages to targeted LDP peers. The LDP hold time can be from 1 - 65535 seconds. The default value is 45 seconds.

The [**no**] option removes the previous LDP Hello timeout target value and returns the value to the default.

Determining the LDP hold time on an MPLS interface

An MPLS interface uses the LDP Hello hold time to determine how long it waits for its LDP peers to send a Hello message. How this determination is made differs for a targeted LDP session and a link LDP session, as follows:

- **For targeted LDP sessions** - The value received in Hello messages from its peers determines the time that the device waits for its LDP peers to send a Hello message. When the timeout value received from a peer is zero, the Hold time is set to the default period of 45 seconds.

- **For link LDP sessions** - In this case, the wait time is determined by any one of the below criteria.

1. When the Hello hold time is set per-interface, that value is used. That value is set as described in Setting the LDP Hello Holdtime per-interface (link only)

2. When the Hello hold time is not set per-interface, the hold time in the received message is used.

3. When the Hello hold time in the received message is zero (0), the default value of 15 seconds is used.

Setting the LDP Hello Holdtime per-interface (link only)

The user can set the LDP Hello Holdtime on a per-interface basis. This holdtime value is sent in Hello messages from the interface. This option is available for Link LDP sessions only. The following example configuration is for the MPLS Interface at Ethernet port 1/3 with a **hello-timeout** of 18 seconds.

```
device(config)# mpls
device(config-mpls)# mpls-interface ethernet 1/3
device(config-mpls-if-e100-1/3)# ldp-params
device(config-mpls-if-e100-1/3-ldp-params)# hello-timeout 18
```

Syntax: [no] hello-timeout *seconds*

The value configured in the *seconds* variable is the LDP Hello Timeout value that are sent in LDP Hello messages from this interface. The minimum value that can be configured for this variable is 2 * the value set for the Hello Interval.

The [**no**] option removes a previously configured LDP Hello Timeout value and sets the value as described in Determining the LDP hold time on an MPLS interface on page 235.

LDP message authentication

The Multi-Service IronWare software supports LDP authentication based upon the TCP MD5 signature option specified in *RFC 2385*. This RFC defines a new TCP option for carrying an MD5 digest in a TCP segment. The purpose of this feature is to protect against spoofed TCP segments in a connection stream.

Configuring LDP message authentication

Extreme devices allow configuration of an authentication key on a per LDP session basis. The LDP session can be to an adjacent peer (basic discovery) or to the targeted peer (extended discovery). This feature must be configured on both sides of an LDP peer link. To configure LDP message authentication use the following commands.

```
device(config)# mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# session 10.10.10.3 key early
```

Syntax: [no] session *remote-ip-addr* key *string*

The *remote-ip-addr* variable specifies the IP address of the LDP peer that authentication is being configured for.

The *string* variable specifies a text string of up to 80 characters used for authentication between LDP peers. It must be configured on both peers.

By default, **key** is encrypted. When the user wants the authentication key to be in clear text, insert a **0** between **key** and *string*.

```
device(config-mpls-ldp)# session 10.10.10.3 key 0 early
```

The software adds a prefix to the key string in the configuration. For example, the following portion of the code has the encrypted code "2".

```
session 10.1.1.1 key
2 $XkBTb24tb0RuXA==
```

The encrypted code can be one of the following:

*   0 = the key string is not encrypted and is in clear text

- 1 = the key string uses proprietary simple cryptographic 2-way algorithm (only for CES 2000 Series devices)
- 2 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for XMR Series and MLX Series devices)

Resetting LDP neighbors

The user can terminate and re-establish an MPLS LDP neighbor session when at least one LDP "hello" adjacency exists with the peer. When the LDP session terminates, the database associated with the LDP session is also cleared. When the session re-establishes, the session-specific information is re-learned from its peer:

- LDP downstream and upstream label database (**show mpls ldp database...**)
- LDP label switched path (**show mpls ldp path ...**)
- LDP peer (**show mpls ldp peer ...**)
- LDP created MPLS tunnels (**show mpls ldp tunnel ...**)
- LDP FECs learned from the resetting neighbor sessions (**show mpls ldp fec ...**). FECs are not cleared immediately but are marked that no LDP session exists.

To reset or clear an MPLS LDP neighbor session, enter the **clear mpls ldp neighbor** command.

Syntax: **clear mpls ldp neighbor** [**all** | *peer-ip-addr* [**label-space-id** *label-space*]]

When the **all** option is specified, all LDP sessions on the Extreme device is reset, including the targeted LDP sessions.

An LDP session is uniquely referred to by *peer-ip-addr: label-space*. This command also allows the user to input *peer-ip-addr* only and ignore *label-space*. In this case, all LDP sessions with the matching peer address is reset.

Executing this command displays a warning message when the LDP session is not found corresponding to the supplied *peer-ip-addr* (and *label-space*). When an LDP session is not in operational state, resetting it has no impact.

Resetting LDP neighbor considerations

The **clear mpls ldp neighbor** feature terminates the specified LDP sessions. The LDP sessions are automatically reestablished when at least one "hello" adjacency exists with the neighbor, and LDP configuration remains unchanged. This command allows a user to reset the following LDP sessions:

- Platform-wide label space
- Interface specific label space

When an LDP session is terminated as a result of the **clear mpls ldp neighbor** command, the Extreme device does not generate any notification message for the neighbor. Instead, the device unilaterally terminates the session and close the associated TCP session. The other end of the LDP session detects this reset operation in either of the following two ways:

- TCP session is broken (half connected). The device detects this while receiving or sending LDP messages on TCP socket fails (with fatal error), indicating that underlying TCP session is aborted by remote peer.
- Receives a new TCP connection request from the neighbor while the older session is still operational (when this is in the passive role).

   NOTE
   Either of the above events trigger the remote end of the LDP session to tear down the session and try to reestablish. Resetting an LDP session impacts the associated VPLS/VLL sessions. Resetting an LDP session which is not in an operational state has no impact.

Validating LDP session reset

The user can check the following LDP session specific parameters to validate that a session has been successfully reset:

- The LDP session state transitions from "Operational" to "Nonexistent" upon clearing it. It may quickly transition from "Nonexistent" to "Operational." In that case, the show mpls ldp session [**detail** | *A.B.C.D*] shows the "Up time", and that must have been reset to zero upon clearing the session.

- The LDP session specific database (mentioned above) is cleaned upon resetting the LDP session.

- The TCP port number (on the active end of the LDP session) may have been changed once the LDP session comes up after reset. In other words, the TCP port number before the reset and after the reset may be different. Use the command **show mpls ldp session** [**detail** | *A.B.C.D*] to view the TCP port number.

- Syslog logs the event of a LDP session going down and then coming back up, as a result of resetting the LDP session. Use the command **show log** to view the syslog events.

Following is an example of how to use the **show log** command to view the syslog.

```
device# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 33 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):
Sep  9 18:38:20:N:MPLS: LDP entity session 10.1.1.1:0 with peer 10.2.2.2:0 is up
Sep  9 18:38:02:N:MPLS: LDP entity session 10.1.1.1:0 with peer 10.2.2.2:0 is down
```

The following command shows two LDP sessions with neighbor 10.234.123.64.

```
device# show mpls ldp session
Peer LDP ID              State         Adj Used  My Role   Max Hold  Time Left
10.234.123.64           Operational   Link      Passive   36        33
```

The following command clears both the link and targeted LDP session with neighbor 10.234.123.64, because the *label_space* optional parameter has not been specified.

```
device# clear mpls ldp neighbor 10.234.123.64
device#
device# show mpls ldp session
Peer LDP ID              State         My Role   Max Hold  Time Left
10.234.123.64           Operational   Passive   36        33
device#
```

This command shows that after waiting for roughly 20 seconds (depends on the hello or keepalive timer periodicity), both the LDP sessions are reestablished.

```
device# clear mpls ldp neighbor 10.234.123.64
Peer LDP ID              State         My Role   Max Hold  Time Left
10.234.123.64           Operational   Passive   36        33
```

The user can also validate the **clear mpls ldp neighbor** command using the syslog command.

```
device# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 47 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):
Sep 9 19:23:24:N:MPLS:LDP entity session 10.2.2.2:0 with peer 10.234.123.64 is up
Sep 9 19:23:08:N:MPLS:LDP entity session 10.2.2.2:10 with peer 10.234.123.64 is down
Sep 9 19:23:08:N:MPLS:LDP entity session 10.2.2.2:0 with peer 10.234.123.64 is down
```

MPLS LDP-IGP synchronization

Packet loss can occur because the actions of the IGP and LDP are not synchronized.

The MPLS LDP-IGP synchronization feature provides the following benefits:

- Provides a means to synchronize LDP and IGPs to minimize MPLS packet loss
- MPLS LDP-IGP synchronization may be enabled per interface, or globally
- OSPF and IS-IS are supported for the IGP; each operates independently
- LDP determines convergence (receipt of all labels) for a link through one of two methods
  - Receive Label silence mechanism
  - End Of Lib mechanism (*RFC 5919*)
- Provides a means to disable LDP-IGP synchronization on interfaces that the user does not want enabled
- Enables the user to globally enable LDP-IGP synchronization on each interface associated with an IGP *Open Shortest Path First (OSPF)* or IS-IS process

FIGURE 50 Example with LDP IGP synchronization
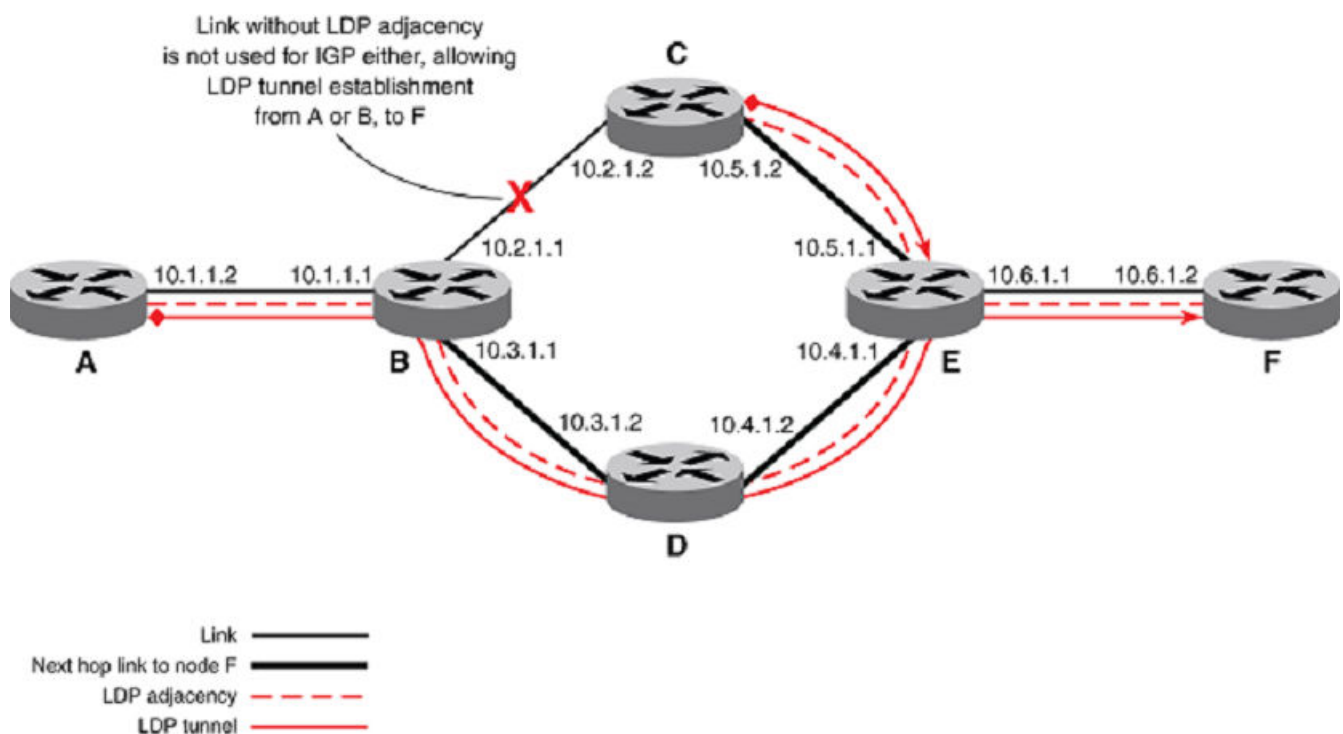
To enable LDP-IGP synchronization on each interface that belongs to an OSPF or IS-IS process, enter the **ldp-sync** command. When the user does not want all of the interfaces to have LDP-IGP synchronization enabled, issue the **no ldp-sync** command on the specific interfaces.

To specify how long an IGP must wait for LDP synchronization to be achieved, enter the **ldp-sync holddown** command in the global configuration mode. When the LDP peer is not reachable, the IGP establishes the adjacency to enable the LDP session to be established.

When an IGP adjacency is established on a link but LDP-IGP Synchronization is not yet achieved or is lost, the IGP advertises the max-metric on that link.

Configuration considerations

- Supports only point-to-point interfaces but not tunnel interfaces
- On IS-IS, wide metric-style is required
- When enabled on IS-IS, the feature applies to both level-1 and level-2 metrics
- Affects IPv4 metrics only

Configuring MPLS LDP-IGP Synchronization

This section contains the following tasks:

- Configuring MPLS LDP-IGP Synchronization with OSPF Interfaces (required)
- Selectively Disabling MPLS LDP-IGP Synchronization from Some OSPF Interfaces (optional)
- Verifying MPLS LDP-IGP Synchronization with OSPF (optional)
- Configuring MPLS LDP-IGP Synchronization with IS-IS Interfaces (required)
- Selectively Disabling MPLS LDP-IGP Synchronization from Some IS-IS Interfaces (optional)
- Verifying MPLS LDP-IGP Synchronization with IS-IS (optional)

Configuring MPLS LDP-IGP synchronization globally

MPLS LDP-IGP synchronization is disabled by default. To globally enable MPLS LDP-IGP synchronization with IS-IS, enter the following commands.

```
device(config)# router isis
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-router-ipv4u)# metric-style wide
device(config-isis-router-router-ipv4u)# ldp-sync
device(config-isis-router-router-ipv4u)# exit
device(config-isis-router)#
```

MPLS LDP-IGP synchronization is disabled by default. To globally enable MPLS LDP-IGP synchronization with OSPF, enter the following commands.

```
device(conf)# router ospf
device(conf-ospf-router)# ldp-sync
device(conf-ospf-router)#
```

Syntax: [no] ldp-sync

Setting the LDP IGP sync hold down time

The **ldp-sync hold-down** command sets the LDP-IGP sync hold down time. The hold down time (in router OSPF and the router IS-IS modes) is the interval which the IGP must advertise the maximum IP metric, while waiting for an update from LDP.

The hold down interval starts whenever the IGP initially is enabled with LDP-IGP sync. It is also started whenever LDP updates the IGP with an update indicating the interface status, from LDP's perspective, is not-in-sync. When the hold down time expires, the IGP resumes advertising the normal metric for the link.

When hold down time is configured (from no hold down time), the router starts the hold-down-timer on every interface that is not-in-sync at the time.

When hold down time is un-configured, the router stops the hold-down-timer on every interface that has hold-down-timer running at the time as if there is no hold down time configured. As a result, these interfaces have infinite hold down time. For those not-in-sync interfaces with hold-down time already expired, IGP continues to advertise Normal metric.

By default, hold-down time is disabled. IGP waits until LDP gives an In Sync indication for the link before it advertised the normal metric.

By default, **ldp-sync hold-down** is disabled. To enable the **ldp-sync hold-down** timer with IS-IS, enter the following commands.

```
device(conf)# router isis
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-router-ipv4u)# metric-style wide
device(config-isis-router-router-ipv4u)# ldp-sync
device(config-isis-router-router-ipv4u)# ldp-sync hold-down 100
```

By default, **ldp-sync hold-down** is disabled. To enable the **ldp-sync hold-down** timer with OSPF, enter the following commands.

```
device(conf)# router ospf
device(conf-ospf-router)# ldp-sync
device(conf-ospf-router)# ldp-sync hold-down 100
```

Syntax: **ldp-sync hold-down** *seconds*

The *seconds* parameter range is 1 to 65535 seconds.

Enabling LDP sync on an interface

Use the **isis ldp-sync** command under the **conf-if-e-1/1** policy to enable the LDP sync feature on a specific IS-IS interface. This overrides the global setting from the MPLS LDP-sync feature. By default, the **isis ldp-sync** is not enabled individually on an interface.

```
device(conf)# interface e 1/1
device(conf-if-e-1/1)# ip router isis
device(conf-if-e-1/1)# isis ldp-sync enable
```

Syntax: **isis ldp-sync** [**enable** | **disable**]

Use the **ip ospf ldp-sync** command under the **conf-if-e-1/1** policy to enable the LDP sync feature individually on an OSPF interface. By default, the **ip ospf ldp-sync** is not enabled individually on an OSPF interface.

```
device(conf)# interface e 1/1
device(conf-if-e-1/1)# ip ospf area 0.0.0.0
device(conf-if-e-1/1)# ip ospf ldp-sync enable
```

Syntax: **ip ospf ldp-sync** [**enable** | **disable**]

Setting the receive label silence timer

When labels are not received from the peer for a short period of time, the session is declared 'In Sync'. When a label is received from a peer, then the 'receive label silence timer' is reset.

Use the **rx-label-silence-time** command under **config-mpls-ldp policy** to define the length of the receive label silence timer.

```
device(conf)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# rx-label-silence-time 30000
```

Syntax: **rx-label-silence-time** *value*

The *value* parameter specifies the length of time of the receive label silence timer in milliseconds. Possible values are from 100 to 60000 milliseconds. The default value is 1000.

Enabling the end-of-lib submode

Configure the **end-of-lib** submode under LDP to contain all the attributes of the end of lib capability and notification.

```
device(conf)# router mpls
device(conf-router-mpls)# ldp
device(conf-router-mpls-ldp)# end-of-lib
device(conf-router-mpls-ldp-eol)#
```

Disabling the end-of-lib submode

Enabling the **end-of-lib** submode determines whether the two RFCs, *RFC 5561* and *RFC 5919* are enabled by the LSR. The user can turn this feature off either by;

- Removing the end-of-lib submode.

- Issuing the **disable** command under the **end-of-lib** submode.

```
device(conf)# router mpls
device(conf-router-mpls)# ldp
device(conf-router-mpls-ldp)# end-of-lib
device(conf-router-mpls-ldp-eol)# disable
```

Syntax: [no] end-of-lib

The [**no**] form of "disable" enables the feature.

Setting the EOL notification timer

Use the **EOL notification timer** command under the **conf-router-mpls-ldp-eol** policy to set the length of the EOL notification timer. This command is LDP global.

```
device(conf)# router mpls
device(conf-router-mpls)# ldp
device(conf-router-mpls-ldp)# end-of-lib
device(conf-router-mpls-ldp-eol)# notification-timer value
```

Syntax: EOL notification timer *value*

The *value* parameter specifies the length of the EOL notification timer in milliseconds. Possible values are from 100 to 120000 milliseconds. The default value is 60000.

Setting the EOL transmit label silence timer

Use the **tx-label-silence-timer** command under **conf-router-mpls-ldp-eol** policy to sets the length of the EOL transmit label silence timer. This command is LDP global.

```
device(config)# router mpls
device(config-router-mpls)# ldp
device(config-router-mpls-ldp)# end-of-lib
device(config-router-mpls-ldp-eol)# tx-label-silence-timer 2000
```

Syntax: tx-label-silence-timer *value*

The *value* parameter specifies the length of the EOL transmit label silence timer in milliseconds. Possible values are from 100 to 60000 milliseconds. The default value is 1000.

MPLS failover support for VPLS

VPLS can preserve its forwarding state across MP failover. Therefore, VPLS using LDP tunnels as its transport mechanism benefits from the LDP GR support. LDP GR recovers both the LDP tunnel labels and the VC labels for VPLS. This is supported for all local switching traffic as well as traffic switched between VPLS peers as long as the peer uses LDP tunnels and that LDP GR is configured.

> **NOTE**
> When MCT is configured for the VPLS instance involved then VPLS preserving its forwarding state between the VPLS peers is not guaranteed.

LDP failover support for transit

LDP GR preserves the LDP transit cross-connects. Therefore, it minimizes the traffic loss of any application that uses an LDP tunnel as its transport mechanism from the transit LSR perspective.

LDP Graceful Restart (GR)

This section describes the functionality of the LDP Graceful Restart feature based on *RFC 3478* (Graceful Restart mechanism for Label Distribution Protocol).

LDP *Graceful Restart (GR)* helps minimize MPLS traffic loss when an LDP component is restarting in a router that is capable of preserving its MPLS forwarding states across restart. LDP GR works between a router and its neighbor and its capability must be advertised when sending an LDP Initialization message.

An LDP restart triggered by MP failover due to a fault of the active MP or user-commanded switchover is the only scenario where the MPLS forwarding state is preserved.

The router can also support LDP GR in helper-only mode. In this mode, a router does not preserve its forwarding entries on a LDP GR restart, however, it can help a neighboring router recover its forwarding entries when the neighbor is going through restart.

A NetIron router implementing LDP GR can play one of the two roles:

- A restarting LSR: An LSR that performs LDP restart. A standby MP must be up on a MLX Series or XMR Series router acting as a restarting LSR for LDP GR to work.
- A GR helper (helper-only mode): An LSR whose neighbor is restarting its LDP component.

> **NOTE**
> MLX Series or XMR Series routers can play either role in an LDP GR procedure. CES 2000 Series and CER 2000 Series routers support only the helper-only mode.

When LDP GR is enabled on a router, the configuration does not apply to the current sessions. The LDP GR configuration is applied for the new sessions brought up after the configuration is added.

Graceful restart procedure

The following section describes the restart procedures of an LSR and a GR helper LSR.

Procedure for the restarting LSR

After an LSR restarts its LDP components, when its MPLS forwarding state is not preserved (as in the case of the CES 2000 Series and CER 2000 Series routers and routers in helper-only mode), it sends out the FT TLV with Recovery Time set to 0 in the LDP Initialization message to its neighbor.

When the MPLS forwarding state has been preserved across the restart, the LSR does the following:

1. Start the Forwarding State Holding timer.
2. Mark all the MPLS forwarding entries as "stale".
3. Set the Recovery Time to the current value of the Forwarding State Holding timer when it sends out LDP Initialization message to its neighbor.

   When the timer is not expired the LSR uses the labels and next-hop information received from the neighbor to lookup and clear the stale flag for the corresponding label-FEC entries. When the timer is expired, all the entries that are still marked as "stale" are deleted and the LDP GR procedure is completed.

Procedure for the GR helper LSR

When the LSR detects that its LDP session with a neighbor went down and the neighbor is capable of preserving its forwarding state, the LSR does the following:

1. Retain the label-FEC bindings received by way of the session and mark them as "stale".
2. Start the Reconnect timer with the timeout value set to the lesser of the peer FT Reconnect Timeout and the locally configured maximum Reconnect timeout.
3. Attempt to re-establish LDP session with the neighbor using the normal LDP procedure.

   All the stale label-FEC bindings are deleted when either condition is true:

   • The Reconnect timer has expired and the LDP session to the neighbor is not established.
   • LSR receives FT TLV in the Initialization message from the neighbor and the FT Recovery Time is set to 0.

   After the session is re-established, the LDP GR helper resends Label Mappings to its neighbor. For the stale label-FEC bindings received from the neighbor, they are recovered during the recovery period which is set to the lesser of the peer Recovery Timeout and the locally configured maximum recovery time. If the stale entries are not recovered after the Recovery Timer has expired, they are deleted.

Session down detection on GR helper

A LDP GR enabled router goes into helper-only mode (GR helper) when any of the following events occur on the router's neighbors.

• MP failover occurs
• HLOS upgrade occurs
• Remove and re-add of the MPLS configuration
• TCP communication broken (such as, session KeepAlive timer expires)
• UDP communication broken (example: adjacency goes down)
• Restarting LDP component by disabling and enabling the loopback
• Restarting a LDP session by issuing the **clear mpls ldp neighbor** command

In helper-only mode, the LDP GR procedure works at the session level. Any of the above events causes the helper to detect session down and start the GR procedure. The operation of the GR helper is the same independent of what has happened on the restarting LSR that triggers the GR procedure.

Graceful Restart scenarios

Re-advertise label to its upstream neighbors

When the restarting router, acting as a transit LSR, can recover a FEC based on the Label Mapping it receives from its GR helper, and the local forwarding state successfully, it re-advertises the same label to all of its upstream neighbors.

As part of supporting GR, the Label Management component also makes sure that those labels that are used to advertise to upstream neighbors before GR happens is not re-used for the new LSP coming up while GR is in-progress. However, when the previously used label is released because the LSP has gone down during GR, the label can be re-allocated for the new LSP.

Clearing mpls ldp neighbors

For **clear mpls ldp neighbor**, the configured reconnect and recovery timer values is sent to the peer when both are configured with LDP graceful restart. Note that in this scenario, both routers are acting in helper-only mode. Therefore, after the session comes back up, both routers exchange their bindings and go through the recovery procedure. There is no traffic loss when the reconnect and the recovery timers do not time out.

On a CES 2000 Series, CER 2000 Series, MLX Series or XMR Series configured for helper-only mode, the **clearing mpls ldp neighbor** command results in immediate reconnect timeout at the remote end. Therefore, in this scenario all bindings at the remote end associated with the session are deleted due to reconnect time out. On the local node the recovery timer of zero results in immediate clearing of the forwarding entries.

Ingress LSR specific processing

VPLS supports failover and must preserve its forwarding state when the LDP tunnel is used to carry VPLS traffic until the GR has finished. LDP GR attempts to recover both the tunnel labels and the VC labels. In the case where a VC label cannot be recovered, the corresponding PW is brought down after GR has finished. In the case where the LDP tunnel cannot be recovered, all the PWs using the LDP tunnel is brought down due to tunnel down event.

VPLS auto-discovery

BGP GR does not support the L2VPN address family type. Therefore, BGP-based auto-discovery peer is not preserved across MP failover, even though BGP GR is enabled. As a result, LDP GR preserves the VC label associated with a VPLS auto-discovery peer only when the peer is relearned during the LDP GR recovery.

Other applications

On ingress LSR, LDP tunnels are preserved as part of LDP GR (helper-only mode excluded), this does not benefit non-L2VPN applications (example: IPoMPLS, L3VPN, PBR) that do not support MP failover. There is no coordination between MPLS and those applications attempting to preserve its CAM entries. Whether its corresponding CAM entries are deleted and re-added or updated is not guaranteed by LDP GR support.

Transit LSR specific processing

For those LDP cross-connects that can be recovered as part of LDP GR, there is no traffic loss for those application using those tunnels if and only if the GR helper (example: downstream neighbor) re-advertises the same label and upstream neighbor also support LDP GR procedure as well.

LDP ECMP (transit only)

The ability to preserve the LDP ECMP transit cross-connects depends on the route information received from RTM during the recovery phase. In the case where the number of ECMP provided by RTM for a route is larger than the LDP load-sharing configuration (for example, the IP load-sharing configuration is larger than LDP load-sharing configuration), the paths are preserved as long as the route provided by RTM, before recovery time expires, contains the installed paths.

LDP over RSVP (transit only)

RSVP GR is not supported. Therefore, when an RSVP tunnel goes down due to MP failover, LDP cross-connects using the RSVP tunnel is not preserved as part of LDP GR support.

LDP over GRE (transit only)

LDP GR treats GRE tunnel interfaces as regular physical interfaces. When the GRE tunnel interface up indications and route using the GRE tunnels are received before the GR recovery timer expired, LDP over GRE tunnel cross-connects is preserved.

Graceful Restart helper-only mode

A router that is configured as GR helper-only indicates to its peers that forwarding state is not preserved by sending an initialization message with the Reconnect Time and the Recovery Time set to zero (0) in FT session TLV.

Configuring LDP graceful restart (GR)

By default LDP GR is disabled. It can be enabled globally under the LDP configuration. When LDP GR is enabled, the CES 2000 Series and CER 2000 Series routers are in helper mode only. The MLX Series and XMR Series routers can act either as a restarting router or a GR helper.

With LDP GR enabled, the router waits until it receives an LDP Initialization message from its neighbor to know whether it must delete its states or start the LDP GR recovery procedure. When LDP GR is enabled, it is applicable to all LDP sessions regardless of the adjacency type exists between the neighbors.

```
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# graceful-restart reconnect-time 150
device(config-mpls-ldp)# graceful-restart recovery-time 240
```

Syntax: [no] graceful-restart [helper-only] [reconnect-time *seconds*] [max-neighbor-reconnect-time *seconds*] [recovery-time *seconds*] [max-neighbor-recovery-time *seconds*]

The **helper-only** option specifies that the LSR acts as a helper-only. In helper mode, the configuration commands for reconnect-time and recovery-time is rejected with informational messages. The [**no**] form of the commands removes the LDP GR helper mode and revert back to full LDP GR mode.

The **reconnect-time** *seconds* option is the amount of time a GR neighbor must wait for the LDP session to be reestablished. This is advertised to the neighbor using the FT Reconnect Timeout field in the FT Session TLV. The default setting is 120 seconds. The available range is 60 to 300 seconds. The [**no**] form of the command reverts the configured value back to the default value.

The **max-neighbor-reconnect-time** *seconds* option is the maximum time this router must wait for a GR neighbor to restore the LDP session. The default setting is 120 seconds. The available range is 60 to 300 seconds. The [**no**] form of the command reverts the configured value back to the default value.

The **recovery-time** *seconds* option is the amount of time this router retains its MPLS forwarding state across restart. This is advertised to the neighbor using the Recovery Time field in the FT Session TLV. The default setting is 120 seconds. The available range is 60 to 3600 seconds. The [**no**] form of the command reverts the configured value back to the default value.

The**max-neighbor-recovery-time** *seconds* option is the maximum amount of time this router waits for a GR neighbor to complete its GR recovery after the LDP session has been reestablished. The default setting is 120 seconds. The available range is 60 to 3600 seconds. The [**no**] form of the command reverts the configured value back to the default value.

Recovery-time must be chosen accordingly taking into account the time it takes for RTM to re-compute the routes and the number of L3 FECs that need to be recovered as part of the LDP GR recovery. This is applicable to GR processing on ingress as well as transit LSRs.

> NOTE
> The reconnect-time and recovery-time commands are not available for CES 2000 Series and CER 2000 Series routers.

LDP GR configuration examples

The following commands only take effect on newly created sessions. For existing sessions, it is required that the sessions be restarted for the new configuration to take effect.

Using default timeout values

Use the **graceful-restart** command to enable LDP GR and use the default timeout values:

```
device(config-mpls-ldp)# graceful-restart
```

Configuring LDP GR timers

Use the following commands to set LDP GR timers before enabling LDP GR.

```
device(config-mpls-ldp)# graceful-restart reconnect-time 150
device(config-mpls-ldp)# graceful-restart recovery-time 240
device(config-mpls-ldp)# graceful-restart
```

Configuring LDP GR helper mode

Use the **graceful-restart helper-only** command to configure LDP GR helper mode.

```
device(config-mpls-ldp)# graceful-restart helper-only
```

LDP Session Keepalive timeout configurations

After an LDP session is established, an LSR maintains the integrity of the session by sending Keepalive messages. The Keepalive timer for each peer session resets whenever it receives any LDP protocol message or a Keepalive message on that session. When the Keepalive timer expires, LDP concludes that the TCP connection is bad or the peer is dead and terminates the session.

Setting the Keepalive timeout

Use the **ka-timeout** command to set the Keepalive interval or the time interval at which the session Keepalive message is sent when no other LDP protocol message is sent to the LDP peer.

To configure the Keepalive timeout or change the timeout value, the user must be in the LDP mode within the MPLS configuration mode. A warning is displayed whenever the **ka-timeout** value is changed as shown below.

```
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# ka-timeout 180
"Please clear LDP sessions for the new KA parameter value to take effect on existing sessions"
```

Syntax: [no] ka-timeout *value*

The *value* parameter specifies the time after which the session is terminated when no Keepalive or LDP protocol message is received. Possible values 1 to 65535 seconds.

Setting the Keepalive intervals

Use **ka-int-count** command to configure the number of ka-intervals after which the session is terminated when no session Keepalive or other LDP protocol message is received from the LDP peer. In the following example, **ka-int-count** is configured when **ka-timeout** is configured.

```
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)#ka-timeout 180
"Please clear LDP sessions for the new KA parameter value to take effect on existing sessions"
device(config-mpls-ldp)# ka-int-count 10
device(config-mpls-ldp)#
```

Syntax: [no] ka-timeout *value*

The *value* parameter specifies the time interval at which the session Keepalive message is sent when no other LDP protocol message is sent to the LDP peer, Possible values 1 to 65535 seconds.

The **ka-interval** and the **ka-timeout** configurations are mutually exclusive. The user may have only one configuration at a time. The user must explicitly remove the configuration for one in order to change to the other configuration.

```
device(config-mpls-ldp)# ka-interval 11
Warning : LDP Session keepalive time changed. Clear sessions for the new value to take effect on existing
sessions
device(config-mpls-ldp)# ka-timeout 40
Error : Please unconfigure ka-interval before configuring the ka-timeout !
device(config-mpls-ldp)#
device(config-mpls-ldp)# no ka-interval 11
Warning : LDP Session keepalive time changed. Clear sessions for the new value to take effect on existing
sessions
device(config-mpls-ldp)#
device(config-mpls-ldp)# ka-timeout 40
Warning : LDP Session keepalive time changed. Clear sessions for the new value to take effect on existing
sessions
```

Configurable LDP router ID overview

LDP protocol uses LDP messages to communicate between LDP peers for correct functioning of LDP protocol. All LDP messages contains a LDP header which is composed of LDP version, length of message, LDP ID, followed by message. The LDP ID for LDP protocol is composed of LSR-ID and label space. A valid IP address is selected as an LSR-ID field.

Using this feature users are able to specify an IP address of their choice to used as LSR-ID for LDP protocol.

Background

When there is no valid IP address available to be selected as LSR-ID, the LDP protocol continues to remain disabled until a valid IP address is configured on an enabled loopback interface.

FIGURE 51 LDP Header format

LDP Header Format

Version	PDU Length
LDP Identifier	
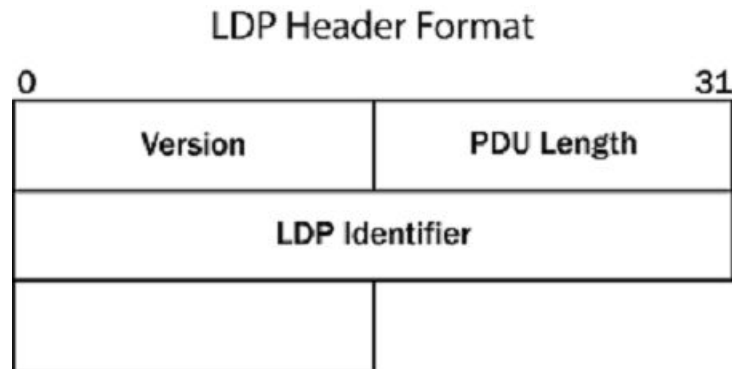
- LDP identifier: LSR-ID: label space
- LSR-ID: First available loopback interface (current behavior)

Configurable LDP LSR-ID allows the user to configure an IP address of their choice as an LSR-ID for LDP protocol.

Once user configure the feature with a valid IP address, LDP protocol must use the feature's configured value as the LSR-ID. In order to compel LDP protocol to use the new value as the LSR-ID, LDP protocol restarts.

The LDP protocol uses the new IP address specified by feature as LSR-ID only when this IP address is configured on one of the enabled loopback interfaces. When this IP address is not configured in enabled state on any of the loopback interface, LDP protocol will continues in the disabled state. LDP protocol will be enabled as soon as this IP address is configured on one of the enabled loopback interfaces.

When the user decides to disable the feature, the LSR-ID selection procedure falls back to default behavior of selecting an LSR-ID for LDP protocol when LDP protocol is enable.

Customer configuration scenarios

Feature behavior is illustrated in different scenarios. When "feature" is mentioned, it means "configurable LDP LSR-ID" feature and its configuration.

1. a) **Precondition:** LDP protocol has default configuration.

   b) **Action:** Feature is enabled with an IP address.

   c) **Post condition:** When the LDP protocol is already using the same IP address specified by the feature as LSR-ID for LDP protocol, LDP continues to work without restarting LDP protocol. This is because LDP protocol is already using the LSR-ID user intend to configure for LDP protocol.

   When LDP protocol is using another value as LSR-ID, LDP protocol restarts. During restart, the LSR-ID selected for LDP protocol is the same IP address as set by the feature. In cases where the feature configured value is not present on any of the enabled loopback interfaces, LDP protocol continues in the disabled state.

2. a) **Precondition:** LDP protocol is disabled and the feature is configured.

   b) **Action:** Feature configured IP address is configured on one of enabled loopback interface, such as when user configures IP address to be used as LSR-ID as on one of the loopback interface.

   c) **Post condition:** LDP protocol is enabled with LSR-ID as configured for the feature.

3. a) **Precondition:** LDP protocol is enabled and feature is configured.

    b) **Action:** IP address same as LSR-ID is deleted from loopback interface OR loopback interface containing IP address configured by feature is disabled, or deleted.

    c) **Post condition:** LDP protocol is disabled and continues to remain in that state until the feature configured IP address is reconfigured on one of the enabled loopback interfaces.

4. a) **Precondition:** LDP protocol is enabled and the feature is configured.

    b) **Action:** The feature configuration is changed to another address.

5. a) **Precondition:** LDP protocol is enabled and the feature is configured.

    b) **Action:** The feature is disabled.

    c) **Post condition:** The LDP protocol will continue to function without restart but features functionality will be disabled. Please make a note that LSR-ID selection process will fall back to default behavior when attempt is made to restart LDP protocol. Since no attempt is made to restart LDP protocol, the LSR-ID for LDP protocol will continue to remain UP at the time of disabling the feature. The reason for this behavior is to minimize the number of restart for LDP protocol.

6. a) **Precondition:** LDP protocol is disabled and the feature is configured.

    b) **Action:** The feature is disabled.

    c) **Post condition:** An attempt is made to restart the LDP protocol. Because the attempt is made after disabling the feature, LSR-ID is selected with the default behavior and the LDP protocol is enabled.

Limitations

- You can not configure value 0.0.0.0. If you try to configure the feature with this value, the feature rejects the configuration.
- You can only configure IPv4 addresses.

Upgrade and downgrade considerations

LDP protocol selects the first operationally UP IP address among the loopback interfaces as LSR-ID for LDP protocol. Once you enable the configurable LDP LSR-ID using valid IP address, LSR-ID selection process changes.

When user decides to disable the feature, the LSR-ID selection process falls back to its default behavior, which is the current behavior of the NetIron products.

LDP over RSVP (for transit LSR only)

LDP over RSVP (for transit LSR only) enables LDP traffic to tunnel across RSVP tunnels. The RSVP tunnel is the transit of the LDP tunnel. On XMR Series and MLX Series devices, LDP over RSVP can run over all types of LSPs (for example, one-to-one or facility *Fast ReRoute (FRR)* LSPs, adaptive LSPs, or redundant LSPs).

> **NOTE**
> LDP over RSVP configuration (for transit LSR only) is supported on all Extreme devices. LDP over facility FRR LSPs is not supported on CES 2000 Series and CER 2000 Series devices.

LDP over RSVP is supported for all cases except when an Extreme device acts as a *Label Edge Router (LER)* for both LDP and RSVP. On the transit for LDP, the RSVP tunnel (RSVP LSP with LDP tunneling enabled) is used to reach the next-hop. The RSVP tunnel is treated as a single hop, and thus external LDP FECs are not advertised to the LSRs which are part of the RSVP core.

LDP depends on the *Routing Table Manager (RTM)* to provide the best next-hop for a particular prefix when LDP decides which label (received from its downstream peers) must be installed. This does not change for LDP over RSVP configuration. For LDP to install a label received from a non-directly connected peer whose route is through an RSVP tunnel, LDP must receive the corresponding route from the RTM indicating that the RSVP tunnel is used to reach the next-hop.

LDP over RSVP is supported under the following conditions:

- The RTM provides MPLS with a shortcut route for a particular prefix
- The shortcut route must be an IS-IS, OSPF, or BGP shortcut
- The RSVP tunnel must be enabled for LDP tunneling. For more information on enabling LDP tunneling, refer to Enabling LDP over RSVP on page 251.

    NOTE
    When an RSVP tunnel is created on ingress LSR with IS-IS or OSPF shortcuts enabled, and LDP tunneling is also enabled, then the LDP tunnel to the egress router of the RSVP tunnel is not formed. An LDP tunnel is not created at the ingress LSR when RTM selects the RSVP tunnel as the next-hop to the destination.

When a targeted session is used for LDP over RSVP, prefix FECs are advertised to its targeted peer, and prefix FEC received from a targeted peer is installed.

A targeted LDP session is brought up when any one of the following configurations exist:

- A targeted peer address is set up on the egress router of an RSVP tunnel. For more information on configuring a targeted peer address, refer to Configuring a targeted peer address on page 253.
- The user enables RSVP LSP with LDP tunneling configured. For more information on configuring LDP tunneling, refer to Enabling LDP over RSVP on page 251.
- A Layer 2 VPN (VLL or VPLS) peer is configured.

Enabling LDP over RSVP

To enable LDP traffic to tunnel across an RSVP tunnel, first create an LSP, then enable LDP tunneling on the LSP as shown in the following example.

```
device(config)# router mpls
device(config-mpls)# lsp blue
device(config-mpls-lsp-blue)# to 10.20.20.20
device(config-mpls-lsp-blue)# ldp-tunneling
```

The following message appears on the CLI.

```
 This LSP can be used for LDP tunneling if it is used as a shortcut.
```

This message implies that when an RSVP tunnel is used as an IS-IS or OSPF shortcut, then the shortcut must be explicitly configured.

    NOTE
    There is no configuration needed for BGP
    shortcut.

To enable IS-IS shortcuts or OSPF shortcuts, enter the **shortcuts isis** command, or the **shortcuts ospf** command as shown in the following example.

```
device(config-mpls-lsp-blue)# shortcuts isis level2
device(config-mpls-lsp-blue)# enable
Connecting signaled LSP blue
```

Syntax: [no] ldp-tunneling

Syntax: [no] shortcuts isis level1 | level 2

Syntax: [no] shortcuts ospf

By default, LDP tunneling is disabled. The user must disable the LSP configuration to change the setting on the **ldp-tunneling** command.

> **NOTE**
> The **ldp-tunneling** command is not available under bypass LSP configuration.

To disable IS-IS shortcuts or OSPF shortcuts, enter the [**no**] form of the command.

The **level1** or **level2** keyword is required and indicates the level of IS-IS routing enabled on the device. The levels are:

- level1 - A level1 router routes traffic only within the area that includes the router. To forward traffic to another area, a level1 router sends the traffic to the nearest level2 router.
- level2 - A level2 router routes traffic between areas within a domain.

The LDP tunneling configuration is displayed in the output of the **show mpls lsp** command. When LDP tunneling is enabled, the line reads "yes." When it is not enabled, the line reads "no."

```
device(config-mpls)# show mpls lsp blue
LSP blue, to 10.20.20.20
  From: 10.10.10.10, admin: UP, status: DOWN (Path not sent)
  Times primary LSP goes up since enabled: 0
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 1
  Pri. path: NONE, up: no, active: no
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  LDP tunneling enabled: yes
```

Syntax: **show mpls lsp** *lsp_name*

The *lsp_name* variable specifies the LSP name the user wants to display.

The LDP tunneling configuration is also displayed in the output of the **show mpls config** command, and the **show mpls config lsp** command. In the following example, LDP tunneling with IS-IS shortcuts is enabled.

```
device(config-mpls)# show mpls config lsp blue
 lsp blue
  to 10.20.20.20
  shortcuts isis level2
  ldp-tunneling
  enable
```

Syntax: **show mpls config lsp** *lsp_name*

The *lsp_name* variable specifies the LSP name the user wants to display.

The output from the **show mpls config** command displays a list of configured peer addresses as shown in the following example.

```
device# show mpls config
router mpls
 policy
  traffic-eng isis level-2
  ingress-tunnel-accounting
 ldp
  label-withdrawal-delay 30
  session 10.7.7.2 key 2 $LSFVPW9iIQ==
  session 10.7.7.3 key 2 $LSFVPW9iIQ==
 bfd
  min-tx 50 min-rx 50 multiplier 3
```

```
mpls-interface e3/3
  ldp-enable
  admin-group 2
  mpls-interface e3/17
  rsvp-authentication 2 key $LSFVPW9iIQ==
  ldp-enable
```

Syntax: show mpls config

Configuring a targeted peer address

An Extreme device does not send a targeted Hello message in response to receiving a targeted Hello message from a peer that is not configured as a L2VPN peer. In the case when a L2VPN peer is not configured on an Extreme device, and the user would like to enable support for LDP over RSVP, the user must specify the IP address of the peer to bring up a targeted session. To trigger a targeted session that is set up on the egress router of an RSVP tunnel, enter the **targeted-peer** command under the MPLS LDP configuration.

```
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# targeted-peer 10.10.10.10
```

Syntax: [no] targeted-peer *ip-address*

The *ip-address* variable specifies the IP address of the targeted peer. To disable the configuration, enter the [**no**] form of the command.

> **NOTE**
> A targeted peer address is not configured on the ingress router of the RSVP tunnel because configuring LDP tunneling for the LSP automatically brings up a targeted session.

Displaying targeted peer addresses

To display a list of configured peer addresses, enter the **show mpls ldp targeted-peer** command on the CLI as shown in the following example.

```
device# show mpls ldp targeted-peer
Peer_address
10.2.2.2
```

Syntax: show mpls ldp targeted-peer

TTL propagation for LDP over RSVP packets

TTL propagation for LDP over RSVP packets is controlled by the **propagate-ttl** command, and the **label-propagate-ttl** command:
- When the label operation involves the swap of the LDP label followed by the push of the RSVP label, the **label-propagate-ttl** command controls the propagation of the LDP label TTL to the RSVP label TTL. By default, the TTL is not propagated. The RSVP label TTL is set to 255. When the **label-propagate-ttl** command is configured by the user, the LDP label TTL is propagated to the RSVP label TTL.
- When the label operation involves the POP (or the removal) of the LDP label followed by the push of the RSVP label, the following two cases are considered:
  - When the LDP label is not the only label in the Layer 2 or Layer 3 VPN stack, the **label-propagate-ttl** command controls the propagation of TTL from the outer LDP label to the VC label and, in turn, from the VC label to the RSVP label. By default, the **label-propagate-ttl** command is turned off. The VC label TTL is not affected when the LDP tunnel label and the RSVP label TTL are set to 255.

- When the LDP label is the only label in the IP over MPLS stack, the **propagate-ttl** command controls the propagation of TTL from the LDP label to the IP header and, in turn, from the IP header to the RSVP label. By default, the **propagate-ttl** command is turned on.

By default, TTL propagation is enabled for IP over MPLS traffic when an RSVP and an LDP tunnel terminate on the same node. For traceroute purposes, when an RSVP tunnel is traced, then TTL propagation must be enabled.

> **NOTE**
> There is **no label-propagate-ttl** command on the CES 2000 Series and CER 2000 Series devices. The propagation of LDP label TTL to RSVP label TTL is controlled by the **propagate-ttl** command. By default, **propagate-ttl** is enabled, therefore TTL is propagated. **Case 1:** SWAP LDP and PUSH RSVP – LDP label TTL is propagated to RSVP label TTL. **Case 2:** POP LDP and PUSH RSVP – Outer LDP label TTL is propagated to VC label and IP header and then from VC label and IP header to RSVP label. When the **no propagate-ttl** command is configured, then TTL is not propagated in the above scenarios, LDP and RSVP label has a TTL value of 255. The LDP and RSVP label TTL is decremented for every hop traversed. Both the VC label and IP header TTL is not affected by LDP and RSVP label TTL. However, at PHP, when **no propagate-ttl** is configured, after the outermost Label is popped, and when there is no MPLS header but an IP header, then the IP header TTL is decremented by one.

> **NOTE**
> For consistent behavior in all cases of TTL propagation for LDP over RSVP packets, Extreme recommends that the user always turn on or turn off both the **label-propagate-ttl** command and the **propagate-ttl** command.

Enabling TTL propagation

By default, MPLS traceroute does not display the LSRs the RSVP tunnel is transiting through, except when the egress router is acting as the egress for both the LDP and the RSVP tunnel. In other words, the RSVP tunnel is treated as a single hop. The **label-propagate-ttl** command and the **propagate-ttl** command must be enabled in order to display details of the RSVP core. By default, the **propagate-ttl** command is enabled. To trace an RSVP path, enable the **label-propagate-ttl** command on all Extreme devices along the RSVP path, as shown in the following example.

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# label-propagate-ttl
```

Syntax: [no] label--propagate-ttl

To disable the configuration, enter the [**no**] form of the command. By default, the **label-propagate-ttl** command is turned off. When MPLS traceroute is configured through an RSVP core, FEC validation for LDP FEC is not preformed at the transit LSR of the RSVP tunnel.

Class of Service (CoS) treatment for LDP over RSVP

The following sections describe CoS treatment for LDP over RSVP (transit) for ingress RSVP and RSVP *Penultimate Hop Pop (PHP)*.

Ingress RSVP

The internal priority of the ingress RSVP is mapped from the incoming LDP label EXP bits. When the RSVP tunnel has a CoS configured, it overrides the internal priority of the ingress RSVP. By default, the EXP bits in the outgoing RSVP label are mapped from the internal priority. When the **qos exp encoding off** command is configured on the outgoing interface, the RSVP label EXP bits are set to the internal priority of the ingress RSVP. The incoming LDP label EXP bits are preserved in the outgoing LDP label EXP bits irrespective of whether the **qos exp encoding** command is turned on or off on the outgoing interface.

RSVP PHP

The internal priority is mapped from the incoming RSVP label EXP bits. On the egress router of the RSVP tunnel, the outgoing LDP label EXP bits are set to the incoming LDP label EXP bits. This is irrespective of whether the **qos exp encoding** command is turned on or off on the outgoing interface.

Setting the backup retry interval

By default, RSVP tries to bring up an FRR LSPs backup or detour session every 30 seconds. When the number of FRR sessions are very large (48000 is the maximum number of FRR sessions supported), RSVP becomes too busy bringing up the backup every 30 seconds for all the LSPs.

Use the **backup-retry-time** command under the **router-mpls policy** to change the backup retry interval.

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# backup-retry-time 100
```

Syntax: [no] backup-retry-time *interval*

The *interval* parameter valid range is: [10 - 600] seconds.

Use the [**no**] form of this command to revert to the default of 30 seconds.

RSVP-TE Hello

TABLE 14 RSVP-TE Hello glossary

Term	Meaning
LSP	Label Switched Path
MPLS	Multiprotocol Label Switching
RESV	Reserve
RSVP	Resource Reservation Protocol
TE	Traffic Engineering

The RSVP-TE Hello feature is an optional extension to RSVP-TE protocols to detect neighbor down scenarios. It makes use of Hello messages as Keepalive poll mechanism between RSVP peers on a link.

A failure along the path of a signaled RSVP-TE LSP can remain undetected for as long as two minutes or longer (reservation or RESV time-out). During this time, bandwidth is held by the non-functioning LSP on the nodes downstream from the point of failure along the path with the state intact. If this bandwidth is needed by head end tunnels to signal or re-signal LSPs, tunnels may fail to come up for several minutes thereby negatively affecting convergence time.

Hello messages enable RSVP nodes to detect when a neighboring node is not reachable. When RSVP-TE Hello protocol notices that a neighbor is not responding, it treats it as a neighbor down case (link layer communication failure) and either deletes the LSP state or reroutes it based on the type of LSP. This action frees the node's resources to be reused by other LSPs.

A Hello message is sent out periodically to each RSVP peer on a link. If no response is received from the peer within a specified period of time, then the peer is announced "dead" (down). RSVP LSPs going over that peer must either be torn down or re-routed based on the nature of the LSPs.

This Hello mechanism is intended for use between immediate neighbors. Hello processing between two neighbors supports independent selection of configurations of failure detections intervals.

The configuration of Hello message is completely optional. All the messages may be ignored by nodes which do not wish to participate in Hello message processing. This feature complies with *RFC 3209*, section 5 (Hello Extension) other than the default hello-interval time which is different in Extreme implementation.

By default, this feature is disabled.

Vital Fractions for RSVP-TE Hello

The Hello extension is composed of three parts:

- Hello Message
- Hello REQUEST object
- Hello ACK object

Each neighbor can individually issue Hello REQUEST objects. Each request may be answered by an Hello ACK object. The Hello extension is designed so that one side can use the mechanism while the other side does not. All messages may be ignored by nodes which do not wish to participate in Hello message processing. If a particular peer never responds to Hello messages, Extreme routers do not assume that the peer is dead, but simply assume that it does not support Hello messages.

The Hello message has a Msg Type of 20 with a message format as follows:

```
Hello Message : := Common Header [ INTEGRITY ]
Hello
```

Working of RSVP-TE Hello feature

Considering both sides of a link support and wish to participate in Hello message processing, the following is the processing of the feature.

1. A node periodically generates a Hello message containing a HELLO REQUEST object for each neighbor whose status is being tracked. The periodicity is governed by the hello-interval. There is support for each interface configuration of RSVP-TE HELLO to be flexible. This value may be configured on a per interface basis. The default value is nine seconds and the configurable range of hello-interval is 1 to 60 seconds.

2. When generating a message containing a HELLO REQUEST object, the sender fills in the 'Src_Instance' field with a value representing its per neighbor instance. This value does not change while the agent is exchanging Hellos with the corresponding neighbor. The sender also fills in the 'Dst_Instance' field with the *Src_Instance* value most recently received from the neighbor. For reference, refer to this variable as the *Neighbor_Src_Instance*. If no value has ever been received from the neighbor or this node considers communication to the neighbor to have been lost, the *Neighbor_Src_Instance* is set to zero (0). The generation of a message must be suppressed when a HELLO REQUEST object is received from the destination node within the prior hello-interval interval.

3. On receipt of a message containing a HELLO REQUEST object, the receiver generates a Hello message containing a HELLO ACK object. The receiver also verifies that the neighbor has not reset. This is done by comparing the sender's 'Src_Instance' field value with the previously received value. If the *Neighbor_Src_Instance* value is zero, and the 'Src_Instance' field is non-zero, the *Neighbor_Src_Instance* is updated with the new value. If the value differs, then the node treats the neighbor as if communication has been lost.

4. The receiver of a HELLO REQUEST object also verifies that the neighbor is reflecting back the receiver's Instance value. This is done by comparing the received 'Dst_Instance' field with the 'Src_Instance' field value most recently transmitted to that neighbor. If the neighbor continues to advertise a wrong non-zero value after a configured number of intervals (hello-tolerance), then the node must treat the neighbor as if communication has been lost.

5. On receipt of a message containing a HELLO ACK object, the receiver must verify that the neighbor has not reset. This is done by comparing the sender's 'Src_Instance' field value with the previously received value. If the *Neighbor_Src_Instance* value is zero, and the 'Src_Instance' field is non-zero, the *Neighbor_Src_Instance* is updated with the new value. If the value differs or the 'Src_Instance' field is zero, then the node must treat the neighbor as if communication has been lost.

6. The receiver of a HELLO ACK object must also verify that the neighbor is reflecting back the receiver's Instance value. If the neighbor advertises a wrong value in the 'Dst_Instance' field, then a node must treat the neighbor as if communication has been lost.

7. If no Instance values are received, through either REQUEST or ACK objects, from a neighbor within a configured number of hello-intervals (hello-tolerance), then a node must presume that it cannot communicate with the neighbor. The default for this number is three (3). So, the time-out is equal to three times the retransmission period. Range for hello-tolerance is 1 to 255.

8. When communication is lost or presumed to be lost, a node may re-initiate HELLOs. If a node does re-initiate, it must use a *Src_Instance* value different than the one advertised in the previous HELLO message. This new value must continue to be advertised to the corresponding neighbor until a reset or reboot occurs, or until another communication failure is detected. If a new instance value has not been received from the neighbor, then the node must advertise zero in the *Dst_Instance* value field.

For those sessions going over the interface on which a neighbor down is detected, the following actions are taken on the basis of the nature of the LSP:

- For RSVP sessions with no backup available, these sessions are brought down.
- For RSVP sessions with available backups, FRR switchover is performed.

The HELLO mechanism is intended for use between immediate neighbors. So, when the HELLO messages are being exchanged between immediate neighbors, the IP TTL field of all outgoing HELLO messages is set to one.

Risk assessment

Configuring hello-interval on both ends of a link

The **hello-interval** command at an mpls-interface level is used to configure the interval time for sending RSVP-TE Hello Request messages. Configuring the Hello-interval allows the interface to initiate Hello Request messages. When both ends of the link are configured to respond to RSVP-TE Hello messages, the neighbor on receiving the Request message generates a ACK message.

Configuring hello-interval only on one end of a link

The **hello-interval** command at an mpls-interface level is used to configure the interval time for sending RSVP-TE Hello Request messages. If the neighbor does not wish to participate in RSVP-TE Hello message communication, it can ignore the Hello Request messages. The neighbor may send out the ACKs only if it chooses to participate in the RSVP-TE Hello messages. If a particular peer never responds to Hello messages, do not assume that the peer is dead, but simply assume that it does not support Hello messages.

Removing Hello support from one end of the link

Consider the case when both ends of the link supported RSVP-TE Hello messages and the exchange of messages was normal as both links were up. Remove the support for Hello from one side of the link. The other side keeps sending Hello Request messages, but the neighbor starts ignoring these requests as it no longer wishes to participate in Hello messages exchange. In this case, because the neighbor stops sending ACKs, the router considers this as a neighbor down case and brings down all the RSVP sessions going over that interface. After a neighbor down event, Hello message exchange starts off from scratch (re-initiates). If the neighbor does not respond to Hello Requests, the router assumes that the neighbor does not support Hello because no ACK was ever received after re-initiating Hello.

In addition, when RSVP Hello is supported only on one end of the link, the end that supports Hello will send Hello Request messages until it hits the *hello_tolerance* limit and then stops sending any further Hellos messages. It restarts sending Hellos when it receives a Hello message from the neighbor and then again continues the two way communication as before.

> ⚠ **CAUTION**
> Caution: When disabling RSVP hello, please disable it on both sides of the link at the same time to avoid bringing down all the RSVP sessions going over that link.

Configuring Hello-tolerance

Hello-tolerance can be individually configured on both ends of the interface. Considering both sides of the link are participating in Hello communication, if no Instance values are received, through either of the REQUEST or ACK objects, from a neighbor within this configured hello-tolerance number of hello-intervals, then this node presumes that it cannot communicate with the neighbor.

Configuring hello-acknowledgments

Configuring **hello-acknowledgments** command (on the global MPLS RSVP Hello level) enables the router to respond back by sending Hello ACKs on neighbors not carrying any RSVP sessions. By default, Hello ACKs are sent only to neighbors carrying RSVP sessions. This is an optimization.

Configuration steps

There are three parameters that can be configured for this feature. Two of them are at an mpls-interface level and one is at the mpls-rsvp level.

RSVP Hello configuration at global MPLS RSVP level

Interval and tolerance for RSVP-TE Hello protocol can be configured at global MPLS RSVP level. The global configuration is pushed to all the mpls-interfaces if interface level configurations are not present. In addition to these two parameters, one more parameter can be configured at global MPLS RSVP level (acknowledgments).

Backward compatibility

This is fully backward compatible because the feature is turned off by default.

Displaying LDP information

The user can display the following information about LDP:

- The LDP version number and the LSPs LDP identifier and loopback number
- Information about active LDP-created LSPs on the device
- Information about LDP-created tunnel LSPs for which this device is the ingress LER
- LDP database content
- Information about the LDP session between this LSR and its LDP peers
- Information about the connection between this LSR and its LDP peers
- Information about LDP-enabled Interfaces on the LSR

Displaying the LDP version

To display the LDP version number, the LSR ID and loopback number, and the LDP hello interval and hold time, enter the **show mpls ldp database** command shown in the example below.

```
device(config)# show mpls ldp database
Label Distribution Protocol version 1
  LSR ID: 10.210.210.21, using Loopback 1 (deleting it stops LDP)
  Hello interval: Link 5 sec, Targeted 15 sec
  Hold time value sent in Hellos: Link 15 sec, Targeted 45 sec
  Keepalive interval: 6 sec, Hold time multiple: 6 intervals
  Load sharing: 8
  Tunnel metric: 0
  FEC used for auto discovered peers: current 129, configured 129
  Graceful restart: enabled
    Reconnect time: 120 seconds, Max peer reconnnect time: 120 seconds
    Recovery time: 120 seconds, Max peer recovery time: 120 seconds
    Forwarding state holding timer: not running
```

For additional information on the **show mpls ldp** command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying LDP tunnel LSP information

The **show mpls ldp tunnel** command displays information about LDP-created LSPs for which this device is the ingress LER.

For addition information, go to the CLI command in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying the contents of the LDP database

The user can display the contents of the LSRs LDP Label Information Base. This database contains all the labels it has learned from each of its LSR peers, as well as all of the labels it has sent to its LDP peers.

For additional information, refer to the **show mpls ldp database** CLI in the MPLS commands chapter.

Displaying LDP neighbor connection information

To display information about the connection between this LSR and its LDP-enabled neighbors, use the **show mpls ldp neighbor** command. For additional information, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying information about specified LDP-enabled interface

To display information about a specific LDP enabled interface on the LSR, use the **show mpls ldp interface ethernet** command. For additional information regarding the command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying the LDP peer information

To display LDP peering information, use the **show mpls ldp peer** command. For additional information, go the **show mpls ldp peer** CLI command page in *NetIron Command Line Interface (CLI) Reference Guide*.

Display considerations for LDP FEC information

The **show mpls ldp fec** command has changed to allows the user to display all Layer 3 FEC information on the CLI, or specify the FEC type the user wants to display. When displaying the **show mpls ldp fec** command, consider the following:

- The prefix option is introduced to the **show mpls ldp fec** command. The **show mpls ldp fec prefix** command displays the total number of Layer 3 FECs. The total number of Layer 3 FECs is displayed in the Total number of prefix FECs field. For more information on this command, refer to *NetIron Command Line Interface (CLI) Reference Guide*.

- All options that are available under the **show mpls ldp fec** command have moved to the **show mpls ldp fec prefix** command.

- The **summary** option is introduced to the **show mpls ldp fec** command. The **show mpls ldp fec summary** command displays summarized FEC information. For more information on this command, refer to *NetIron Command Line Interface (CLI) Reference Guide*.

- The **vc-fec** option is renamed to **vc** option. All options that are under the **show mpls ldp vc-fec** command have moved under the **show mpls ldp fec vc** command. For more information on this command, refer to *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying information for a specified LDP FEC type

The **show mpls ldp fec prefix** *IPaddress_with_NetMask* command has changed. To display L3 FEC information for a specific FEC type, enter the following command.

```
device# show mpls ldp fec prefix 10.125.125.1/32
FEC_CB: 0x29391f8c, idx: 1, type: 2, pend_notif: None
 State: current, Ingr: Yes, Egr: No, UM Dist. done: No
 Prefix: 10.125.125.1/32, next_hop: 10.90.90.20, out_if: e2/2
 Downstream mappings:
 Local LDP ID       Peer LDP ID      Label  State      CB
 10.128.128.28:0    10.125.125.1:0   3      Installed  0x29391cb0(-1)
```

The table below lists the output displayed for **show mpls ldp fec prefix** command.

TABLE 15 Output from the show mpls ldp fec **prefix** command

Field output	Description
FEC_CB	Memory address of the FEC CB.
idx	A monotonically increasing number assigned to each FEC in the LDP internal FEC tree.
type	FEC type - Prefix FEC is type 2 and Host Address is assigned type 3.
pend_notif	Any notification pending on this FEC.
State	State of the FEC which indicates the FEC advertised to any LDP session (state equal to "current"). When it has no session, it is either called "cur_no_sess" (currently no session) for local FECs or is marked "retained" for non-local.
Ingr	Whether the FEC is an ingress FEC.
Egr	Whether the FEC is an egress FEC.
UM Dist	Specifies when Upstream Mapping Distribution is complete.
Prefix	The IP Prefix associated with the host address or the prefix FEC type.
next_hop	For an ingress FEC, this mentions the next- hop IP address. When LDP selects its outgoing interface as an RSVP tunnel, the next_hop field displays the RSVP tunnel destination address.
out_if	For an ingress FEC, this mentions the output interface to reach to the Next-hop. When applicable, the Out-Intf field displays a VE interface specified by the *vid* variable.
Downstream Mappings	Contents of the downstream mapping CB created as a result of the label mapping received from the downstream LDP peer.
Local LDP ID	Local LDP ID of the LDP session to which this downstream mapping CB belongs.
Peer LDP ID	Remote LDP ID of the LDP session to which this downstream mapping CB belongs.
Label	MPLS label received from the downstream LSR.
State	State of label. Either installed or retained.
CB	Memory address of the downstream mapping CB.

Displaying the LDP FEC VC information

The **show mpls ldp vc-fec** command is renamed to **show mpls ldp fec vc** command. The output from the **show mpls ldp fec vc** command is enhanced to show the total number of VC FECs. The total number of VC FECs is displayed in the total number of VC FECs field.

Displaying information for a specified LDP FEC VC

The output from the **show mpls ldp fec vc** *vc-id* command has changed in the following:

- When a VLL or VPLS peer is up, only one FEC_CB is displayed in the output of the **show mpls ldp fec vc** *vc-id* command. Previously, two FEC_CBs were displayed in the output.

- The MTU enforcement field is introduced in the **show mpls ldp fec vc** *vc-id* command output. The MTU enforcement field indicates whether a MTU enforcement has been enabled. The MTU enforcement field, together with the Local MTU field and Remote MTU field indicates whether a MTU mismatch has occurred.

- When the local and remote VC types for a specified VC ID do not match, two FEC_CBs are displayed.

The examples below describe these changes to the **show mpls ldp fec vc** *vc-id* command in more detail.

For additional information, go to the **show mpls ldp fec** command page in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying the LDP packet statistics

The **show mpls ldp statistics** command displays packet statistics for packet types and packet errors. For additional information, see *NetIron Command Line Interface (CLI) Reference Guide*.

Clearing the LDP packet statistics

The user can clear the LDP packet statistics, as shown in the following command.

```
device# clear mpls ldp statistics
```

Syntax: **clear mpls ldp statistics**

Sample LDP configurations

Figure 52 illustrates a sample configuration with three LDP-enabled LSRs.

FIGURE 52 Sample LDP configuration
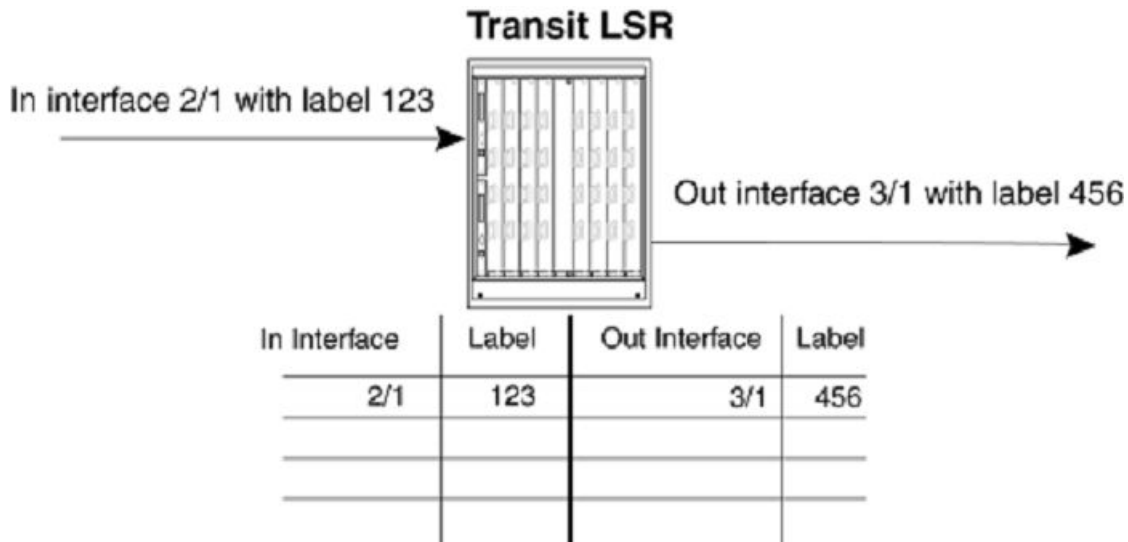
Router device1

The following commands configure Router device1 in

```
device1(config)# interface loopback 1
device1(config-lbif-1)# ip address 10.1.1.1/32
device1(config-lbif-1)# exit
device1(config)# router mpls
device1(config-mpls)# mpls-interface e 2/10
device1(config-mpls)# ldp-enable
device1(config-mpls)# mpls-interface e 2/20
device1(config-mpls)# ldp-enable
device1(config-mpls)# exit
device1(config)# ip route 10.2.2.2/32 10.1.1.2
device1(config)# ip route 10.3.3.3/32 10.1.1.2
device1(config)# route-only
device1(config)# interface ethernet 2/10
device1(config-if-2/10)# enable
device1(config-if-2/10)# ip address 10.1.1.1/24
device1(config-if-2/10)# exit
device1(config)# interface ethernet 2/20
device1(config-if-2/20)# enable
device1(config-if-2/20)# ip address 10.1.1.1/24
```

Router device2

The following commands configure Router device2 in

```
device2(config)# interface loopback 1
device2(config-lbif-1)# ip address 10.2.2.2/32
device2(config-lbif-1)# exit
device2(config)# router mpls
device2(config-mpls)# mpls-interface e 2/10
device2(config-mpls)# ldp-enable
device2(config-mpls)# exit
device2(config)# ip route 10.1.1.1/32 10.1.1.1
device2(config)# ip route 10.3.3.3/32 10.1.1.1
device2(config)# route-only
device2(config)# interface ethernet 2/20
device2(config-if-2/20)# enable
device2(config-if-2/20)# ip address 10.1.1.2/24
device2(config-if-2/20)# exit
```

Router device3

The following commands configure Router device3 in

```
device3(config)# interface loopback 1
device3(config-lbif-1)# ip address 10.3.3.3/32
device3(config-lbif-1)# exit
device3(config)# router mpls
device3(config-mpls)# mpls-interface e 2/10
device3(config-mpls)# ldp-enable
device3(config-mpls)# exit
device3(config)# ip route 10.1.1.1/32 10.1.1.1
device3(config)# ip route 10.2.2.2/32 10.1.1.1
device3(config)# route-only
device3(config)# interface ethernet 2/20
device3(config-if-2/20)# enable
device3(config-if-2/20)# 10.1.1.2/24
device3(config-if-2/20)# exit
```

Sample LDP configuration with VLL

Figure 53 illustrates a sample Virtual Leased Line (VLL) configuration that uses LDP tunnel LSPs.

FIGURE 53 MPLS VLL configuration with LDP tunnel LSPs
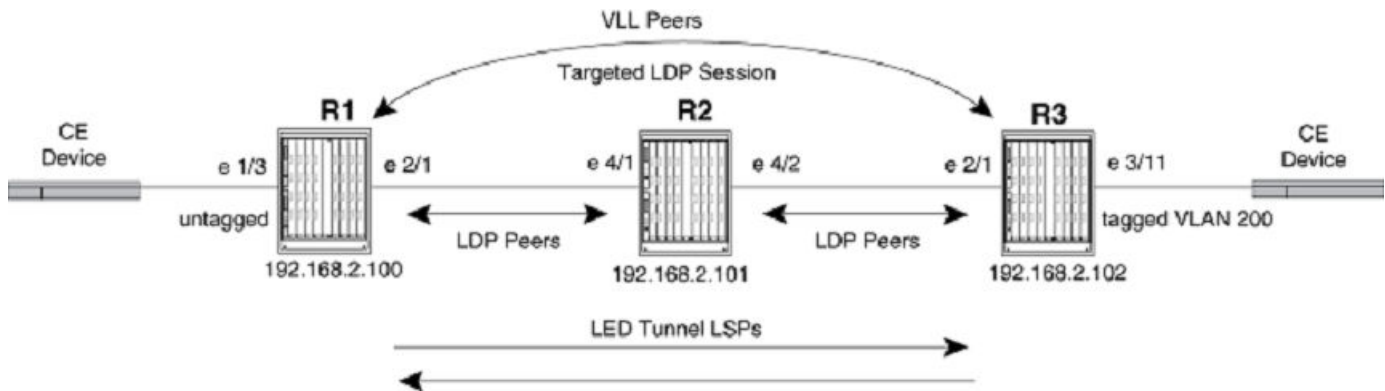
In this example, routers R1 and R3 are *Provider Edge (PE)* routers configured as VLL peers. R1 and R3 have established a targeted LDP session to exchange VLL label information. When this targeted LDP session is established, each router advertises its locally assigned VC label and VC ID to its VLL peer.

In addition, LDP sessions have been established between R1 - R2 and R2 - R3. LDP tunnel LSPs exist in each direction between R1 and R3. When the CE device forwards a Layer 2 packet to R1, the router assigns the packet to an LSP whose destination is R3. R1 encapsulates the packet as an MPLS packet, adding a tunnel label and the VC label advertised to the router by R3. The MPLS packet is then forwarded over the outbound interface indicated by the tunnel label to the next hop in the LSP.

When the MPLS packet reaches R2, the penultimate LSR in the tunnel LSP, R2 pops the tunnel label, leaving the packet with only the VC label, then forwards the packet to R3.

R3 examines the VC label in the packet. On R3, the VC label is mapped to the user-specified endpoint for the VLL. In this example, the endpoint consists of VLAN ID 200 and interface 3/11. R3 then pops the VC label, tags the Layer 2 packet with VLAN 200, then forwards the packet out interface 3/11.

In the opposite direction, R3 assigns traffic received from the CE device to a tunnel LSP destined for R1, pushes tunnel and VC labels onto the packets, and forwards them to the next hop in the LSP. When the packets reach R1, the router pops the VC label and forwards the Layer 2 packets out the interface indicated by the VLL endpoint. In this example, the endpoint consists of interface 1/3, so the packets are forwarded untagged out interface 1/3 to the CE device.

Router device1

The following commands configure Router device1 in Sample LDP configuration with VLL on page 264.

```
device1(config-mpls)# interface loopback 1
device1(config-lbif-1)# port-name Generic All-Purpose Loopback
device1(config-lbif-1)# ip address 192.168.2.100/32
device1(config-lbif-1)# ip ospf area 0
device1(config-lbif-1)# exit
device1(config)# router mpls
device1(config-mpls)# mpls-interface e 2/1
device1(config-mpls)# ldp-enable
device1(config-mpls)# exit
device1(config-mpls)# vll VLL_to_R3 40000
device1(config-mpls-vll)# vll-peer 192.168.2.102
device1(config-mpls-vll)# untagged e 1/3
```

```
device1(config-mpls-vll)# exit
device1(config)# ip router-id 192.168.2.100
device1(config)# router ospf
device1(config-ospf-router)# area 0
device1(config-ospf-router)# exit
device1(config-mpls)# interface e 1/3
device1(config-if-e100-1/3)# port-name VLL_endpoint
device1(config-if-e100-1/3)# enable
device1(config-if-e100-1/3)# exit
device1(config-mpls)# interface e 2/1
device1(config-e10000-2/1)# port-name Connection_to_R2
device1(config-e10000-2/1)# enable
device1(config-e10000-2/1)# ip address 192.168.37.1/30
device1(config-e10000-2/1)# ip ospf area 0
device1(config-e10000-2/1)# exit
```

Router device2

The following commands configure Router device2 in Sample LDP configuration with VLL on page 264.

```
device2(config-mpls)# interface loopback 1
device2(config-lbif-1)# port-name Generic All-Purpose Loopback
device2(config-lbif-1)# ip address 192.168.2.101/32
device2(config-lbif-1)# ip ospf area 0
device2(config-lbif-1)# exit
device2(config)# router mpls
device2(config-mpls)# mpls-interface e 4/1 e 4/2
device2(config-mpls)# ldp-enable
device2(config-mpls)# exit
device2(config)# ip router-id 192.168.2.101
device2(config)# router ospf
device2(config-ospf-router)# area 0
device2(config-ospf-router)# exit
device2(config-mpls)# interface e 4/1
device2(config-e10000-4/1)# enable
device2(config-e10000-4/1)# ip address 192.168.40.1/30
device2(config-e10000-4/1)# ip ospf area 0
device2(config-e10000-4/1)# exit
device2(config-mpls)# interface e 4/2
device2(config-e10000-4/2)# enable
device2(config-e10000-4/2)# ip address 192.168.40.9/30
device2(config-e10000-4/2)# ip ospf area 0
device2(config-e10000-4/2)# exit
```

Router device3

The following commands configure Router device3 in Sample LDP configuration with VLL on page 264.

```
device3(config-mpls)# interface loopback 1
device3(config-lbif-1)# port-name Generic All-Purpose Loopback
device3(config-lbif-1)# ip address 192.168.2.102/32
device3(config-lbif-1)# ip ospf area 0
device3(config-lbif-1)# exit
device3(config)# router mpls
device3(config-mpls)# mpls-interface e 2/1
device3(config-mpls)# ldp-enable
device3(config-mpls)# exit
device3(config-mpls)# vll VLL_to_R1 40000
device3(config-mpls-vll)# vll-peer 192.168.2.100
device3(config-mpls-vll)# vlan 200
device3(config-mpls-vll-vlan)# tagged e 3/11
device3(config-mpls-vll-vlan)# exit
device3(config-mpls-vll)# exit
device3(config)# ip router-id 192.168.2.102
device3(config)# router ospf
device3(config-ospf-router)# area 0
device3(config-ospf-router)# exit
```

```
device3(config-mpls)# interface e 3/11
device3(config-if-e100-3/11)# port-name VLL_endpoint
device3(config-if-e100-3/11)# enable
device3(config-if-e100-3/11)# exit
device3(config-mpls)# interface e 2/1
device3(config-e10000-2/1)# port-name Connection_to_R2
device3(config-e10000-2/1)# enable
device3(config-e10000-2/1)# ip address 192.168.41.1/30
device3(config-e10000-2/1)# ip ospf area 0
device3(config-e10000-2/1)# exit
```

MPLS over GRE tunnel

Multi-Protocol Label Switching (MPLS) packets can traverse a non-MPLS network using *Label Distribution Protocol (LDP)* in transit mode over a *Generic Routing Encapsulation (GRE)* tunnel. This method can be convenient for networks that do not require traffic engineering.

> **NOTE**
> LDP ingress and egress functionality over a GRE tunnel is not supported.

> **NOTE**
> RSVP is not supported for GRE tunnels. In the case of LDP over GRE, when the user enters the **mpls-interface tunnel** command, RSVP is not enabled; for other types of interfaces RSVP is enabled when the interface is configured.

> **NOTE**
> All LDP enabled interfaces must have the same IP MTU. Otherwise, when LDP is enabled on lower MTU interfaces, existing Hello adjacencies flap once to negotiate the LDP MAX PDU size with the peers.

> **NOTE**
> Do not forward packets from one type of tunnel to another type of tunnel in XPP. Packets may not be routed properly.

> **NOTE**
> MP switchover event may not be handled properly by MPLS or RSVP module. This may result in inconsistent state for RSVP LSPs sessions. This could be fixed by adding support for RSVP Hello feature.

LDP LSP over GRE tunnel

This feature works when a GRE tunnel connects two LDP LSP transit nodes and all LDP sessions establish with each peer. Figure 54 shows four routers:

- Routers A and D represent the LDP LSP ingress and LDP LSP egress router.
- Routers B and C represent the LDP LSP transit and GRE ingress and egress points.

The LDP LSP ingress router passes a labeled packet to the GRE ingress and LDP LSP transit router (1), which switches the label, encapsulates the packet and adds a GRE header (2). The packet goes through the GRE tunnel (3). Next, the GRE egress and LDP LSP transit router removes the GRE header, pops the label, and then forwards the packet to the LDP LSP egress router (4).

FIGURE 54 LDP LSP over GRE tunnel example
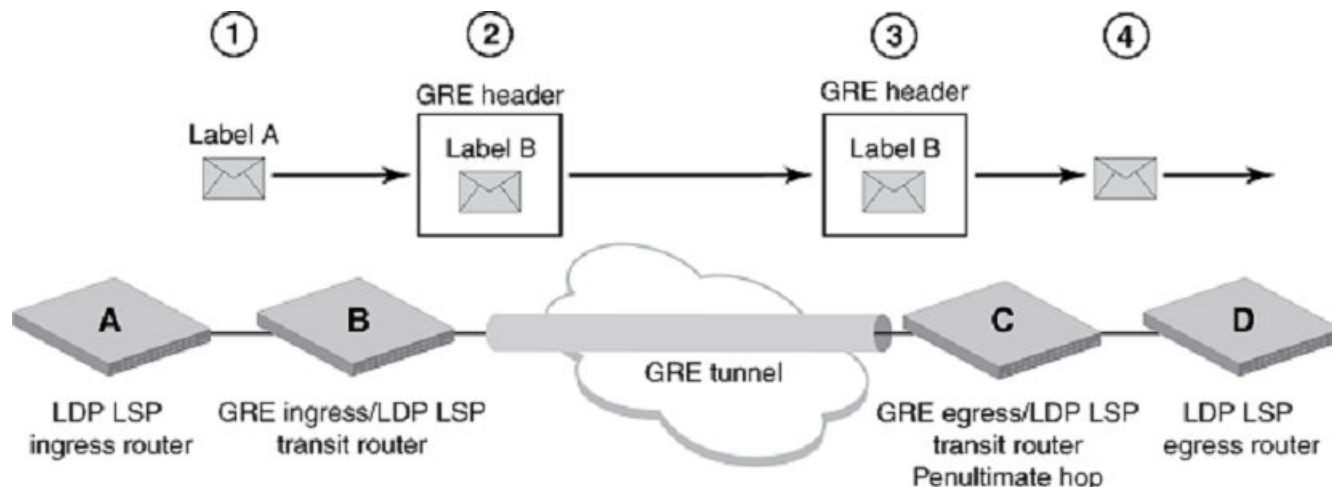
The letters A, B, C, and D in Figure 54 represent the names of the routers in the code configuration example.

Redundant tunnels

The user can configure multiple GRE tunnels between two nodes using a loopback address or an interface address as the source or destination address. In GRE tunnel configuration, the same source and destination combination cannot be used for more than one GRE tunnel.

When multiple GRE tunnels exist between two nodes with LDP enabled on them, multiple LDP hello adjacencies establish between those nodes. Even though multiple hello adjacencies form, each LDP session is based an LSR-ID, so only one session is maintained between those two nodes. This scenario is treated the same as multiple links between two nodes with LDP enabled on them.

When the user creates multiple GRE tunnels to the same destination, each can have its own hello timeout and hello interval. These parameters apply on a per interface basis and apply to the corresponding GRE tunnels. The hello timeout must be at least twice the hello interval. These parameters can be set using the **ldp-params** command found at the MPLS interface configuration level of the command prompt. To modify the hello interval, see Setting the LDP Hello interval values on page 233.

Equal Cost Multipath Routing

Equal Cost Multipath (ECMP) is supported for LDP; and ECMP members can be a combination of GRE tunnels, RSVP tunnels, and regular IP interfaces. When a GRE tunnel is an ECMP member, the packets flowing through the GRE tunnel are encapsulated with a GRE header; the LDP label appears below the level of the GRE header as shown in LDP LSP over GRE tunnel on page 266.

LDP VPLS over a GRE tunnel

Within a *Virtual Private LAN Service (VPLS)* LDP LSPs can carry VPLS traffic over a core network by exchanging VPN labels through LDP targeted sessions with each VPLS peer. Figure 55 shows VPLS traffic in one direction. Four routers are required for this configuration:

- The two outer routers represent the VPLS *Provider's Edge (PE)* ingress and VPLS PE egress router.
- The two inner routers represent the LDP LSP transit and GRE ingress and egress points.

The VPLS PE ingress router passes a packet labeled and marked with a VPN label to the GRE ingress and LDP LSP transit router (1), which switches the label, encapsulates the packet, and then labels the encapsulated packet with a GRE header (2). The packet goes

through the GRE tunnel (3). Next, the GRE egress and LDP LSP transit router removes the GRE header, pops the label, and then forwards the packet with its unaffected VPN label to the VPLS PE egress router (4).

Label popping occurs on the penultimate hop.

FIGURE 55 LDP VPLS over a GRE tunnel example
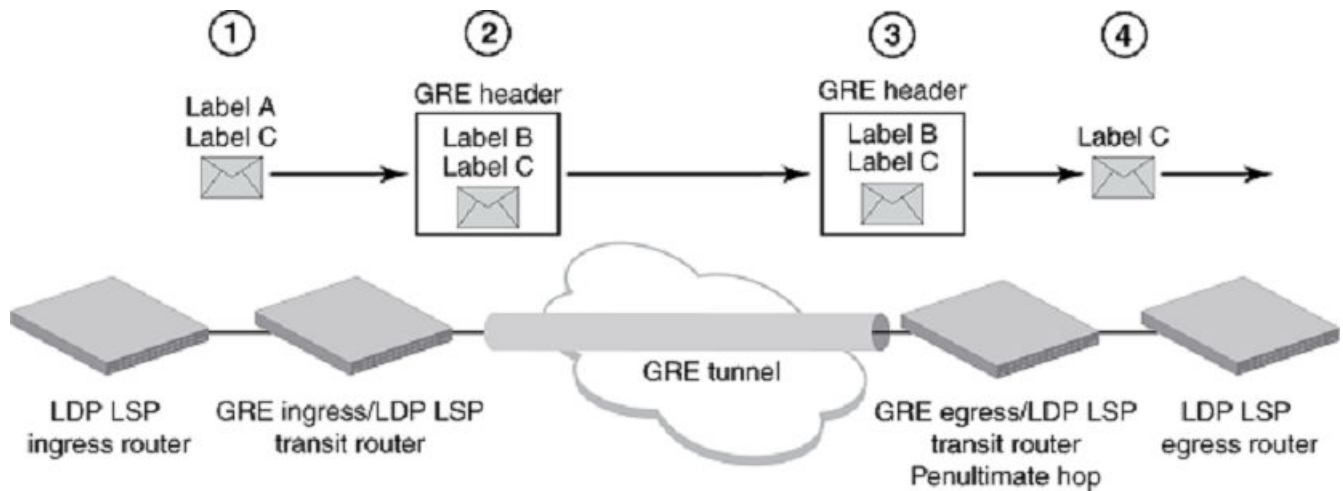
LDP over a GRE tunnel within an encrypted network

Figure 56 shows an implementation of LDP/MPLS over GRE over an encrypted network. Traffic is forced through a non-MPLS network because of the mandatory encrypted network that traffic must cross.

Customer equipment (CE) is connected to PE1 and PE2. PE1 and PE2 negotiate VPLS labels, and an LDP tunnel is created between PE1 and PE2. MPLS traffic is not supported between P1 and P2. LDP transit traffic passes through a GRE tunnel between P1 and P2. Traffic exiting P1 is encrypted and is sent into the non-MPLS cloud. Traffic received at P2 is already decrypted.

In this scenario, LDP transit over GRE tunnel is configured on the P1 and P2 nodes.

FIGURE 56 LDP over GRE with encryption present in network
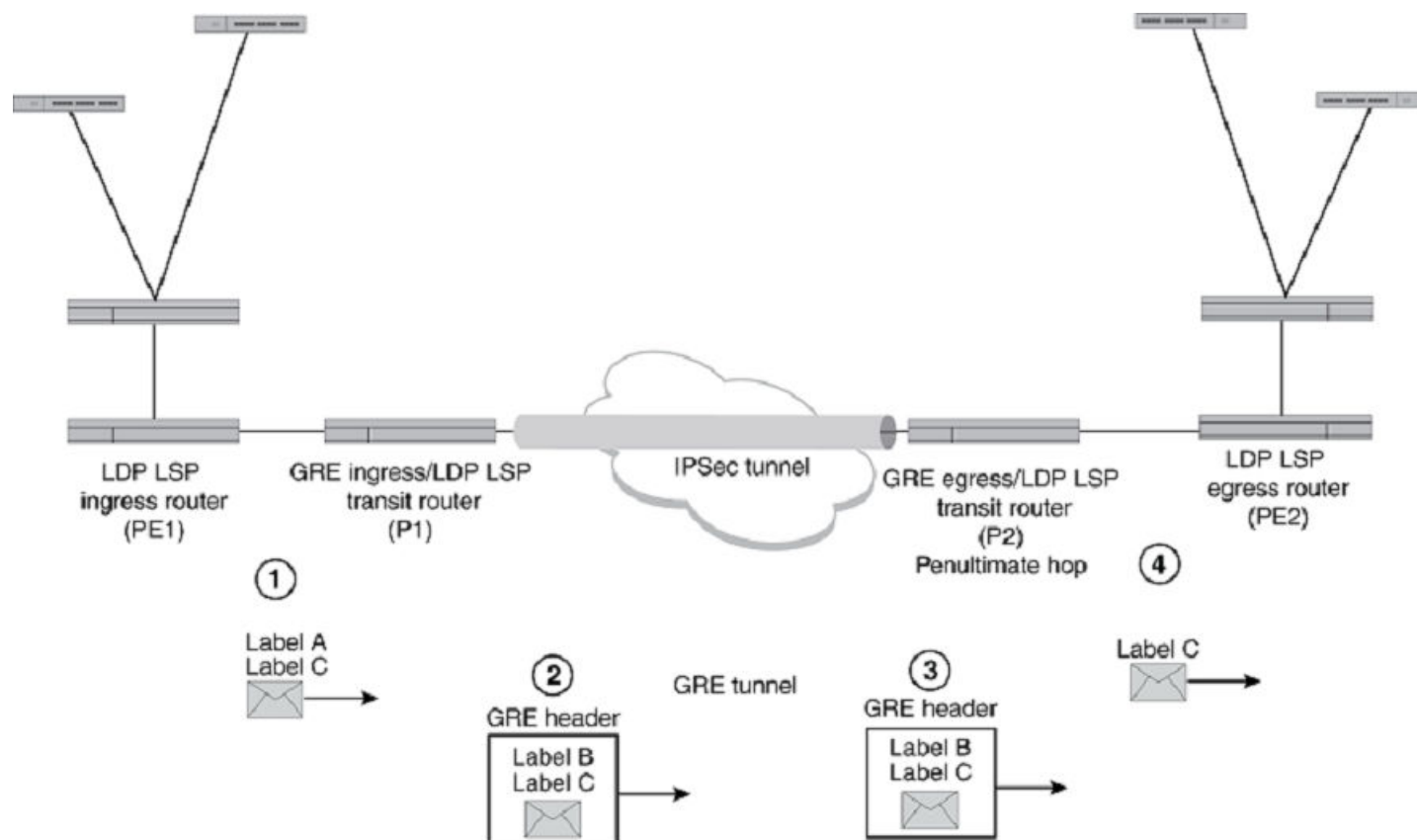
Configuration example

To configure MPLS over a GRE tunnel, first enable an interface as an MPLS interface, then configure the MPLS tunnel and enable LDP.

```
device(config)# router mpls
device(config-mpls)# mpls-interface e1/1
device(config-mpls-if-e1000-1/1)# ldp-enable
device(config-mpls-if-e1000-1/1)# mpls-interface tunnel 200
device(config-mpls-if-e1000-1/1)# ldp-enable
```

Syntax: mpls-interface tunnel *tunnel-id*

The *tunnel-id* variable is a number between 1 and 8192 (the default value is 256). The maximum number of GRE tunnels for the system can be changed by entering the **system-max gre-tunnels** command.

Syntax: system-max gre-tunnels *number*

Router A configuration

Router A is the LDP LSP ingress router. The user needs to configure a routing instance, which in this example is OSPF. Next, the user configures the loopback address, the Ethernet interface, and then the MPLS tunnel with LDP enabled. See LDP LSP over GRE tunnel on page 266.

```
router ospf
    area 0
interface loopback 1
```

```
    enable
    ip ospf area 0
    ip address 10.1.1.1/32
interface ethernet 1/1
    enable
    ip ospf area 0
    ip address 10.11.11.1/24
router mpls
    mpls-interface e1/1
    ldp-enable
```

Router B configuration

Router B is the LDP LSP transit router and the GRE tunnel ingress router. The user needs to configure an OSPF routing instance. Next, the user configures the loopback address, the Ethernet interface, and then the GRE tunnel. Lastly, the user configures MPLS and enable LDP. See LDP LSP over GRE tunnel on page 266.

```
router ospf
    area 0
interface loopback 1
    enable
    ip ospf area 0
    ip address 10.2.2.2/32
interface ethernet 1/1
    enable
    ip ospf area 0
    ip address 10.11.11.2/24
interface ethernet 1/2
    enable
    ip ospf area 0
    ip address 10.22.22.1/24
interface tunnel 200
    tunnel mode gre ip
    tunnel source 10.2.2.2
    tunnel destination 10.3.3.3
    ip ospf area 0
    ip address 10.80.80.1/24
router mpls
    mpls-interface e1/1
    ldp-enable
mpls-interface tunnel 200
    ldp-enable
```

Router C configuration

Router C is the LDP LSP transit router and the GRE tunnel egress router. The user needs to configure an OSPF routing instance. Next, the user configures the loopback address, the Ethernet interface, and then the GRE tunnel. Lastly, the user configures MPLS and enable LDP. See LDP LSP over GRE tunnel on page 266.

```
router ospf
    area 0
interface loopback 1
    enable
    ip ospf area 0
    ip address 10.3.3.3/32
interface ethernet 1/1
    enable
    ip ospf area 0
    ip address 10.22.22.2/24
interface ethernet 1/2
    enable
    ip ospf area 0
    ip address 10.33.33.1/24
interface tunnel 200
    tunnel mode gre ip
    tunnel source 10.3.3.3
```

```
    tunnel destination 10.2.2.2
    ip ospf are 10.80.80.2/24
router mpls
    mpls-interface e1/2
    ldp-enable
mpls-interface tunnel 200
    ldp-enable
```

Router D configuration

Router D is the LDP LSP egress router. The user needs to configure an OSPF routing instance. Next, The user configures the loopback address, the Ethernet interface, and then the MPLS tunnel with LDP enabled. See .

```
router ospf
area 0
interface loopback 1
    enable
    ip ospf area 0
    ip address 10.4.4.4/32
interface ethernet 1/1
    enable
    ip ospf area 0
    ip address 10.33.33.2/24
router mpls
    mpls-interface e1/1
    ldp-enable
```

Deleting a GRE tunnel configuration

A GRE tunnel configured as an MPLS interface cannot be directly deleted. First, the user must delete the GRE tunnel from the mpls-interface configuration, and then the user can delete the GRE tunnel.

Viewing MPLS over GRE information and statistics

The user can view configuration information and various statistics about MPLS over GRE.

The user can view the tunnel interface administrative and operational state by entering the **show mpls interface tunnel** command.

```
device# show mpls interface tunnel 200
gre-tnl200

        Admin: Up Oper: Up
```

Syntax: **show mpls interface tunnel** *tunnel-id*

> NOTE
> Traffic parameters do not apply to
> GRE.

The user can view the LDP tunnel interface configuration information, such as the hello interval and timeout, by entering the **mpls ldp interface tunnel** command. The user can include a tunnel ID to retrieve specific information.

Syntax: **show mpls ldp interface tunnel** [*tunnel-id*]

```
device# show mpls ldp interface tunnel 200
gre-tnl200, label-space ID: 0
 Nbr count: 1
 Hello interval: 5 sec, next hello: 0 sec
 Hello timeout: 15 sec

device#show mpls ldp interface
```

```
Label-space      Nbr    Hello    Next
Interface        ID     Count    Interval    Hello
e1/1             0      1        5           0 sec
gre_tnl200       0      1        5           2 sec
(targeted0       0      0        0           --
```

The user can view the details of an LDP neighbor by entering the **show mpls ldp neighbor** command. To view more detail, enter the **detail** keyword.

Syntax: show mpls ldp neighbor detail

```
device# show mpls ldp neighbor
Nbr Transport    Interface    Nbr LDP ID    Max Hold    Time Left
10.1.1.1         e1/1         10.1.1.1:0    15          12
10.3.3.3         gre-tnl200   10.3.3.3:0    15          12

device# show mpls ldp neighbor detail
Nbr Transport Addr: 10.1.1.1, Interface: e1/1, Nbr LDP ID: 10.1.1.1:0
 MaxHold: 15 sec, Time Left: 10 sec, Up Time: 18 hr 12 min 55 sec

Nbr Transport Addr: 10.3.3.3, Interface: gre-tnl200, Nbr LDP ID: 10.3.3.3:0
 MaxHold: 15 sec, Time Left: 10 sec, Up Time: 18 hr 12 min 55 sec
```

The user can view MPLS LDP path information by entering the **show mpls ldp path** command.

The **show mpls ldp path** command has been enhanced to have a filter for path prefix. The path prefix can be either the IPv4 host address or the IPv4 prefix with subnet mask. Output of the CLI with the path prefix filter displays a single path entry for a specified path IP prefix.

```
device# show mpls ldp path
Destination route    Upstr-session(label)    Downstr-session(label, intf)
10.2.2.2/32          10.1.1.1:0(3)
10.1.1.1/32          10.1.1.1:0(3, e1/1)     10.1.1.1:0(3, e1/2)
10.3.3.3/32          10.3.3.3:0(5050)        10.4.4.4:0(2057,gre-tnl200)

device# show mpls ldp path10.22.22.22
Destination route    Upstr-session(label)    Downstr-session(label, intf)
10.22.22.22/32       10.44.44.44:0(1024)     10.22.22.22:0(3, e2/3  (Trunk11)

device# show mpls ldp path 10.33.33.33/32
Destination route    Upstr-session(label)    Downstr-session(label, intf)
10.33.33.33/32       10.44.44.44:0(1025)     10.44.44.44:0(1024,
ve55)
                                             10.22.22.22:0(1041, e2/3 (Trunk11)
```

Syntax: show mpls ldp path [*path-ip-address* [*ip-mask*]]

The user can view a prefix list by entering the **show mpls ldp fec prefix** command.

```
device# show mpls ldp fec prefix
Total number of prefix FECs: 3
Destination     State     Out-intf     Next-hop       Ingress Egress
10.2.2.2/32     current   --           --             No      Yes
10.1.1.1/32     current   e1/2         10.22.22.22    Yes     No
                          e1/1         10.11.11.11
10.3.3.3/32     current   gre-tnl200   10.33.33.33    No      No
```

Syntax: show mpls ldp fec prefix

The user can view the MPLS RSVP state and settings by entering the **show mpls rsvp interface** command.

> **NOTE**
> MP switchover event may not be handled properly by MPLS/RSVP module. This may result in inconsistent state for RSVP LSPs/Sessions. This could be fixed by adding support for RSVP Hello feature.

```
device# show mpls rsvp interface
Interface State MD5 Auth
```

```
e1/1      Up    OFF
e1/2      Up    OFF
```

Syntax: show mpls rsvp interface

LDP Shortcuts

LDP shortcut

The Label Distribution Protocol (LDP) adds support including shortcut routes for LDP tunnels.

An IP static route is used to add the shortcut route for the LDP tunnel. This enables the static module to install a Label Distribution Protocol (LDP) shortcut route in the Routing Table Manager (RTM), which can now be used by the host generated traffic, such as Boarder Gateway Protocol (BGP) or Operations, Administration and Maintenance (OAM).

LDP shortcut configurations

The following tables use the figure above as a reference. The following table displays conditions and actions when installing an LDP tunnel route.

TABLE 16 Installing an LDP tunnel route

Pre-condition	MPLS LDP is set up, and the LDP tunnel from the left LSR-1 to the right LSR-2 is up. The **next-hop-enable-mpls** is enabled.
	A static route for LER loopback immediately connects to the router as the next hop.
Trigger	Trigger is the configuration of the command **ip route** *LER-2 LER-2* "
Action	The IP route is added to the RTM for the right LER-2 loopback that points to the LDP tunnel next hop.
Post-condition	The IP routing table shows the LDP tunnel next hop as the outgoing port for the newly added route.

The following table displays conditions and actions for a ping from LER-1 to LER-2 loopback.

TABLE 17 Ping from LER-1 to LER-2 loopback

Pre-condition	MPLS LDP is set up, and the LDP tunnel from the left LSR-1 to the right LSR-2 is up.
	The **next-hop-enable-mpls** command is enabled.
	the IP route installed for the LDP tunnel next hop.
	A static route for LER loopback immediately connects to the router as the right LER-2.
Trigger	The user starts a ping from the left LER-1 to right LER-2 loopback.
Action	The IP destination address lookup resolves to the LDP tunnel next hop.
Post-condition	The ICMP packet is sent to the LDP tunnel next hop. The LP packet capture shows the outgoing packets with the respective MPLS tunnel labels.
Alternate Triggers	Any IP-based control packet generated from the left LER-1 that has the destination address of right LER-2 loopback takes the same route.

The following table displays conditions and actions for IP data by way of LER-1 to LER-2 loopback

TABLE 18 IP data traffic by way of LER-1 to LER-2 loopback

Pre-condition	MPLS LDP is set up, and the LDP tunnel from LSR-1 to LSR-2 is up.
	The **next-hop-enable-mpls** command is enabled.
	The IP route is installed for the LDP tunnel next hop.
	A static route for LER loopback immediately connects to the router as next hop.
Trigger	The data packet comes to LER-1 with LER-2 loopback as the destination address.
Action	The IP destination address lookup resolves to the LDP tunnel next hop.
Post-condition	The data packet is sent to the LDP tunnel next hop.

Configuring an LDP shortcut

To configure an LDP shortcut, use the **ip route** command to add an LDP shortcut route using static IP routes.

Use the following steps to configure an LDP shortcut.

- Setup LDP.

```
device>enable
device# configure terminal
device(config)# router mpls
device(config-mpls)# mpls-interface e 4/5
device(config-mpls-if-e10000-4/5)# ldp-enable
```

1. For LER-1, add a static route for LER-2 loopback with the immediately connected router as the next-hop (the route to reach LER-2 loopback using LER-A) or attained by IGP.

2. Enable the **next-hop-enable-mpls** command.

3. Once the LDP tunnel comes up, add static route again for the LER-2 loopback with LER-2 loopback as the next hop.

4. Repeat steps 1 through 3 for LER-1.

Configuration example

The following example shows a configuration for LER-1.

```
device>enable
device# configure terminal
device(config)# ip route next-hop-enable-mpls
device(config)# ip route 2.2.2.2/32 192.168.10.25
device(config)# ip route 2.2.2.2/32 2.2.2.2


device# configure terminal
device(config)# interface loopback 1
device(config-lbif-1)# ip address 1.1.1.1/32


device# configure terminal
device(config)# interface ethernet 4/5
device(config-if-e1000-4/5)# enable
device(config-if-e1000-4/5)# ip address 192.168.10.15/24


device# configure terminal
device(config)# router mpls
device(config-mpls)# mpls-interface e4/5
device(config-mpls-if-e1000-4/5)# ldp-enable
```

Configuring MPLS Virtual Private LAN Services

Overview

This chapter explains how to configure *Virtual Private LAN Services (VPLS)* . VPLS is a method for carrying Layer 2 frames between *Customer Edge (CE)* devices across an MPLS domain. The implementation supports VPLS as described in the IETF *RFC 4762* (Virtual Private LAN Services over MPLS Using LDP Signaling).

How VPLS works

Virtual Private LAN Services (VPLS) enhances the point-to-point connectivity defined in the Draft-Martini IETF documents by specifying a method for *Virtual Circuits (VCs)* to provide point-to-multipoint connectivity across the MPLS domain, allowing traffic to flow between remotely connected sites as if the sites were connected by a Layer 2 switch.

VPLS can be used to transport Ethernet frames to and from multiple, geographically dispersed sites belonging to a customer Virtual Private Network (VPN). The *Provider Edge (PE)* devices connecting the customer sites provide functions similar to a Layer 2 switch. The PE devices learn the MAC addresses of locally connected customer devices, flood broadcast and unknown unicast frames to other PE devices in the VPN, and create associations between remote MAC addresses and the VC *Label Switch Patches (LSPs)* used to reach them.

Figure 57 shows an illustration of a VPLS configuration with two customer VPNs. Two separate VPLS instances have been created, one for Customer A's VPN and one for Customer B's VPN. A VPLS instance consists of a full mesh of VC LSPs between the customers' PE devices. In the example, Customer A's VPLS instance consists of VC LSPs between routers R1, R2, and R3. Customer B's VPLS instance consists of VC LSPs between routers R3 and R4. Because VC LSPs are unidirectional, separate VC LSPs exist in each direction between each of the PE devices. When *Label Distribution Protocol (LDP)* is enabled on the MPLS interfaces on the PE devices, the VC LSPs are established automatically through LDP when the user specifies the VPLS peers on the PE devices.

Alternatively, LSPs can be established using *Resource ReSerVation Protocol- Traffic Engineering (RSVP-TE)* by manually configuring LSPs to all PE devices. The same LSP from one PE to another PE can be shared by multiple VPLS instances for traffic belonging to different customers. In this case, traffic belonging to different customers has the same tunnel label, but different VC labels. When more than one LSP exists from one PE to another PE for multiple VPLS instances, traffic belonging to the different VPLS instances are load-balanced across the LSPs. In this case, traffic belonging to the different VPLS instances has different tunnel and VC labels.

In Figure 57, the VPLS instance for Customer A links its CE devices so that they appear to be a single Layer 2 broadcast domain. The VPLS instance for Customer B has two VLANs configured within the VPLS instance, VLAN 100 and VLAN 200. The VPLS instance for Customer B has two endpoints on PE device R4. Unlike a *Virtual Leased Line (VLL)*, a VPLS instance can have multiple endpoints. The PE device performs local and remote VLAN tag translation, so that multiple VLANs are specified under a single VPLS instance.

FIGURE 57 Sample VPLS configuration
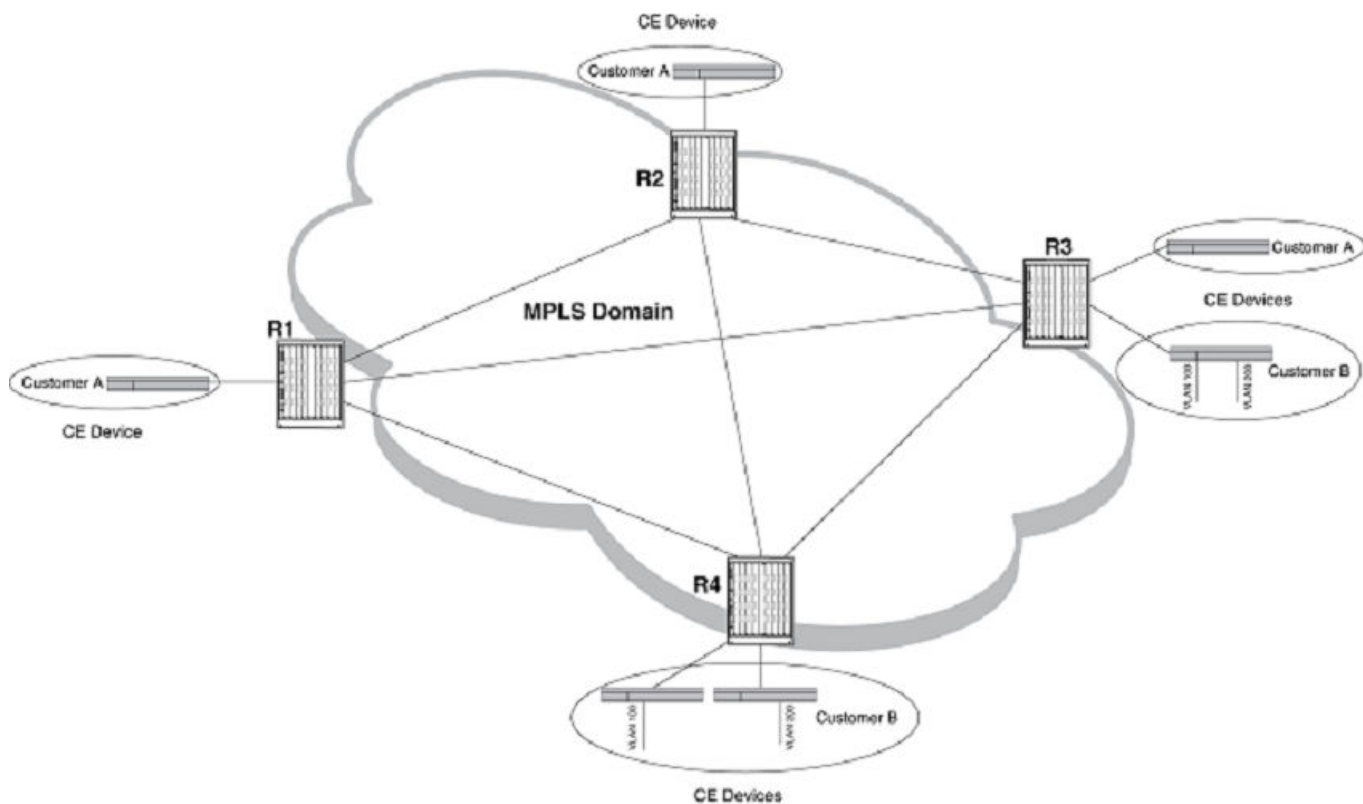
A PE device in the VPLS configuration operates like a standard Layer 2 switch, in that it performs MAC address learning, flooding, and forwarding for the CE devices in each VPLS instance. For example, when PE device R1 receives a Layer 2 frame with a given MAC destination address from Customer A's CE device, it looks up the MAC address in a Layer 2 forwarding table that records associations between MAC addresses and VC LSPs. This forwarding table is known as the *VPLS MAC database*.

When the MAC address is found in the VPLS MAC database, the PE device finds the associated VC LSP, encapsulates the frame as an MPLS packet, and pushes an inner VC label and outer tunnel label onto the packet. The packet is then sent over a tunnel LSP to the VC peer. When the MAC address is not found in the VPLS MAC database, the frame is flooded to all of the PE devices and locally connected CE devices (except for the CE device that originated the frame) in the customer's VPLS instance. When a response is received, an entry for the MAC address and the VC from which it arrived is added to the VPLS MAC database. Subsequent frames targeting the MAC address are not flooded to the other devices in the VPLS instance. In this way, the PE device learns the MAC addresses of the remotely connected customer devices. MAC addresses received at the local VPLS endpoints are also learned in the VPLS MAC database for the VPLS instance.

The PE devices do not run *Spanning Tree Protocol (STP)* over the MPLS domain. The full mesh of PE devices in a VPLS configuration allows one PE device to reach any other PE device in the VPN in exactly one hop, with no transit PE devices in between. The PE devices apply a split horizon rule when forwarding frames within the VPN. When a PE receives a customer frame from a VC LSP, it can forward the frame only to a directly attached customer device, not to another VC LSP. This allows the VPLS instance to have a loop-free topology without having to run STP.

> **NOTE**
> Packets are forwarded in hardware just like packets with other destination addresses.

> **NOTE**
> On CES 2000 Series and CER 2000 Series devices, the **route-only** command must not be configured on untagged MPLS uplinks when using it for VPLS or VLL. Otherwise, the incoming VPLS or VLL traffic is dropped.

> **NOTE**
> The CES 2000 Series and CER 2000 Series devices can only support 127 endpoints plus peers in a VPLS instance.

> **NOTE**
> MPLS control traffic is not supported on OpenFlow Hybrid ports.

Configuring VPLS instances

This section explains how to set up VPLS instances.

Limitations

- In CES 2000 Series and CER 2000 Series devices, a CoS value configured during VPLS instance creation is not used in the back-end. However, in XMR Series and MLX Series devices, the configured CoS value is carried in the MPLS label EXP field in case of ingress PE.
- In CES 2000 Series and CER 2000 Series devices, the customer PCP/DSCP in the outgoing packets through customer endpoint is replaced with the VCB Label EXP bits in the MPLS packet. Therefore, PCP/DSCP bits cannot be preserved end-to-end.
- When more than 1000 VPLS instances, that share the same endpoint, are configured and mapped to a single functional port; the functional port goes down and flaps certain protocols like BFD and UDLD for shorter durations.

Creating a VPLS instance

The user creates a VPLS instance by entering VPLS configuration statements on two or more PE routers. The endpoints of a VPLS instance are associated by having the same VPLS *Virtual Circuit Identifier (VCID)* on each PE router.

To create a VPLS instance, enter commands such as the following:

```
device(config)# router mpls
device(config-mpls)# vpls v1 100
device(config-mpls-vpls-v1)#
```

On the VPLS peers (when they are devices), the user would enter commands such as the following:

```
device(config)# router mpls
device(config-mpls)# vpls v1 100
device(config-mpls-vpls-v1)#
```

Syntax: **vpls** *name vpls-vcid* [**cos** *cos-value*] [**max-mac** *max-mac-entries*]

The **vpls** *name* variable specifies the VPLS instance name.

The *vpls-vcid* variable is the VPLS ID number of the VPLS instance. The *vpls-vcid* variable can take a value in the range of 1 through 4294967294.

The user can optionally specify a *Class of Service (CoS)* setting for the VPLS instance. When a CoS value is set, the device selects a tunnel LSP that also has the CoS value when one is available. When no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VPLS instance). The CoS value has a range from 0 through 7.

Setting a per-VPLS MAC table limit

The user can configure a maximum number of MAC entries that can be learned for a specified VPLS instance. This number cannot be exceeded. This limit can be configured at any time, although operation is more robust when the user configures the limit at the same time that the user configures the VPLS instance.

Configuring the maximum number of MAC entries for a VPLS

To configure a maximum number of MAC entries available to a VPLS instance, enter commands such as the following:

```
device(config)# router mpls
device(config-mpls)# vpls v1 100 max-mac 3000
```

Syntax: **vpls** *name / vpls-vcid* [**cos** *cos-value*] [**max-mac** *max-mac-entries*]

The *name* variable is the name of the VPLS instance for which the user is configuring the maximum number of MAC entries.

The *vpls-vcid* variable is the VPLS ID number of the VPLS instance for which the user is configuring the maximum number of MAC entries. The *vpls-vcid* variable can take a value in the range of one through 4294967294.

The user can optionally specify a *Class of Service (CoS)* setting for the VPLS instance. When a CoS value is set, the device selects a tunnel LSP that also has this CoS value, when one is available. When no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VPLS instance). The CoS value has a range from zero through seven.

The *max-mac-entries* variable specifies the maximum number of MAC entries that can be learned for the VPLS instance. The *max-mac-entries* value can range from one to global VPLS MAC database size.

Specifying the maximum size of the VPLS MAC database

The VPLS MAC database serves as a Layer2 forwarding table that associates local MAC addresses with CE devices and remote MAC addresses with VC LSPs used to reach the remote CE devices. By default, the VPLS MAC database can contain a total of 2048 entries on a MLX Series device and 8192 entries on a XMR Series device. This number represents the total number of MAC addresses that can be learned for all VPLS instances configured on the device.

The user can globally specify a different maximum size for the VPLS MAC database by entering a command such as the following:

```
device(config)# system-max vpls-mac 4096
```

Syntax: **system-max vpls-mac** *number-of-entries*

> NOTE
> The user must reload the system for the **system-max vpls-mac** command to take
> effect.

VPLS LDP MAC Address Withdrawal

For faster VPLS convergence, the user can remove or unlearn the MAC addresses that are learned dynamically. *The Label Distribution Protocol (LDP)* Address Withdrawal message is sent with the list of MAC addresses, which need to be withdrawn to all other PEs that are participating in the corresponding VPLS service.

The MAC address withdrawal feature is added through the LDP Address Withdrawal message. To enable the MAC address withdrawal feature, use the **mac-address withdrawal** command. By default, MAC address withdrawal is disabled.

Configuration considerations:

- This configuration is not applicable for PBB enabled VPLS Instances.
- ON CCEP ports, MAC withdrawal messages are sent only when both local and remote links of the CCEP goes down.
- On a MCT standby switch, the LDP MAC withdrawal messages are sent to the active switch on MCT spoke PW and the MCT active switch relays the LDP messages to all the active PWs.

Configuring MAC address withdrawal globally

Use the **mac-address withdrawal** command at the global configuration mode to enable sending LDP MAC address withdrawal messages to all VPLS instances.

```
device(config-mpls)# vpls-policy
device(config-mpls-vpls-policy)# mac-address withdrawal
```

Syntax: **[no] mac-address withdrawal**

Configuring MAC address withdrawal per VPLS instance

Use the **mac-address withdrawal** command at the interface configuration mode to enable sending LDP MAC address withdrawal messages to specific VPLS instances.

```
device(config-mpls)# vpls sample1
device(config-mpls-vpls-sample1)# mac-address withdrawal
```

Limiting the number of MACs withdrawn

Use the **mac-address withdrawal-limit** command at the global configuration mode to limit the number of MACs withdrawn to all VPLS instances in a five second interval.

```
device(config-mpls)# vpls-policy
device(config-mpls-vpls-policy)# mac-address withdrawal-limit
```

Syntax: **[no] mac-address withdrawal-limit**

Use the **mac-address withdrawal-limit** *num* command at the global configuration mode to specify the number of MACs withdrawn to all VPLS instances in a five second interval.

```
device(config-mpls)# vpls-policy
device(config-mpls-vpls-policy)# mac-address withdrawal-limit 1000
```

Syntax: **[no] mac-address withdrawal-limit** *num*

The *num* parameter can be a value between 100-2000. Default value is 500.

Clearing the contents of the VPLS MAC database

To clear the entries stored in the VPLS MAC database belonging to a VPLS instance, enter a command such as the following.

```
device# clear mac vpls name v1
```

Syntax: **clear mac vpls name** [*name* | **id** *vpls-vcid* | **ethernet** *portnum* | **label** *label*]

The **name** *name* parameter clears all entries associated with the named VPLS instance.

The **id** *vpls-vcid* parameter clears all entries associated with the specified VPLS VCID.

The **ethernet** *portnum* parameter clears all local MAC entries on the specified port.

The **label** *label* parameter clears all entries associated with a local VC label.

Specifying the maximum number of VPLS instances on the device

By default, the maximum number of VPLS instances is 512 on a MLX Series device and 2048 on a XMR Series device. The configured maximum number of VPLS instances has an effect on the size of the label range for each VPLS instance. The label range is the set of labels that the VPLS instance can assign to its peers for use as the peer's local VC label.

The product of the maximum number of VPLS instances and the label range is always equal 65536. This means that when the maximum number of VPLS instances is 2048, then the label range is 32; when the maximum number of VPLS instances is 8192, then the label range is eight; and so on.

To change the maximum number of number of VPLS instances to 8192, enter the following command.

```
device(config)# system-max vpls-num 8192
```

Syntax: **system-max vpls-num** *number-of-VPLS-instances*

> NOTE
> The user must reload the system for this command to take
> effect.

The user can display the configured maximum number of VPLS instances, as well as the size of the label range, with the **show mpls vpls summary** command.

Specifying VPLS peers

As part of the VPLS configuration, the user specifies the IP address of each VPLS peer. VPLS requires a full mesh of tunnel LSPs; each PE router must have tunnel LSP reachability to each of its VPLS peers. Tunnel LSP reachability is defined as having at least one operational RSVP- or LDP-signaled LSP with the destination (the 'to" address of the LSP) matching the VPLS peer's IP address. An LSP terminating on the VPLS peer but configured with a different destination address would not be considered a match.

By default, each PE router attempts to initiate an LDP session through extended discovery with its VPLS peers, when a session is not already established. Each VPLS instance is allocated a range of 32 labels. The PE router assigns one label in the range to each of its peers to be used as the peer's local VC label. When there are more than 32 peers in the VPLS instance, an additional label range is automatically allocated to the VPLS instance. The size of the label range depends on the configured maximum number of VPLS instances on the device. Refer to Specifying the maximum number of VPLS instances on the device on page 280 for more information.

Once the LDP session is established, the PE device advertises the local VC label, along with the VPLS ID, to its VPLS peers in a downstream-unsolicited manner. In a similar way, the PE also learns the remotely assigned VC labels from its VPLS peers.

To specify three remote VPLS peers within a VPLS instance, enter a command such as the following.

```
device(config-mpls-vpls-v1)# vpls-peer 192.168.2.100 192.168.2.101 192.168.2.102
```

Syntax: vpls-peer *ip-addr* [*ip-addr...*]

The IP address of each VPLS peer must match that of a destination for a tunnel LSP configured on the device.

Preserving EXP bits in MPLS header

In the **extended-qos-mode** configuration, the traffic class (TC) value retrieved from packets are often lost, resulting in MPLS uplink packets queued in zero queue. To prevent this, the traffic class value is forced based on the received EXP value in MPLS header.

The match criteria in hardware includes EXP in addition to the existing MPLS Label. Packets undergo regular VPLS processing and are queued based on the traffic class specified in EXP. This results in creating TTI entry for every EXP value, preventing zero queues. This is enabled by the **set-force-tc-match-label-exp** command.

```
device(config)#extended-qos-mode set-force-tc-match-label-exp
```

After configuring all the VPLS instances and required peers on both ingress and egress peers, issue the **set-force-tc-match-label-exp** command under the **extended-qos-mode** command. Reload the device to enable the feature.

For more information on commands, please refer the *Extreme NetIron Command Reference*.

Setting the VPLS VC mode

The Extreme devices support the following VPLS VC modes, which determine whether or not VLAN tags are carried across the MPLS cloud:

- **Raw mode** - This is the default VC mode. When this mode is in effect, the VLAN tag information in the original payload *is not* carried across the MPLS cloud
- **Tagged mode** - When tagged mode is enabled, the VLAN tag information in the original payload is carried across the MPLS cloud

VPLS raw mode

By default, VPLS packets are sent to remote peers over the MPLS cloud in *raw mode* . This means that no VLAN tag information in the payload is carried across the MPLS cloud. In raw mode, the VLAN priority (Class of Service) of the original (incoming) packets is lost once the packets are sent through the cloud.

> **NOTE**
> When desired, the user can enable the device to preserve the VLAN tag information in the payload and carry it across the MPLS cloud to remote peers. For more information, see VPLS tagged mode on page 286.

CoS behavior for VPLS raw mode

> **NOTE**
> This section assumes that the user understands how QoS works.

The system level command **extended-qos-mode** and interface level command **qos pcp encode-policy off** for CES 2000 Series and CER 2000 Series affects the existing QoS behavior for raw mode in VPLS. It is advisable to use only either raw mode or raw-pass-through-mode for all VPLS instance in a system for the CES 2000 Series and the CER 2000 Series.

Table 19 describes the expected Class of Service (CoS) behavior for VPLS packets when VPLS raw mode is in effect.

TABLE 19 Expected class of service behavior for VPLS raw mode

VPLS endpoints	Incoming packet		MPLS cloud		Outgoing packet	
Outer VLAN	Inner VLAN	Tunnel/VClabel (Z)	Payload tag	Outer VLAN	Inner VLAN	
Dual-tagged to dual-tagged	X	Y	V or internal priority	N/A	W or Z	Z
Single-tagged to dual-tagged	X	N/A				Z
Untagged to dual-tagged	N/A	N/A				Z
Dual-tagged to single-tagged	X	Y				N/A

V - Mapped EXP bits from internal priority (**X** contributes to internal priority) using the EXP encode table.

W - Mapped CoS from internal priority (**Z** contributes to internal priority) using the CoS encode table.

X - Original outer VLAN CoS.

Y - Original inner VLAN CoS.

Z - Incoming EXP bits as described by the 'Tunnel/VC Label' column - **V** or internal priority.

- The 'Tunnel/VC Label' column differentiates the behavior when the **qos exp encode** policy is ON (default) or OFF.
- The 'Outgoing Packet Outer VLAN' column differentiates the behavior when the **qos pcp encode** policy is ON (default) or OFF.

VPLS raw pass through mode

By default, VPLS packets are sent to remote peers over the MPLS cloud in raw mode. This means that no VLAN tag information in the payload is carried across the MPLS cloud. In raw mode, the VLAN priority (Class of Service) of the original (incoming) packets is lost once the packets are sent through the cloud.

Although Extreme implementation follows *RFC 4448* in terms of how raw mode and tagged mode operates, Extreme devices occasionally cannot interoperate with certain VPLS VC raw mode third party equipment that has interpreted *RFC 4448* differently.

When a third party device remote peer is connected to an Extreme device, and it was identified under raw mode, the remote peer may expect the presence of the tag in the packet it received from its MPLS uplink. It may also send the payload tag towards its remote peer when sending packets by way of the MPLS uplink towards the Extreme device peer. This causes the two peers to not communicate correctly.

Single tag to single tag packet tag handling into or from the MPLS uplink with raw-pass-through mode with ports configured as untagged.

Using the raw pass through option enables the user to configure the VC mode to interoperate between third party devices. The raw pass through option allows the user to:

1. Select the **raw-pass-through** mode which behaves like a tagged mode when all endpoints are configured as tagged endpoints.

2. Select the **raw** mode which behaves like an untagged mode when all endpoints are configured as untagged endpoints.

3. Select **raw** mode when all endpoints are configured as untagged endpoints even though the peers continue to signal the PW VC –type as raw mode.

Difference in behavior between devices

Table 20 describes the behavioral differences of the Extreme devices.

TABLE 20 Difference in behavior between devices

Behavior	MLX Series and XMR Series	CES 2000 Series and CER 2000 Series
QoS Command Configuration	QoS Commands must be configured at egress interface.	QoS Commands must be configured at Ingress interface.
Carrying PCP end to end in raw-pass-through mode	The **qos pcp encode-policy off** command must be configured at VPLS end point interface.	The user must: 1. Configure the **qos pcp encode-policy off** command at ingress PE for VPLS end point interface. 2. Configure the **qos pcp encode-policy off** command at the egress PE for MPLS interface.
Carrying DSCP end to end in raw-pass-through mode	By Default DSCP is carried end to end.	Configure the **extended-qos-mode** at the egress PE.
Carrying both PCP and DSCP end to end in raw-pass-through mode	The **qos pcp encode-policy off** command must be configured at VPLS end point interface.	The user must: Configure the **qos pcp encode-policy off** command at ingress PE for VPLS end point interface. Configure **extended-qos-mode** at egress PE.

FIGURE 58 Sample configuration
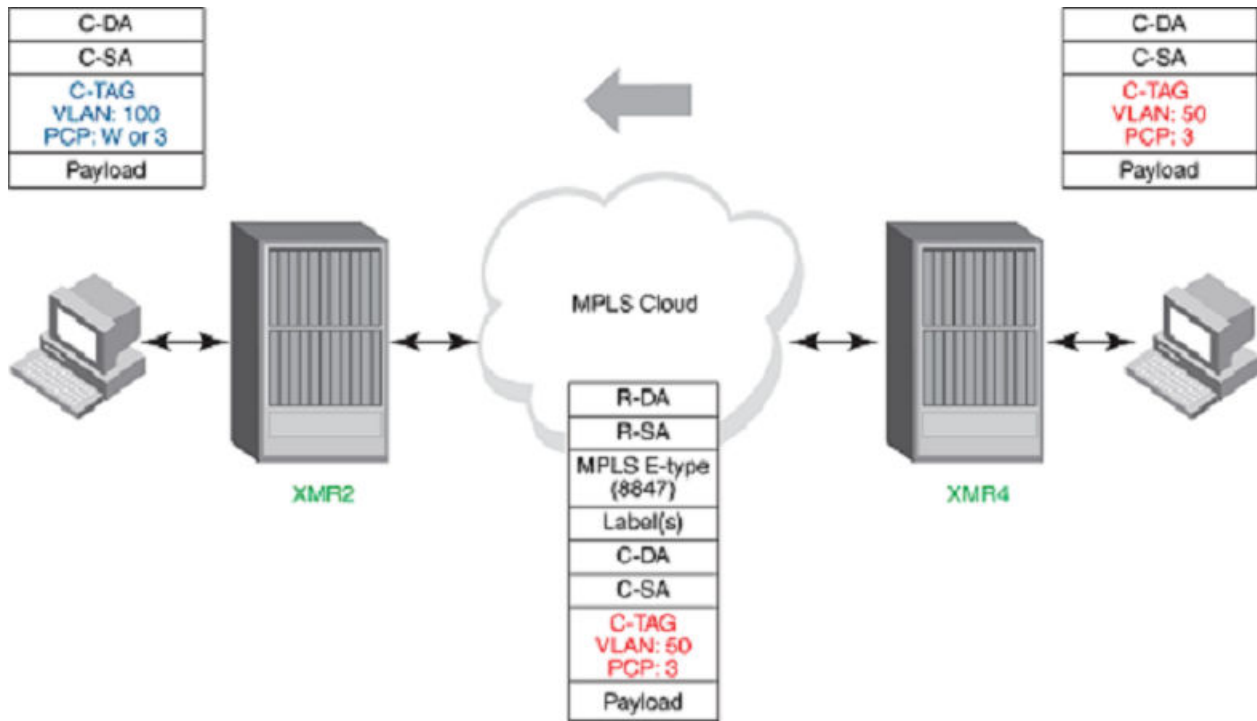
Interoperability with third party devices

This section assumes that the user understands how QoS works.

Third party device to Extreme device

CoS behavior for VPLS raw mode on page 282 describes the expected *Class of Service (CoS)* behavior for VPLS packets when VPLS raw pass through mode is in effect.

TABLE 21 Expected class of service behavior for VPLS raw pass through mode (third party device to Extreme device)

VPLS endpoints	Incoming packet		MPLS cloud		Outgoing packet	
Outer VLAN	Inner VLAN	Tunnel/VClabel (Z)	Payload tag	Outer VLAN	Inner VLAN	
Double-tag to Double-tag	X	Y	Vendor specific	X, Y	W or X	X
Single-tag to Double-tag	X	N/A	Vendor specific	X	W or X	X
Single-tag to Single-tag	X	N/A	Vendor specific	X	W or X	N/A
Untag to Untag	Any	Any	Vendor specific	Any	N/A	N/A
Double-tag to Single Tag	X	Y	Vendor specific	X, Y	W or X	N/A

W - Mapped CoS from internal priority (**Z** contributes to internal priority) using the CoS encode table.

X - Original outer VLAN CoS.

Y - Original inner VLAN CoS.

Z - Incoming EXP bits as described by the 'Tunnel' or 'VC label' column - **V** or internal priority.

The 'or' option in the 'Tunnel/VC label' column is to differentiate when the **qos exp encode** policy is ON (default) or OFF.

The 'or' option in the Outgoing Outer VLAN column is to differentiate when the **qos pcp encode policy** is ON (default) or OFF.

Third party device VC mode is set to RAW mode.

Specifying the VPLS VC type

The default VC type for all VPLS instances is set to 0x5 or "Ethernet". For compatibility with previous versions, the VC type can be changed to 0xB or "Ethernet VPLS". The VC type must match between peers for the VPLS session to be established.

To change the VPLS VC type, use the following command at the MPLS configuration level.

```
device(config-mpls)# vpls-vc-type-ethernet-vpls
```

VPLS VC type is 0xB (Ethernet VPLS) after the user saves to 'config' and reboots.

Syntax: [no] vpls-vc-type-ethernet-vpls

The default is raw mode.

Configuration considerations

- When the VPLS instance is configured with the raw pass through mode, all of its local endpoints must be either all untagged or all tagged. No intermix of modes is allowed.
- Whenever the VC mode configuration changes, whether it is from raw to raw pass through or vice versa, the VC for all the peers is torn down and then re-binds even though the actual VC-Type remains the same.

Configuring VPLS raw pass through mode

This section describes how to enable, disable, and view the configuration details of VPLS raw pass through mode.

Enabling VPLS raw-pass-through mode

To enable VPLS raw-pass-through mode, the user must first create the VPLS instance. When the VPLS instance does not already exist, then enter commands such as the following at the MPLS VPLS configuration level of the CLI.

```
device(config-mpls)# vpls test 100
device(config-mpls-vpls-test)# vc-mode raw-pass-through
```

Syntax: [no] vc-mode raw-pass-through

When the VPLS instance already has mixed endpoints configured, or the VC mode is already configured as raw-pass-through and an attempt to add a different type of endpoint then what it already contains, the following error message is displayed.

```
"Error: All endpoints under raw-pass-through must be all untagged or all tagged."
```

Disabling VPLS raw-pass-through mode

Except for VPLS instances on which ISID is configured, use the **no vc-mode raw-pass-through** command to disable VPLS raw pass through mode.

```
device(config-mpls)# vpls test 100
device(config-mpls-vpls-test)# no vc-mode raw-pass-through
```

For VPLS instances with an ISID configuration, first remove the ISID configuration, then disable VPLS raw-pass-through mode.

When the user attempts to disable VPLS raw pass through mode on a VPLS instance with ISID, the system displays the following error message.

```
device(config-mpls-vpls-test)# no vc-mode raw-pass-through
Error - Cannot remove tagged-mode setting while VPLS ISID configuration exists
```

Syntax: [no] vc-mode raw-pass-through

For information on displaying the raw mode configuration, refer to show mpls vpls detail on page 298.

Configuration example

The following is an example output of a running configuration with VPLS raw pass through mode configured.

```
router mpls
vpls test 100
vc-mode raw-pass-through
vpls-peer 10.100.100.100
vlan 100 inner-vlan 45
tag e 2/1
vpls name_raw 3
vpls-peer 10.200.200.200
vlan 300 inner-vlan 500
tagged ethe 3/1 ethe 3/11 ethe 3/13
vpls vctagged 200
vc-mode tagged
vpls-peer 10.300.300.300
vlan 200
tag e 2/2
```

VPLS tagged mode

VPLS tagged mode enables the preservation of the VLAN tag information in the payload. In VPLS tagged mode, the VLAN priority of the original (incoming) packets is carried across the MPLS cloud to remote peers.

By default, VPLS packets are sent across the MPLS cloud in raw mode. To use VPLS tagged mode, enable it per VPLS instance on both sides of the communicating edge routers. When this feature is enabled, the VLAN tag is determined as follows:

- When the original packet has one VLAN tag, the payload tag is the (outer) VLAN tag of the original packet.
- When the original packet has dual VLAN tags, the payload tag is the inner VLAN tag of the original packet.
- When the original packet is untagged, the payload tag is the configured VLAN on the VPLS untagged endpoint, and the CoS is zero.
- When the original packet has an I-component *Service Identifier (ISID)* tag, the payload tag is the unmodified ISID tag.

For more information about CoS behavior for VPLS tagged mode, see CoS behavior for VPLS tagged mode on page 287.

VPLS tagged mode must be enabled on both sides of the communicating edge routers. When the VPLS VC type does not match, the remote peer does not transition into operational state. Because each remote peer has its own operational state, the impact may differ

from one remote peer to another, depending on its current state. The remote peer state can be categorized into two general categories, as follows:

- *Remote peer in operational state* – When the remote peer is in operational state, a VC withdraw message and a new VC bind message is sent to the remote peer to tear down the current VC binding and to communicate the new VC type, respectively. This scenario assumes that the remote router is also an Extreme device running the same code level. The VC tear-down and re-bind should cause the remote peer to transition its peer state to "VC Parameter Check" state, because its own VC type is now mismatched with that of the new VC type received. Once the same tagged mode configuration is also applied to the remote router, the peer state for both routers should transition into operational state. As part of the VC tear-down, the hardware forwarding entries on the Interface module (LP) is cleaned up. When the peer transitions to operational state, its hardware forwarding entries is reprogrammed based on its tagged mode setting.

- *Remote peer not in operational state* – The category indicates that the VC has not yet been formed with the VPLS peer on the remote router. Remote peers may be in this category for many reasons (for example, "No local port defined", "No Tunnel", "No LDP Session", "VC Parameter Check", and so on). In this scenario, there is no need to tear down the VC binding. When the VPLS tagged mode configuration changes, most of the peers in this category does not change their operational state or perform any actions triggered by this configuration change. For remote peers that are in the state "VC Parameter Check" state because of a VC type mismatch, the configuration change triggers the sending of a VC bind message with the new VC type to the remote router. When the remote peer's VC type becomes compatible due to this configuration change and there is no other VC parameter mismatch, then the state of the remote peer transitions to the operational state.

To enable VPLS tagged mode, see Configuring VPLS raw pass through mode on page 285.

CoS behavior for VPLS tagged mode

NOTE
This section assumes that the user understands how QoS works.

Table 22 describes the expected *Class of Service (CoS)* behavior for VPLS packets when VPLS tagged mode is enabled.

TABLE 22 Expected Class of Service behavior for VPLS tagged mode

VPLS endpoints	Incoming packet		MPLS cloud		Outgoing packet		
Outer VLAN	Inner VLAN	Tunnel/VClabel (Z)	Payload tag	Outer VLAN	Inner VLAN		
Dual-tagged to dual-tagged	X	Y	V or internal priority	Y	W or Y	Y	
Single-tagged to dual-tagged	X	N/A		X	W or X	X	
Untagged to dual-tagged	N/A	N/A		O	W or O	O	
Dual-tagged to single-tagged	X	Y		Y	W or Y	N/A	

V - Mapped EXP bits from internal priority (**X** contributes to internal priority) using the EXP encode table.

W - Mapped CoS from internal priority (**Z** contributes to internal priority) using the CoS encode table.

X - Original outer VLAN CoS.

Y - Original inner VLAN CoS.

Z - Incoming EXP bits as described by the 'Tunnel/VC Label' column - **V** or internal priority.

- The 'Tunnel/VC Label' column differentiates the behavior between the **qos exp encode** policy when it is ON (default) or OFF.

- The 'Outgoing Packet Outer VLAN' column differentiates the behavior between the **qos pcp encode** policy when it is ON (default) or OFF.

QoS for VPLS traffic

By default, packets traveling through an MPLS domain are treated equally from a QoS standpoint, in a best effort manner. However, when a Layer 2 packet has an internal priority in its 802.1q tag, or the LSP or VPLS to which the packet is assigned has a configured *Class of Service (CoS)* value, QoS can be applied to the packet in the MPLS domain. The internal priority or CoS value is mapped to a value in the EXP field of the packet's MPLS header. The value in the EXP field is then mapped to an internal forwarding priority, and the packet is sent to the hardware forwarding queue that corresponds to the internal forwarding priority.

QoS for VPLS traffic at the ingress LER

The following methods can be used to provide QoS to packets entering a VPLS:

- Use the CoS value assigned to the tunnel LSP used to reach the VPLS peer.

When a tunnel LSP has a user-configured CoS value, all packets in all VPLS traveling through the tunnel LSP receive the same QoS.

- Use the CoS value assigned to the VPLS.

The VPLS CoS is a configurable option. Show commands will display the CoS if it is configured. If the CoS value it is not configured, the show commands will not display any CoS value.

When a CoS value is set for the VPLS, the device selects a tunnel LSP that also has this CoS value, when one is available. When no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VPLS).

When the selected tunnel LSP does not have a CoS value, the VPLS configured CoS value is used to provide QoS. The VPLS CoS value is mapped to a value in the EXP field. This allows traffic multiple VPLS using a single tunnel LSP, traffic from each VPLS can receive different QoS treatment.

- Use the priority in the packet's 802.1q tag.

When neither the tunnel LSP nor the VPLS has a configured CoS value, the device examines the priority in the Layer 2 packet's 802.1q tag, when the packet has one. Consequently, Layer 2 packets with the same 802.1q priority receive the same QoS in the VPLS.

- Use the configured priority of the port.

When neither the tunnel LSP nor the VPLS has a configured CoS value, and the Layer 2 packet does not have an 802.1q priority, QoS can be provided based on the priority of the incoming port. A port can be assigned a priority from zero (0) (lowest priority) to seven (7) (highest priority). The default port priority is zero (0).

By assigning different priorities to the ports where *Customer Edge (CE)* devices are connected (that is, the VPLS endpoints), the user can provide QoS to untagged Layer 2 traffic received from different customer locations.

When a packet enters a VPLS, the PE router that serves as both the VPLS endpoint and the ingress of a tunnel LSP pushes two labels onto the packet the inner VC label and the outer tunnel label. The packet's priority resides in the EXP field of the MPLS label header. The VC label and the tunnel label carry the same value in the EXP field.

The following table lists how a Layer 2 packet's priority is mapped to a value in the EXP field and how the EXP value is mapped to a priority queue.

Tunnel LSP configured CoS or VPLS configured CoS or 802.1q priority or Configured port priority	Value placed in the tunnel and VC label EXP field	Priority queue
7	7	qosp7 (highest priority)
6	6	qosp6
5	5	qosp5
4	4	qosp4
3	3	qosp3
2	2	qosp2
1	1	qosp1
0	0	qosp0 (best effort)

Specifying an LSP to reach a peer within a VPLS

The user can specify the LSPs that can be used to reach a peer within a VPLS. The user can specify up to four *Resource ReSerVation Protocol (RSVP)* LSPs per VPLS peer. VPLS subsequently selects one of the LSPs configured to reach the specified peer. Any of the configured LSPs can be used, and the order of configuration is not relevant to the selection of the LSP. When none of the assigned LSPs is operational, the VPLS session with the peer is down. An LSP is considered down when the LSPs primary, secondary, and detour paths are all down.

RSVP LSPs must be pre-configured prior to their assignment to the VPLS peer. Additionally, the VPLS peer's IP address must match the target IP address of any RSVP LSPs assigned to it. When these addresses do not match, the configuration is rejected. An LSP that is assigned to any VPLS is not allowed to be deleted from the configuration unless the VPLS LSP assignment is deleted first. When no LSPs have been assigned to a VPLS peer, the existing mechanism is used to select an appropriate LSP for the peer.

When LSP assignment is configured, ignore the configured CoS of the LSP, and ignore the VPLS to select an LSP for the VPLS peer. However, traffic sent on the LSP uses the CoS of the LSP. When LSP load balancing is enabled for a VPLS peer, traffic is load-balanced on all assigned LSPs that are operational.

To specify LSPs for a VPLS peer within a VPLS instance, enter a command such as the following.

```
device(config-mpls-vpls-v1)# vpls-peer 192.168.0.0 lsp t1 t2 t3 t4
```

Syntax: vpls-peer *ip-address* **lsp** *lsp1* /*lsp2* / *lsp3* / *lsp4*]

The *ip-address* variable specifies the IP address of the VPLS peer to which the user wants to assign LSPs.

The *lsp1 / lsp2 / lsp3 / lsp4* variables are the names of the LSPs that the user wants to assign to the VPLS peer. The user can assign up to four LSPs to a peer using this command. When a VPLS peer is not assigned any LSPs, the default mechanisms for selecting an LSP for the VPLS peer are used.

LSP load balancing for VPLS traffic

In a VPLS instance, traffic from one VPLS peer to another is forwarded over an MPLS tunnel LSP. When more than one tunnel LSP exists from the device to a VPLS peer, the device can select multiple tunnel LSPs to forward VPLS traffic to the peer. Known unicast traffic is load-balanced across the selected tunnel LSPs. Broadcast and unknown unicast traffic is always sent over a single tunnel LSP, however.

For VPLS LSP load-balancing, select an LSP based on a hash-index which is calculated as follows:

> **NOTE**
> For VPLS traffic, source and destination MAC addresses come from the inner customer Ethernet header.

- **Layer-2, non-IPv4, and IPv6 packets:** Source MAC address and destination MAC address.
- **IPv4, non-TCP/UDP packets:** Source MAC address and destination MAC address, source IP address and destination IP address.
- **IPv4 TCP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and TCP source port and TCP destination port.
- **IPv4 UDP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and UDP source port and UDP destination port.
- **IPv6 non-TCP/UDP packets:** Source MAC address and destination MAC address, source IP address and destination IP address.
- **IPv6 TCP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and TCP source port and TCP destination port.
- **IPv6 UDP packets:** Source MAC address and destination MAC address, source IP address and destination IP address, and UDP source port and UDP destination port.

A tunnel LSPs eligibility for load balancing depends on whether CoS values are defined for the VPLS instance and the tunnel LSP:

- When the VPLS instance does not have a CoS value defined, then all tunnel LSPs to the peer are eligible for load balancing.
- When a VPLS instance has a CoS value defined, and at least one tunnel LSP to the peer has a CoS value less than or equal to the VPLS instance CoS value, then all tunnel LSPs with the highest CoS value less than or equal to the VPLS instance CoS value are eligible for load balancing.
- When a VPLS instance has a CoS value defined, and none of the tunnel LSPs to the peer has a CoS value less than or equal to the VPLS instance CoS value, then all tunnel LSPs to the peer that do not have a CoS value are eligible for load balancing.

> **NOTE**
> The LSPs picked for load-balancing must have the same CoS values. For example: When CoS of LSP1 = 4, LSP2 = 4, LSP3 = 2, LSP4 =2, LSP5 = 1 and VPLS instance CoS = 3. Then traffic is load balanced with LSP3 and LSP4 which has same CoS values.

LSP load balancing

The device evenly distributes VPLS traffic across tunnel LSPs.

In early software releases, VPLS traffic was unevenly balanced across tunnel LSPs when exactly three tunnels were used for load balancing. For example, for tunnels A, B, and C, VPLS traffic might be distributed among the tunnels as follows: A: 50%, B: 25%, and C: 25%.

Now, the tunnels are fully utilized. Using the same example above, VPLS traffic might be distributed among tunnels A, B, and C as follows: A: 33.3%, B: 33.3%, and C: 33.3%. These percentages are based on a fully distributed hash index generated by the incoming traffic. Actual distribution percentages may vary and are based on the hash index.

Configuring LSP load balancing for VPLS traffic

To configure a VPLS instance to load balance known unicast traffic sent to a VPLS peer across multiple tunnel LSPs, enter a command such as the following.

```
device(config-mpls-vpls-v1)# vpls-peer 192.168.0.0 load-balance
```

Syntax: [no] vpls-peer *ip-addr* [**load-balance**]

> NOTE
> To disable the LSP load balancing, the user must delete the VPLS peer with the **no vpls-peer** command, then re-enter the
> **vpls-peer** command without the **load-balance** option.

In the prior example, when the **load-balance** option is specified, VPLS traffic originating from the device and sent to peer 192.168.0.0 is
load balanced across eligible tunnel LSPs whose destination is the peer.

VPLS LSP load balancing

TABLE 23 VPLS LSP load balancing terms

Term	Meaning
MAC	Media Access Control
LSP	Label Switched Path
VPLS	Virtual Private LAN Service

This functional specification documents the VPLS LSP load balancing which is to be incremented from four LSPs to eight LSPs for the
XMR Series and MLX Series product lines.

Limitations and prerequisites

The hashing technique to load balance is not consistent. This is an existing limitation.

Feature enhancement

This feature increases the number to eight LSPs to a VPLS peer.

In the CES 2000 Series and the CER 2000 Series, you would be able to assign up to eight LSP to a VPLS peer but at any time. Only
one of them is chosen for all traffic forwarding for this VPLS peer because load balancing is not supported in both the CES 2000 Series
and the CER 2000 Series.

Assumptions and dependencies

The hashing decision has not changed for this feature support. It is based on the fields in each packet received.

VPLS can use both LDP and RSVP tunnels for load balancing as long as they all matched the CoS criteria. A tunnel reachable to a peer
with the right CoS value is all that is required to be used as a candidate for VPLS tunnel load balancing. There are no preferences over a
particular type of tunnel dependencies.

Specifying the endpoint of a VPLS instance

When the user configures the VPLS endpoint, the user specifies what happens to packets exiting the VPLS instance, which VLAN the
packet belongs to, as well as whether it is transmitted from the PE device to the CE device over a dual-tagged, single-tagged, or
untagged port. The user can also specify a server *Link Aggregation Group (LAG)* group as the endpoint of a VPLS instance.

A VPLS instance can be configured between any combination of dual-tagged, single-tagged, and untagged endpoints. For dual-tagged ports, traffic flows between the dual-tagged endpoint and the MPLS cloud are also supported for traffic switched between a local endpoint and remote peers.

> **NOTE**
> Unless VPLS tagged mode is enabled, VPLS operates in raw mode, meaning no VLAN tags are carried across the MPLS cloud to remote peers. For more information, see

The *Customer Edge (CE)* device is connected to the PE router over one or more dual-tagged, single-tagged, or tagged ports.

- With a *single-tagged* port, each pair (port, VLAN ID) is identified as a unique endpoint. When VPLS raw mode is in effect, the tag is of significance between the CE and the PE and is not sent across the MPLS cloud. When VPLS tagged mode is enabled, the tag is sent across the MPLS cloud.

- In the case of an *untagged* port, an endpoint is identified by the physical port alone, and the packets are sent in untagged Ethernet format within the MPLS payload.

- In the case of a *dual-tagged* port, the packets contain both an outer VLAN tag and an inner VLAN tag. In this configuration, an endpoint can receive packets with two tags and forward them to the other endpoint either single-tagged or dual-tagged. When VPLS tagged mode is enabled, the inner VLAN tag is sent across the MPLS cloud.

All VPLS endpoints can be dual mode ports (tagged-untagged). An untagged endpoint port is removed from the default VLAN ID 1 and cannot be added back to the default VLAN. A VPLS endpoint can be tagged in multiple VPLS and Layer 2 VLANs and untagged in one other VLAN.

Special considerations for dual-tagged endpoints

Before configuring a dual-tagged VPLS endpoint, consider the following:

- The *Tag Protocol Identifier (TPID)* of the inner VLAN tag must be 0x8100 be classified as dual-tagged and recognized by dual-tagged endpoints. When the TPID is not 0x8100, the packet is classified as a single-tagged packet.

- The TPID of the outer VLAN tag must be the port's configured tag type (the default tag type is 0x8100).

- The System Max value for the *Internal Forwarding Lookup (IFL)* CAM partition must not be set to 0 (zero). When it is set to zero, an informational message such as the following is displayed.

```
device(config-mpls)# vpls test 10
device(config-mpls-vpls-test10)# vlan 100 inner-vlan 200
device(config-mpls-vpls-test_10-vlan-100-inner-vlan-200)# tagged Ethernet2/1
```

> **NOTE**
> The system-max size for the Internal Forwarding Lookup CAM is zero. Use the command **system-max ifl-cam** to specify a size.

The informational message only warns that the configuration must be changed. It does not cause the system to reject the VPLS configuration. For example, in the sample case, the dual-tagged endpoint configuration of vlan 100 inner-vlan 200 on port ethernet 2/1 has been accepted assuming the port, outer VLAN, and inner VLAN combination has not already been assigned elsewhere.

- When the IFL CAM partition on the Interface module exceeds a configured threshold, there is a warning log message which is similar to the way other CAM partitions are handled currently. The system does not generate any logs when it cannot program the IFL CAM because of exhaustion of an IFL CAM resource.

- When an outer VLAN is specified for a given endpoint, it is called a less-specific VLAN. When both an outer VLAN and inner VLAN are specified, it is called a more-specific VLAN (in relation to the outer VLAN).

- Similar to single-tagged endpoints, the outgoing VLANs for a dual-tagged endpoint are based solely on the outgoing endpoint configuration, and not on the incoming packet VLAN values.

- The same port, outer VLAN, and inner VLAN combination cannot be specified across VPLS instances. For example, when a dual-tagged endpoint with VLAN 100 and inner VLAN 200 is configured on port ethernet 2/1 on VPLS instance "test", same endpoint cannot be configured as part of another VPLS instance (for example, "test 1").This is also true across applications. When a port, outer VLAN, and inner VLAN combination belongs to a VPLS instance, it cannot simultaneously belong to a Layer 2 VLAN, local VLL, or VLL.

- When CPU protection is enabled for a VPLS instance, the system does not support a configuration with two different dual-tagged VPLS VLANs as part of the same VPLS instance. Consider the following configuration example.

```
device(config)# router mpls
device(config-mpls)# vpls test 10
device(config-mpls)# cpu-protection
device(config-mpls-vpls-test)# vlan 10 inner-vlan 20
device(config-mpls-vpls-test-vlan-10-20)# tagged eth 2/1
device(config-mpls-vpls-test-vlan-10)# exit
device(config-mpls-vpls-test)# vlan 10 inner-vlan 30
device(config-mpls-vpls-test-vlan-10-30)# tagged eth 2/1
Error - VPLS port 2/1 cannot be shared by multiple end-points when CPU protection is enabled. Remove CPU
protection for VPLS 10 to make this configuration change.
```

Similarly, CPU protection cannot be enabled for a VPLS instance that has a port configured under two different dual-tagged VPLS VLANs. Consider the following configuration example.

```
device(config)# router mpls
device(config-mpls)# vpls test 10
device(config-mpls-vpls-test)# vlan 10 inner-vlan 20
device(config-mpls-vpls-test-vlan-10-20)# tagged eth 2/1
device(config-mpls-vpls-test-vlan-10)# exit
device(config-mpls-vpls-test)# vlan 30 inner-vlan 40
device(config-mpls-vpls-test-vlan-30-40)# tagged eth 2/1
device(config-mpls-vpls-test-vlan-20)# exit
device(config-mpls-vpls-test)# cpu-protection
Error - Cannot configure CPU protection for VPLS 10 as multiple end-points share the same physical port.
```

The restrictions exist because packets are hardware-forwarded when CPU protection is enabled. In this case, source port suppression cannot be properly performed when there are multiple endpoints on the same physical interface.

Specifying an untagged endpoint

To specify an untagged endpoint for a VPLS instance, enter commands such as the following.

```
device(config-mpls)# vpls v1 40000
device(config-mpls-vpls-v1)# vlan 100
device(config-mpls-vpls-v1-vlan-100)# untagged ethernet 2/1
```

Syntax: [no] untagged [ethernet] *portnum*

> NOTE
> *Foundry Discovery Protocol (FDP)* must not be enabled on an untagged VPLS or VLL endpoint.

Specifying a single-tagged endpoint

Tagged ports are configured under a VLAN ID. A VPLS instance can have multiple ports configured under the same VLAN ID, and can have ports configured under different VLAN IDs. Another VPLS instance can reuse the same VLAN ID on other physical ports. Because the VLANs are configured under different VPLS instances, they are different VPLS VLANs even though they use the same VLAN ID.

To specify a tagged endpoint for a VPLS instance, enter commands such as the following:

```
device(config-mpls)# vpls v1 40000
device(config-mpls-vpls-v1)# vlan 200
device(config-mpls-vpls-v1-vlan-200)# tagged ethernet 3/11
```

Syntax: vlan *num*

Syntax: [no] tagged ethernet *slot/port*

Specifying a dual-tagged endpoint

Dual-tagged ports are configured with two VLAN IDs. A VPLS instance can have multiple ports configured under the same dual-tagged VLAN ID, and can have ports configured under different VLAN IDs. Another VPLS instance can reuse the same VLAN ID on other physical ports. Because the VLANs are configured under different VPLS instances, they are different VPLS VLANs even though they use the same VLAN ID.

> **NOTE**
> Before configuring a dual-tagged endpoint, see Special considerations for dual-tagged endpoints on page 292.

To specify a dual-tagged endpoint for a VPLS instance, use the following commands.

```
device(config-mpls)# vpls v1 40000
device(config-mpls-vpls-v1)# vlan 200 inner-vlan 300
device(config-mpls-vpls-v1-vlan-200)# tagged ethernet 3/11
```

Syntax: [no] vlan *VLAN-ID* **inner-vlan** *VLAN-ID*

Syntax: [no] tagged ethernet *slot/port*

The **vlan** *VLAN-ID* variable, which is the outer VLAN ID, can be in the range from 1 through 4094 and excludes the default VLAN.

The **inner-vlan** *VLAN-ID* variable, can be in the range from 1 through 4095 and includes the default VLAN.

Use the [**no**] form of the command to remove the dual-tagged VPLS VLAN configuration and its associated endpoints. For example, the command **no vlan** *200* **inner-vlan** *300* removes the dual-tagged VLAN and associated endpoints. The single-tagged VLAN, **vlan 200**, is not deleted. Similarly, the command **no vlan** *200* removes the single-tagged VLAN, **vlan 200**, and associated endpoints. The dual-tagged VLAN, **vlan** *200* **inner-vlan** *300* , is not deleted.

Example of dual-tagged endpoints mapped to different VPLS instances

The following example shows two dual-tagged endpoints on the same physical interface with the same outer VLAN ID and different inner VLAN IDs mapped to different VPLS instances.

```
device(config-mpls)# vpls test_10 10
device(config-mpls-vpls-test_10)# vlan 100 inner-vlan 200
device(config-mpls-vpls-test_10-vlan-100-inner-vlan-200)# tagged ethernet 2/1
device(config-mpls-vpls-test_10-vlan-100-inner-vlan-200)# exit
device(config-mpls-vpls-test_10)# exit
device(config-mpls)# vpls test_20 20
device(config-mpls-vpls-test_20)# vlan 100 inner-vlan 300
device(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)# tagged ethernet 2/1
device(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)# exit
```

Example of dual-tagged endpoints mapped to the same VPLS instance

The following example shows two dual-tagged endpoints on the same physical interface with the same outer VLAN ID and different inner VLAN IDs mapped to the same VPLS instance.

```
device(config-mpls)# vpls test 10
device(config-mpls-vpls-test)# vlan 100 inner-vlan 200
device(config-mpls-vpls-test-vlan-100-inner-vlan-200)# tagged ethernet 2/1
device(config-mpls-vpls-test-vlan-100-inner-vlan-200)# exit
device(config-mpls-vpls-test)# vlan 100 inner-vlan 300
device(config-mpls-vpls-test-vlan-100-inner-vlan-300)# tagged ethernet 2/1
```

In the above example, when packets are received with outer VLAN ID 100 and no inner VLAN ID, the packets are not handled as part of VPLS instance 10.

Example of a less-specific and more-specific VLAN mapped to different VPLS instances

The following example shows a less-specific VLAN and more-specific VLAN (with the same outer VLAN ID) of the same port in different VPLS instances.

```
device(config-mpls)# vpls test_10 10
device(config-mpls-vpls-test_10)# vlan 100
device(config-mpls-vpls-test_10-vlan-100)# tagged ethernet 2/1
device(config-mpls-vpls-test_10-vlan-100)# exit
device(config-mpls-vpls-test_10)# exit
device(config-mpls)# vpls test_20 20
device(config-mpls-vpls-test_20)# vlan 100 inner-vlan 300
device(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)# tagged ethernet 2/1
device(config-mpls-vpls-test_20-vlan-100-inner-vlan-300)# exit
```

Example of a less-specific and more-specific VLAN mapped to the same VPLS instance

The following example shows a less-specific VLAN and more-specific VLAN (with the same outer VLAN ID) of the same port in the same VPLS instance.

```
device(config-mpls)# vpls test_10 10
device(config-mpls-vpls-test_10)# vlan 100
device(config-mpls-vpls-test_10-vlan-100)# tagged ethernet 2/1
device(config-mpls-vpls-test_10-vlan-100)# exit
device(config-mpls-vpls-test_10)# vlan 100 inner-vlan 300
device(config-mpls-vpls-test_10-vlan-100-inner-vlan-300)# tagged ethernet 2/1
device(config-mpls-vpls-test_10-vlan-100-inner-vlan-300)# exit
```

In the above example, when packets are received on interface ethernet 2/1 with outer VLAN ID 100 and an inner VLAN ID other than 300, the packets are handled as part of VPLS instance 10. In this case, the inner VLAN is treated as payload.

Specifying a LAG group as the endpoint of a VPLS instance

The endpoint of a VPLS instance can be a static or a dynamic LAG. When the endpoint of a VPLS instance is a LAG, the VPLS traffic load is distributed to the CE device across all of the LAG's ports by way of a hashing mechanism that utilizes the source and destination MAC addresses.

For example, to configure a LAG, enter commands such as the following.

```
device(config)# lag blue dynamic
device(config-lag-blue)# ports ethernet 1/1 to 1/2
device(config-lag-blue)# primary-port 1/1
```

To configure a VPLS instance that uses the LAG defined as the endpoint by the previous example commands, enter the commands as in the following example.

```
device(config)# router mpls
device(config-mpls)# vpls test1 40000
device(config-mpls-vpls-test1)# vpls-peer 10.10.10.10
device(config-mpls-vpls-test1)# vlan 200
device(config-mpls-vpls-test1)# tagged e 1/1
```

When the user first creates a LAG and then configure a VPLS instance, the port the user specifies as the VPLS endpoint must also be the port the user specified as the primary port of the LAG group:

- When the user first configures a VPLS instance and then create a LAG, all ports of the LAG must be specified as endpoints of the VPLS instance. The VPLS instance uses all the ports of the LAG.

- When the user later deletes the LAG from the configuration, all ports in the LAG become independent endpoints in the VPLS instance.
- When the user specified a tagged endpoint for the VPLS instance, all of the ports in the LAG must be tagged.
- Traffic received from any port in the LAG is forwarded to the VPLS instance. All traffic is matched to its VLAN.

Support for VPLS endpoints within a Topology group

The user can configure VPLS VLANs into Topology groups so that the user can use any of the following protocols within a VPLS VLAN:

- *Spanning Tree Protocol (STP)*
- *Rapid Spanning Tree Protocol (RSTP)*
- *Foundry Metro Ring Protocol (MRP)*
- *Virtual Switch Redundancy Protocol (VSRP)*

Flooding Layer 2 BPDUs in VPLS

By default, Layer 2 STP and *Per VLAN Spanning Tree (PVST) Bridge Protocol Data Units (BPDUs)* entering a VPLS endpoint are not transparently flooded within the VPLS instance. The BPDUs are dropped when they enter the VPLS endpoint. The user can change this default behavior to not block BPDUs and transparently flood them within the VPLS instance, by configuration on a per-physical-port basis. Because the BPDU block option is configurable per physical interface, it affects all VPLS instances that have endpoints on that interface.

To flood BPDUs in VPLS, use the following command.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# no vpls-bpdu-block
```

Syntax: [no] vpls-bpdu-block

Specifying the VPLS VC type

> **NOTE**
> This is a global setting that affects all VPLS instances, except those in VPLS tagged mode. The user must save the configuration and reload the software to place the change into effect.

The default VC type for all VPLS instances is set to 0x5 or "Ethernet". For compatibility with previous versions, the VC type can be changed to 0xB or "Ethernet VPLS". The VC type must match between peers for the VPLS session to be established.

> **NOTE**
> When VPLS tagged mode is enabled for a VPLS instance, the VC type for that instance is set to 0x04 or "Ethernet Tagged", regardless of the global VPLS VC type setting.

To change the VPLS VC type, use the following command at the MPLS configuration level.

```
device(config-mpls)# vpls-vc-type-ethernet-vpls
```

VPLS VC type is 0xB (Ethernet VPLS) after the user saves to 'config' and reboot.

Syntax: [no] vpls-vc-type-ethernet-vpls

Configuring VPLS tagged mode

This section describes how to enable, disable, and view the configuration details of VPLS tagged mode. For details about how VPLS tagged mode works, see VPLS tagged mode on page 286.

Enabling VPLS tagged mode

To enable VPLS tagged mode, first create the VPLS instance when it does not already exist, and then enter commands such as the following at the MPLS VPLS configuration level of the CLI.

```
device(config-mpls)# vpls test 100
device(config-mpls-vpls-test)# vc-mode tagged
```

Syntax: vc-mode tagged

Disabling VPLS tagged mode

Except for VPLS instances on which ISID is configured, use the **no vc-mode tagged** command to disable VPLS tagged mode. For VPLS instances with an ISID configuration, first remove the ISID configuration, then disable VPLS tagged mode.

When the user attempts to disable VPLS tagged mode on a VPLS instance with ISID, the system displays the following error message.

```
device(config-mpls-vpls-test)# no vc-mode tagged
Error - Cannot remove tagged-mode setting while VPLS ISID configuration exists
```

Syntax: no vc-mode tagged

Viewing the VPLS tagged mode configuration

Use the **show running config** and **show mpls vpls detail** commands to view the VPLS tagged mode configuration. The following shows an example **show running config** output. For examples and details of the **show mpls vpls detail** command, see Enabling MPLS VPLS traps on page 305.

```
device(config-mpls)# vpls test 100
device(config-mpls-vpls-test)# vc-mode tagged
device# show running config
....
router mpls
vpls test 100
  vc-mode tagged
  vpls-peer 10.100.100.100
  vlan 100 inner-vlan 45
    tag e 2/1
vpls name_raw 3
  vpls-peer 10.200.200.200
```

In the above example, **vc-mode tagged** indicates that VPLS tagged mode is enabled on **vpls test 100,** whereas **vpls name_raw 3** is in VPLS raw mode.

show mpls vpls detail

Use the **show mpls vpls detail** command to view the VPLS raw pass through mode configuration. A manual LSP assignment for a peer can now accept a maximum of eight LSPs instead of just four LSPs.

Syntax

show mpls vpls detail

Parameters

detail   Displays detailed information for each VPLS.

Modes

Global configuration mode.

Command Output

The **show mpls vpls detail** command displays the following information:

Field	Description
VPLS	The configured name of the VPLS instance.
Id	The ID of this VPLS instance.
Max mac entries	The maximum number of MAC address entries that can be learned for this VPLS instance. This is a soft limit only and can be exceeded when there is space available in the VPLS MAC database.
Total vlans	The number of VLANs that are translated for this VPLS instance.
Tagged ports	The total number of tagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
Untagged ports	The total number of untagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
IFL-ID	The *Internal Forwarding Lookup Identifier (IFL-ID)* for dual-tagged ports in the VPLS instance.
Vlan	The ID of each VLAN in this VPLS instance.
L2 Protocol	
Tagged	The numbers of the tagged ports in each VLAN.
Untagged	The numbers of the untagged ports in each VLAN.
VC-Mode	The VC mode for the VPLS instance: • Raw – The VLAN tag information in the original payload is not carried across the MPLS cloud • Tagged – The VLAN tag information in the original payload is carried across the MPLS cloud • Raw-pass-through –The VLAN tag information behaves like tagged mode when all endpoints are configured as tagged endpoints
Total VPLS peers	The number of VPLS peers this device has for this VPLS instance, as well as the number of these VPLS peers with which this device has an LDP session.

Field	Description
Peer address	The IP address of the VPLS peer.
State	The current state of the connection with the VPLS peer. This can be one of the following states: • Operational – The VPLS instance is operational. Packets can flow between the device and the peer • Wait for functional local ports – The physical endpoint port that must be connected to the Customer Edge device is down due to a link outage or is administratively disabled • Wait for LSP tunnel to Peer – The device cannot find a working tunnel LSP • Wait for PW Up (Wait for LDP session to Peer)– The LDP session is not yet ready • Wait for PW Up (Wait for remote VC label) – The device has advertised its VC label binding to the VPLS peer, but has not yet received the peer's VC label binding • Wait for PW Up (VC type mismatched) - A session is not formed because the VC type does not match with its peer's VC type • Wait for PW Up (MTU mismatched) - The MTU sent to a peer is derived from the device's global setting by the following formula: (*system-mtu* minus 26 bytes). When a *system-mtu* value is not configured, a default value of 1500 is sent • Wait for PW Up (Wait for LPD session to Peer) - The LDP session to the peer is down • Wait for PW Up (No Label Resource) - When configuring a new VPLS peer, the maximum amount of VC labels that can be supported may exceed 64K, and cause the configuration to be rejected. The maximum amount of VC labels available for VPLS instances is equal to 64K.
Uptime	The time in minutes that the entry has been operational.
Tnnl in use	The tunnel LSP used to reach the VPLS peer. When VPLS traffic to the peer is load balanced across multiple tunnel LSPs, the tunnel LSPs used to reach the peer are displayed.
Local VC lbl	The VC label value locally allocated for this peer for this VPLS instance. Packets forwarded from the VPLS peer to this device are expected to contain this label. This is the label that is advertised to the VPLS peer through LDP.
Remote VC lbl	The VC label allocated by the VPLS peer and advertised to this device through LDP. The device applies this label to outbound MPLS packets sent to the VPLS peer.
Local VC MTU	The MTU value locally configured for this peer.
Remote VC MTU	The MTU value configured for the remote VPLS peer.
Local VC-Type	The VC type for this peer.
Remote VC-Type	The VC type for the remote VPLS peer.
CPU-Protection	Whether CPU protection configured on this VPLS instance is ON or OFF. On XMR Series and MLX Series devices only: When CPU protection is enabled on this VPLS instance but is temporarily unavailable due

Field	Description
	to 100% multicast FID usage, this field includes the message shown above.
Local Switching	Whether local switching behavior on a per-VPLS basis is enabled or disabled.
Extended Counter	Indicates whether or not the extended counter is enabled for the configured VPLS.
Multicast Snooping	Indicates whether the multicast snooping is enabled or disabled.

Examples

Example of output of the **show mpls vpls detail** command.

```
device# show mpls vpls detail
VPLS v2, Id 200, Max mac entries: 2048
Total vlans: 2, Tagged ports: 2 (2 Up), Untagged ports 0 (0 Up)
IFL-ID: n/a
Vlan 10
L2 Protocol: NONE
Tagged: ethe 2/19
Vlan 20
L2 Protocol: NONE
Tagged: ethe 2/20
VC-Mode: Raw
Total VPLS peers: 1 (1 Operational)
Peer address: 19.19.19.19, State: Operational, Uptime: 1 min
Tnnls in use (load balance): Candidate count:8 (only 1st 8 is displayed):
tnl3(3)[RSVP] tnl5(3)[RSVP] tnl0(3)[RSVP] tnl7(3)[RSVP] tnl2(3)[RSVP]
tnl4(3)[RSVP] tnl1(3)[RSVP] tnl6(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)
CPU-Protection: OFF
Local Switching: Enabled
Extended Counter: ON
Multicast Snooping: Disabled
```

CAM2PRAM format is defined below:

CAM2PRAM Data Format				
Bits[31:25]	Bit[24]	Bit[23]	Bits[22:19]	Bits[18:0]
Reserved	ECMP_MASK[4]	LOCAL ADDRESS (Used for DA-MAC Aging)	ECMP_MASK[3:0]	PRAM_INDEX_BASE[18:0]

```
device# show cam l2vpn 2/19 0000.00b0.00b0
Slot Index MAC Age Port IFL/ VC Label Out Port Remote DA/ PRAM
(Hex) VLAN SA (Hex)
2 9f9df 0000.00b0.00b0 0 2/20 20 N/A 2/15 0 DA 0033e
2 9fb38 0000.00b0.00b0 2 N/A N/A 983040 Drop 1 SA 003d7
device# dm cam me/15 0x9f9df
(CAM 0x9f9df): data 0072-00140000-00b000b0 mask ffff-ffffffff-ffffffff Valid
(Shadow CAM): data 0072-00140000-00b000b0 mask ffff-ffffffff-ffffffff
(Access): device 2, block 7, block index 0x000039df
(Flags): VALID 1 INUSE(1, 0) RESTORING(0, 0) AGING(1, 0) AGE COUNT(0, 0)
(CAM2PRAM entry 0x13f3be): 0038033e
(CAM2PRAM entry 0x13f3bf [MAC SA or Right IP]): 00380001
```

For the above case, the value of **0038033e** indicates an ECMP mask of seven (eight tunnels).

The ECMP value must always be one less than the actual number of tunnels.

Release version	Command history
Multi-Service IronWare Release Unknown	This command was introduced.
Multi-Service IronWare Release 5.6.00	This command was modified for the option to a VPLS manual LSP assignment for a peer. It can now accept a maximum of eight LSPs instead of four LSPs.

VPLS CPU protection

The VPLS CPU protection feature protects the CPU of the line card from being overwhelmed by excessive VPLS packets that would require the CPUs attention, including unknown unicast, multicast packets, and packets requiring source-MAC learning. Once this feature is enabled, all VPLS multicast traffic is hardware-flooded. Furthermore, when the CPU is too busy, this feature hardware-floods unknown unicast traffic, as well as reduce the rate of source-MAC learning traffic to the line card CPU, so that the line card CPU has enough resources to handle other types of packets.

> **NOTE**
> VPLS CPU protection is not applicable to CES 2000 Series or CER 2000 Series devices.

Configuration Considerations

Note the following configuration rules before enabling VPLS CPU protection.

- VPLS CPU protection cannot be concurrently enabled with IGMP snooping on a VPLS instance.
- CPU protection cannot be enabled for a VPLS instance that has a port configured under two different VPLS VLANs. Similarly, when CPU protection is enabled for a VPLS instance, the system does not support a configuration with two different VPLS VLANs as part of the same VPLS instance.
- When VPLS FID usage reaches 100%, CPU protection is temporarily disabled until adequate FID resources are available.

Configuring VPLS CPU protection

VPLS CPU protection can be configured in either of the following two ways:

- Globally - This enables VPLS CPU protection to affect all VPLS instances on the router
- Per-VPLS - This enables VPLS CPU protection on one or more specified VPLS instances

Configuring VPLS CPU protection globally

VPLS CPU protection can be enabled for all VPLS instances on a router. To enable VPLS CPU protection on all VPLS instances, enter the following command.

```
device(config)# router mpls
device(config-mpls) vpls-cpu-protection
```

Syntax: **[no] vpls-cpu-protection**

Configuring VPLS CPU protection per VPLS

VPLS CPU protection can be enabled per VPLS instance. To enable VPLS CPU protection on a specified VPLS instance, enter the following command.

```
device(config)# router mpls
device(config-mpls)# vpls test 1
device(config-mpls-vpls-test)# cpu-protection
```

Syntax: [no] cpu-protection

Layer 2 control traffic behavior on VPLS endpoints

This section describes the Layer 2 control traffic behavior on VPLS endpoints.

802.1x Protocol packets on a VPLS endpoint

802.1x does not support VPLS endpoints.

Cisco Discovery Protocol packets

Cisco Discovery Protocol (CDP) cannot be configured on a VPLS endpoint port and a VPLS endpoint cannot be configured on a physical port that has CDP enabled. This restriction is enforced by the CLI. When a VPLS endpoint receives any CDP traffic, this traffic is transparently flooded within the VPLS.

The behavior of CDP control packets is as follows:

- When CDP is globally enabled on the device, and the **priority force** command is configured on an incoming port, the VPLS local switched packets is sent out with a priority of seven (7).
- When CDP is not enabled on the device, packets are switched locally according to the priority in the configured **qos exp encode** command or **qos pcp encode-policy** command.

Foundry Discovery Protocol packets

Foundry Discovery Protocol (FDP) cannot be configured on a VPLS endpoint port and a VPLS endpoint cannot be configured on a physical port that has FDP enabled. This restriction is enforced by the CLI. When a VPLS endpoint receives any FDP traffic, this traffic is transparently flooded within the VPLS.

The behavior of FDP control packets is as follows:

- When FDP is globally enabled on the device and **priority force** command is configured on an incoming port, the VPLS local switched packets is sent out with a priority of seven (7)
- When FDP is not enabled on the device, packets are switched locally according to the priority in the configured **qos exp encode** command or **qos pcp encode-policy** command

Configuring VPLS endpoint over FDP/CDP enabled interface

Configuring VPLS endpoint over an FDP/CDP enabled interface will implicitly disable FDP/CDP configuration and also will be enable back implicitly when the VPLS endpoint is deleted on that specific interface, considering FDP/CDP is enabled globally.

Info messages are displayed to notify the user as below for these cases:

For example, when VPLS endpoint is created the info messages displayed are as below.

- When only FDP is enabled globally.

```
device(config-mpls-vpls-vpls1-vlan-100)#tag eth 4/3 eth 4/5 eth 4/7
info- FDP is disabled on port 4/3
info- FDP is disabled on port 4/5
info- FDP is disabled on port 4/7
```

- When only CDP is enabled globally.

```
device(config-mpls-vpls-vpls1-vlan-100)#tag eth 4/3 eth 4/5 eth 4/7
info- CDP is disabled on port 4/3
info- CDP is disabled on port 4/5
info- CDP is disabled on port 4/7
```

- When both FDP/CDP is enabled globally.

```
device(config-mpls-vpls-vpls1-vlan-100)#tag eth 4/3 eth 4/5 eth 4/7
info- FDP/CDP is disabled on port 4/3
info- FDP/CDP is disabled on port 4/5
info- FDP/CDP is disabled on port 4/7
```

For example, when VPLS endpoint is deleted the info messages are displayed as below.

- When only FDP is enabled globally.

```
device(config-mpls-vpls-vpls1-vlan-100)#no tag eth 4/3 eth 4/5 eth 4/7
info- FDP is enabled on port 4/3
info- FDP is enabled on port 4/5
info- FDP is enabled on port 4/7
```

- When only CDP is enabled globally.

```
device(config-mpls-vpls-vpls1-vlan-100)#no tag eth 4/3 eth 4/5 eth 4/7
info- CDP is enabled on port 4/3
info- CDP is enabled on port 4/5
info- CDP is enabled on port 4/7
```

- When both FDP/CDP is enabled globally.

```
device(config-mpls-vpls-vpls1-vlan-100)#no tag eth 4/3 eth 4/5 eth 4/7
info- FDP/CDP is enabled on port 4/3
info- FDP/CDP is enabled on port 4/5
info- FDP/CDP is enabled on port 4/7
```

> **NOTE**
> Configuring VPLS endpoint over FDP/CDP enabled interface will implicitly disable FDP/CDP on that specific interface and **show run** do not display any info about FDP/CDP configuration for that interface. Deleting the VPLS endpoint retains the prior FDP/CDP configuration on that specific interface and **show run** command now displays the FDP/CDP information again for that specific interface.

Uni-directional Link Detection packets

Uni-directional Link Detection (UDLD) cannot be configured on a VPLS endpoint port and a VPLS endpoint cannot be configured on a physical port that has UDLD enabled. This restriction is enforced by the CLI. When a VPLS endpoint receives any UDLD traffic, this traffic is dropped by the router at ingress. However, when the VPLS has CPU protection enabled, this traffic is intermittently hardware-flooded.

Flooding Layer 2 BPDUs with a VPLS instance

By default, Layer 2 *Spanning Tree Protocol (STP)* and *Per VLAN Spanning Tree (PVST)* BPDUs entering a VPLS endpoint are not transparently flooded within the VPLS instance. The BPDUs are dropped when they enter the VPLS endpoint. The user can change this default behavior to not block BPDUs and transparently flood them within the VPLS instance, by configuration on a per-physical-port basis.

To flood BPDUs with a VPLS, use the following command.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# no vpls-bpdu-block
```

Syntax: **[no] vpls-bpdu-block**

Specifying a VPLS MTU

The **vpls-mtu** command allows the user to specify an MTU value per VPLS instance. The newly configured VPLS MTU takes effect immediately to refresh or re-establish the VPLS sessions with peers in the following manner:

- When the VPLS session is Operational and the VPLSs MTU is changed by configuration, bring down the peer, send a label withdraw message to the peer, followed by the current VC binding message.

- When the VPLS session is not Operational and the VPLSs MTU is changed by configuration and the current state of the peer is VC-Parameter-MTU-Mismatch, the peer is brought UP when the MTU is equal, and a VC withdraw message is sent to clean up the old binding on the peer, and then a VC binding message is sent with the newly configured MTU. When the current state of the peer is anything other than VC-Parameter-MTU-Mismatch, the VPLSs configured MTU is changed.

- When a VC binding is received from a peer and VPLS MTU enforcement is enabled, the received MTU is compared with the VPLS's MTU. When they are not equal, the peer is kept in the VC-Parameter-MTU-Mismatch state, and otherwise made Operational. When the MTU enforcement is disabled, the peer's MTU is saved and the peer is made Operational irrespective of the MTU values.

To configure a new MTU value for a VPLS instance, use the **vpls-mtu** command as shown in the following example.

```
device(config-mpls)# vpls myvpls 40000
device(config-mpls-vpls-myvpls)# vpls-mtu 1000
```

Syntax: **[no] vpls-mtu** *mtu-value*

The *mtu-value* variable can be set to any value between 64 through 9190.

When the user employs the [**no**] parameter to remove the configured MTU value for a VPLS instance, that instance's MTU becomes one of two possible values:

- When a global default MTU has been configured, then the MTU for this VPLS instance becomes that global maximum frame size minus 26.

- When no global default exists, the MTU is 1500.

For example, when the user removes the MTU value for the VPLS named myvpls, and the global default maximum frame size is 5000, then the MTU for VPLS myvpls becomes 4974 (5000 - 26 = 4974).

> **NOTE**
> This MTU parameter is not enforced on the data plane (hardware). Consequently, packets larger than the configured MTU can still be sent or received.

Configuring VPLS MTU enforcement

The user can set the device to enforce the VPLS MTU value when establishing control sessions with peers. This is done globally on the Extreme device using the **vpls-mtu-enforcement** command.

```
device(config)# router mpls
device(config-mpls)# vpls-mtu-enforcement
```

Syntax: [no] vpls-mtu-enforcement

> **NOTE**
> The **vpls-mtu-enforcement** command is global to all VPLS instances. It requires a reload to take
> effect.

Configuring VPLS local switching

VPLS local switching is enabled by default, so packets received on a VPLS endpoint are flooded or forwarded to other VPLS endpoints belonging to the VPLS instance. This mode of operation does not require any configuration.

Using the **no vpls-local-switching** command, the user can disable VPLS local switching. With VPLS local switching disabled, packets are only flooded to the VPLS peers in a VPLS instance and not to the other VPLS endpoints belonging to that instance. Also, unicast traffic is discarded when it is received on a VPLS endpoint and is meant to go out on another VPLS endpoint.

The user can disable VPLS local switching behavior on a per-VPLS basis using the **no vpls-local-switching** command.

```
device(config)# router mpls
device(config-mpls)# vpls test 100
device(config-mpls-vpls-test)# no vpls-local-switching
```

Syntax: [no] vpls-local-switching

Once the **no vpls-local-switching** command has been used to disable VPLS local switching, the user can use the command without the [**no**] option to turn VPLS local switching on.

Special considerations

When using the VPLS local switching feature, consider the following:

- When the user toggles this option, all the MAC addresses that were learned on the VPLS endpoints are flushed and re-learned.
- This option does not affect IGMP and PIM snooping. Multicast traffic continues forwarding only to those VPLS endpoints and peers from which a Join for the (S, G) is requested, regardless of the status of the local switching option.
- IEEE 802.1ag packets follow the local switching option. In other words, packets are forwarded or flooded to other VPLS endpoints when local switching is enabled and discarded when local switching is disabled.

Enabling MPLS VPLS traps

The user can enable traps that are generated for MPLS VPLS by entering the following command.

```
device(config)# snmp-server enable trap mpls vpls
```

Syntax: [no] snmp-server enable trap mpls vpls

Refer to the *Unified IP MIB Reference* for MPLS VPLS trap notifications.

Disabling Syslog messages for MPLS VPLS

The generation of Syslog messages for MPLS VPLS and MPLS VLL Local is enabled by default. When the user wants to disable the logging of these events, enter the following command.

```
device(config)# no logging enable mpls
```

Syntax: [no] logging enable mpls

VPLS extended counters

With the support of ingress and egress port VLAN counters on the MLX Series series 8x10G module, the port VLAN counters are enabled by default for all the VPLS instances. As a result, the user can count the number of packets and bytes that are received and sent on a particular endpoint or all the endpoints of the VPLS instances. The user can also count per-priority statistics on each endpoint by enabling per-VLAN, port, and priority-based accounting mode on the ingress and egress counters at the global configuration level.

> NOTE
> The extended counters for dual tag endpoints are not supported both on the ingress and egress ports.

To disable the extended counters globally for all the VPLS instances, enter the following command:

```
device(config-mpls)# vpls-policy
device(config-mpls-vpls-policy)# no extended-counters
```

Syntax: [no] extended-counters

When the extended counters are disabled globally, the user can enable the extended counters for a particular VPLS instance by entering the following command:

```
device(config-mpls-vpls-test10)# extended-counters on
```

Syntax: [no] extended-counters [on | off]

The **on** option enables extended counters for a particular VPLS instance. The **off** option disables extended counters for a particular VPLS instance.

Displaying VPLS extended counters

When extended counters are enabled for a particular VPLS instance either by default or explicit configuration, the user can display the number of bytes and packets received and sent on a particular endpoint or all the endpoints of that particular VPLS instance. The counters are displayed whether or not the per-VLAN, port, and priority-based accounting mode is enabled at the global configuration level.

When the per-VLAN, port, and priority-based accounting mode is enabled at the global configuration level, the following output is displayed for the **show mpls statistics vpls extended-counters** command.

```
device# show mpls statistics vpls 10 extended-counters vlan 10 ethernet 3/2
VPLS Extended Counters (only applicable for G2 modules):
VPLS Name: test, VPLS  Id: 10
VPLS Vlan: vlan 10
Interface   RxPkts     TxPkts     RxBytes       TxBytes
eth 3/2     4841670    4841670    2595135120    2595135120
p0          4841670    4841670    2595135120    2595135120
p1          0          0          0             0
p2          0          0          0             0
p3          0          0          0             0
```

```
p4          0          0          0          0
p5          0          0          0          0
p6          0          0          0          0
p7          0          0          0          0
```

When the per-VLAN, port, and priority-based accounting mode is disabled, the following output is displayed for the **show mpls statistics vpls extended-counters** command.

```
device# show mpls statistics vpls 10 extended-counters vlan 10 ethernet 3/2
VPLS Extended Counters (only applicable for G2 modules):
VPLS Name: test, VPLS  Id: 10
VPLS Vlan: vlan 10
Interface   RxPkts       TxPkts      RxBytes       TxBytes
eth 3/2     4841670      4841670     2595135120    2595135120
```

Syntax: **show mpls statistics vpls** [*vpls-name* | *vpls-id* [**extended-counters** [[**vlan** *vlan-id*] [**ethernet** *port-id*]]]]

The *vpls-name* parameter specifies the configured name for a VPLS instance.

The *vpls-id* parameter specifies the ID of a VPLS instance.

The **extended-counters** keyword enables the extended counters for a particular VPLS instance.

The **vlan** *vlan-id* parameter specifies the ID of the configured VLAN.

The **ethernet** *port-id* parameter specifies the port ID of the interface for which the user wants to display the counters.

Table 24 describes the output parameters of the **show mpls statistics vpls extended-counters** command.

TABLE 24 Output of the show mpls statistics **vpls extended-counters** command

Output field	Description
VPLS Name	The configured name for a VPLS instance.
VPLS Id	The ID of the VPLS instance.
VPLS Vlan	The ID of the configured VLAN.
Interface	The port ID of the interface for which the user wants to display the counters.
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 VPLS extended counters

To clear all the port VLAN counters for a particular VPLS instance, enter the following command. The command does not clear the existing 'Endpt-Out-Pkts' and 'Tnl-Out-Pkts' statistics.

```
device# clear mpls statistics vpls 10 extended-counters
```

To clear all the port VLAN counters for a particular VPLS instance and port under a specific VPLS VLAN, enter the following command. This command is supported only for a single VLAN instance and is not supported for the dual tag endpoints.

```
device# clear mpls statistics vpls 10 extended-counters vlan 10
```

To clear all the port VLAN counters for all the endpoints of a particular VPLS instance, enter the following command. When the VPLS endpoint is a *Link Aggregation Group (LAG)*, then the counters only for the given physical port are cleared.

```
device# clear mpls statistics vpls 10 extended-counters vlan 10 ethernet 3/2
```

To clear all the port VLAN counters for the given priority of a particular VPLS endpoint, enter the following command:

```
device# clear mpls statistics vpls 10 extended-counters vlan 10 ethernet 3/2 p1
```

Syntax: **clear mpls statistics vpls** [*vpls-name* | *vpls-id* [**extended-counters** [[**vlan** *vlan-id*] [**ethernet** *port-id* [**priority** *pri*]]]]]

The *vpls-name* parameter specifies the configured VPLS name for which the user wants to clear the counters.

The *vpls-id* parameter specifies the ID of a VPLS instance for which the user wants to clear the counters.

The **vlan** *vlan-id* parameter specifies the ID of the configured VLAN for which the user wants to clear the counters.

The **ethernet** *port-id* parameter specifies the port ID of the interface for which the user wants to clear the counters.

The **priority** *pri* parameter specifies a priority queue for a particular VPLS endpoint for which the user wants to clear the counters.

Local VPLS

Local VPLS is used to create a VPLS circuit with endpoints in the same device. A Local VPLS can be configured between two or more ports in a router, two or more LAGs in a router, or between a port and a LAG as shown in Figure 59. Each entity (port or LAG) is identified as "Endpoint 1", Endpoint 2" or "Endpoint 3".

> NOTE
> Trunks supported include server LAGs, per-packet server LAGs, and Link Aggregation Control Protocol (LACP) LAGs.

FIGURE 59 Local VPLS port and LAG configurations
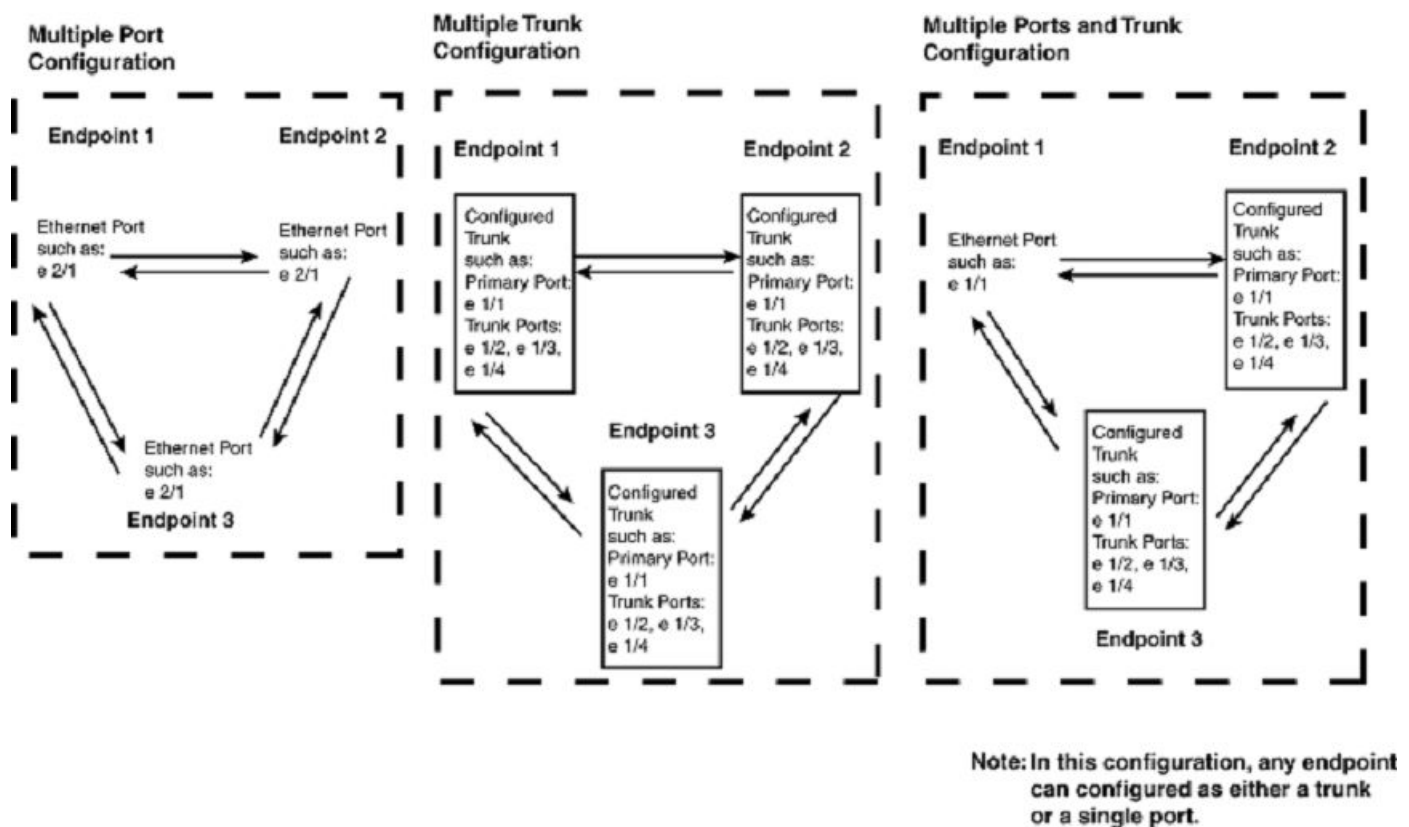
NOTE

When configuring a LAG as an endpoint, only the primary port of the LAG is specified in the Local VPLS configuration.

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.

The endpoints connected to the Local VPLS can be untagged, dual tagged, or single-tagged as members of the same or different VLANs. Using this function of Local VPLS, a router can receive packets with particular tags or no tag on one endpoint and forward them to the Local VPLSs other endpoint, which may be untagged, dual-tagged, or single-tagged with a different VLAN tag. When so configured, the tags within the packets are changed to reflect the configuration of the egress port as they leave the router.

Example Local VPLS configuration

In Figure 60 the Local VPLS named "Test1" contains Ethernet ports 1/1, 2/1, and 3/1. Port 1/1 is a member of VLAN 100, port 2/1 is a member of VLAN 200, and port 3/1 is a member of VLAN 300. Because all of the ports belong to Local VPLS "Test1", traffic tagged with any of the configured tags (100, 200, or 300) can reach traffic within any of the three VLANs. For example, traffic that ingresses on port 1/1 must have a tag with the value "100" and egresses on port 2/1 with a tag value of "200" or egress on port 3/1 with a tag value of "300".

FIGURE 60 Local VPLS "Test1" with three tagged VLANs
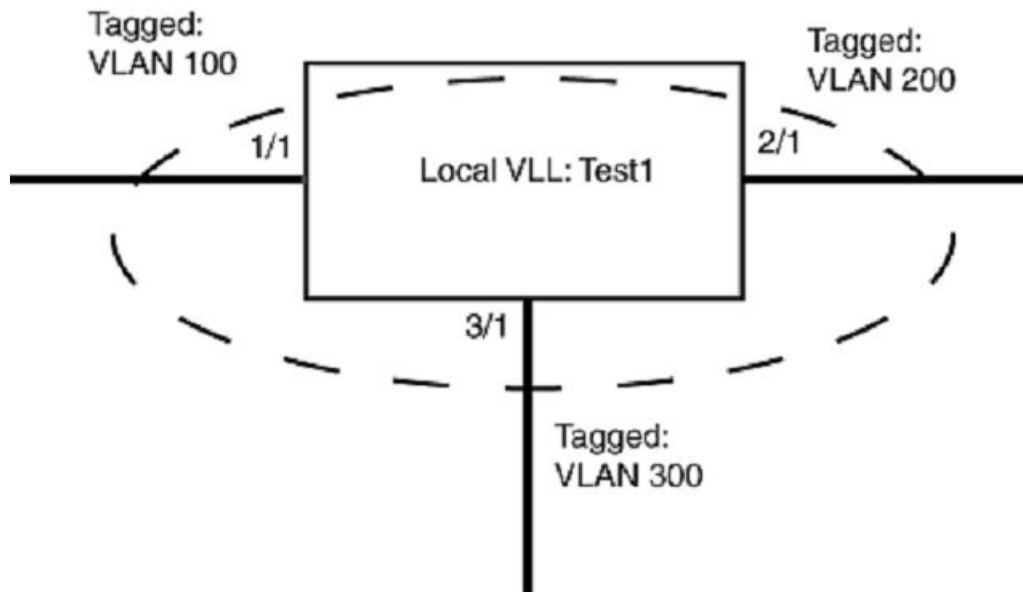

```
device(config)# router mpls
device(config-mpls)# vpls test1 5000
device(config-mpls-vpls-test1)# vlan 100
device(config-mpls-vpls-test1-vlan-100)# tagged ethernet 1/1
device(config-mpls-vpls-test1-vlan-100)# vlan 200
device(config-mpls-vpls-test1-vlan-200)# tagged ethernet 2/1
device(config-mpls-vpls-test1-vlan-200)# vlan 300
device(config-mpls-vpls-test1-vlan-300)# tagged ethernet 3/1
```

CoS behavior for Local VPLS

NOTE
This section assumes that the user understands how QoS works.

Table 25 describes the expected *Class of Service (CoS)* behavior for VPLS packets when Local VPLS is in effect.

TABLE 25 Expected class of service behavior for Local VPLS

Local VPLS endpoints	Incoming packet		Outgoing packet	
Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN	
Dual-tagged to dual-tagged	X	Y	X or X	Y
Single-tagged to dual-tagged	X	N/A	X or X	X
Untagged to dual-tagged	N/A	N/A	X or 0	0
Dual-tagged to single-tagged	X	Y	X or Y	N/A

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

X = Mapped CoS from internal priority (X contributes to internal priority) using CoS encode table.

Legend for Table 71

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

X = Mapped CoS from internal priority (X contributes to internal priority) using CoS encode table.

Specifying Local VPLS endpoints

Local VPLS can be configured between any combination of dual-tagged, single-tagged, and untagged endpoints.

The following procedures describe how to configure VPLS endpoints:

- Configuring an untagged endpoint on page 310
- Configuring a single-tagged endpoint on page 311
- Configuring a dual-tagged endpoint on page 311

Configuring an untagged endpoint

To configure untagged port 1/1 into Local VPLS instance "test1", use the following commands.

```
device(config)# router mpls
device(config-mpls)# vpls test1 5000
device(config-mpls)# vlan 100
device(config-mpls-vpls-test1)# untagged ethernet 1/1
```

Syntax: [no] untagged ethernet *slot /port /vpls-id*

The *vpls-id* variable is the ID of a VPLS instance.

Configuring a single-tagged endpoint

Tagged ports are configured under a VLAN ID. This VLAN ID is only meaningful for the tagged port.

For tagged ports, a *vlan-id* , *port* variable pair constitutes a VPLS endpoint. When a port is currently a member of a non-default VLAN as an untagged port, it must be returned to the default VLAN before it can be assigned to a VPLS as a tagged port.

To configure a tagged port 1/2 with VLAN 200 into Local VPLS instance "test1", use the following commands:

```
device(config)# router mpls
device(config-mpls)# vpls test1
device(config-mpls-vpls-test1)# vlan 200
device(config-mpls-vpls-test1-vlan-200)# tagged ethernet 1/2
```

Syntax: **vlan** *VLAN-ID*

The range for *VLAN-ID* from 1 through 4094. (This parameter range excludes the default VLAN ID.)

Syntax: [**no**] **tagged ethernet** *slot/port*

The *slot/port* variable specifies the port that is a tagged ethernet port.

Configuring a dual-tagged endpoint

A dual-tagged endpoint enables packets to have both an outer VLAN tag and an inner VLAN tag. In this configuration, an endpoint can receive packets with two tags and forward them to the other endpoint either untagged, single-tagged, or dual-tagged.

> **NOTE**
> Dual-tagged endpoints for Local VPLS follow the same configuration rules as do endpoints of a VPLS instance. Before configuring a dual-tagged endpoint, see Special considerations for dual-tagged endpoints on page 292.

To configure a dual-tagged endpoint for Local VPLS, use the following commands:

```
device(config)# router mpls
device(config-mpls)# vpls test1
device(config-mpls-vpls-test1)# vlan 200 inner-vlan 300
device(config-mpls-vpls-test1-vlan-200)# tagged ethernet 1/2
```

Syntax: [**no**] **vlan** *VLAN-ID* **inner-vlan** *VLAN-ID*

Syntax: [**no**] **tagged ethernet** *slot/port*

The **vlan** *VLAN-ID* variable, which is the outer VLAN ID, can be in the range from 1 through 4094 and excludes the default VLAN ID.

The **inner-vlan** *VLAN-ID* variable can be in the range from 1 through 4095 and includes the default VLAN ID.

Use the [**no**] form of the command to remove the dual-tagged VPLS VLAN configuration and its associated endpoints. For example, the command **no vlan** *200* **inner-vlan** *300* removes the dual-tagged VLAN and associated endpoints. The single-tagged VLAN, **vlan 200**, is not deleted. Similarly, the command **no vlan** *200* removes the single-tagged VLAN, **vlan 200**, and associated endpoints. The dual-tagged VLAN, **vlan 200 inner-vlan 300**, is not deleted.

Displaying VPLS information

The user can display the following information about the VPLS configuration on the device:

- VPLS summary information
- Information about individual VPLS instances configured on the device
- Detailed information about VPLS instances

- Information about a specified VPLS ID or VPLS name
- Information about VPLS instances that are not fully operational
- The contents of the VPLS MAC database for a VPLS instance
- The VPLS MAC database entries on the *Management Processor (MP)*
- VPLS traffic statistics
- VPLS CPU protection configuration status

Display considerations for VPLS information

The VPLS information that is displayed in the output of the **show mpls vpls** commands has changed. Previously, when a VPLS was created, a range of VC labels was allocated to the VPLS instance. Now, there is no pre-allocation of VC label ranges to a VPLS instance.

The range of the allocated VC labels is no longer displayed in the output of the following **show mpls vpls** commands. Refer to the subsequent sections for more information on changes to the **show mpls vpls** command outputs:

- **show mpls vpls brief** – Displaying information about VPLS instances on page 312
- **show mpls vpls detail** – Displaying detailed information about VPLS instances on page 313
- **show mpls vpls down** – Displaying information about VPLS instances that are not operational on page 319
- **show mpls vpls id** – Displaying information about a specified VPLS ID or VPLS name on page 316
- **show mpls vpls summary** – Displaying VPLS summary information on page 312

Displaying VPLS summary information

The VC label allocation range size field is no longer displayed in the output of the **show mpls vpls summary** command.

You can display a summary of VPLS information, including the number of VPLS instances, the number of VPLS peers, the maximum size of the VPLS MAC database, VPLS raw mode, and the values of the VPLS global MTU, and the value of the remote VC MTU.

```
device# show mpls vpls summary
Virtual Private LAN Service summary:
  Total VPLS configured: 4072, maximum number of VPLS allowed: 4096
  Total number of IFL-ID's allocated by VPLS: 0
  Total VPLS peers configured: 8139, total peers operational: 8138
  Total VPLS Local end-points configured: 0
  Maximum VPLS mac entries allowed: 160000, currently installed: 150530
  VPLS global raw mode VC-Type is Ethernet (0x05)
  VPLS global MTU is 8974, MTU enforcement is OFF
  Global CPU protection: OFF
  VPLS policy parameters:
    vpls-pw-redundancy:  1
  MVIDs in use: 0 of 1 total allocated
  mac-address withdrawal-limit:  500
  MAC age time for local: 300
  MAC age time for remote: 600
```

Syntax: **show mpls vpls summary**

Displaying information about VPLS instances

The **show mpls vpls brief** command has changed. The Num VC-label field is no longer displayed in the output of the **show mpls vpls brief** command.

To display information about VPLS instances configured on the device, enter the following command.

```
device #show mpls vpls brief
                      Num   Num    Ports  Num    Peers         CPU  VC
Name        Id        Vlans Ports  Up     Peers  Up     IFL-ID Prot Mode
====        ==        ===== =====  =====  =====  ===== ====== ==== ======
1           1         2     2      2      1      1      4096   OFF  TAGGED
2           2         1     0      0      1      0      n/a    OFF  RAW
3           3         2     6      4      2      1      n/a    OFF  RAW
```

Syntax: show mpls vpls brief

Table 26 lists the output displayed by the **show mpls vpls brief** command.

TABLE 26 Output from the show mpls vpls **brief** command

Output field	Description
Name	The configured name of the VPLS instance.
Id	The ID of this VPLS instance.
Num Vlans	The total number of single-tagged and dual-tagged VLANs associated with this VPLS instance.
Num Ports	The number of ports in 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.
IFL-ID	The Internal Forwarding Lookup Identifier (IFL-ID) for dual-tagged VLAN ports in this VPLS instance.
CPU Prot	Whether CPU protection configured on this VPLS instance is ON or OFF.
VC Mode	The VC mode for the VPLS instance: • Raw - The VLAN tag information in the original payload is not carried across the MPLS cloud. • Tagged - The VLAN tag information in the original payload is carried across the MPLS cloud.

Displaying detailed information about VPLS instances

The **show mpls vpls detail** command has changed. The total VC labels allocated field is no longer displayed in the output of the **show mpls vpls detail** command.

To display more detailed information about each VPLS instance, enter a command similar to the following:

```
device# show mpls vpls detail
VPLS 3, Id 3, Max mac entries: 8192
 Total vlans: 2, Tagged ports: 2 (1 Up), Untagged ports 0 (0 Up)
 IFL-ID: n/a
  Vlan 500
    Tagged: ethe 1/3
  Vlan 600
    Tagged: ethe 1/4
 VC-Mode: Raw
 Total VPLS peers: 1 (1 Operational)
 Peer address: 21.21.21.21, State: Operational, Uptime: 1 min
  Tnnl in use: tnl0(3)
  LDP session: Up, Local VC lbl: 983040, Remote VC lbl: 983040
  Local VC MTU: 1500, Remote VC MTU: 9174
  Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
 CPU-Protection: OFF [Resource FID Failure, Retry in 18 seconds (approximate)]
```

```
        Local Switching: Enabled
           Extended Counter: ON
```

Multicast Snooping: Disabled

Syntax: show mpls vpls detail

The information related to the status of extended counters is shown in bold text in the previous output.

Table 27 lists the output displayed by the **show mpls vpls detail** command.

TABLE 27 Output from the show mpls vpls **detail** command

Output field	Description
VPLS	The configured name of the VPLS instance.
Id	The ID of this VPLS instance.
Max mac entries	The maximum number of MAC address entries that can be learned for this VPLS instance. This is a soft limit only and can be exceeded when there is space available in the VPLS MAC database.
Total vlans	The number of VLANs that are translated for this VPLS instance.
Tagged ports	The total number of tagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
Untagged ports	The total number of untagged ports that are associated with VLANs in this VPLS instance, as well as the number of these ports that are up.
IFL-ID	*The Internal Forwarding Lookup Identifier (IFL-ID) for dual-tagged ports in the VPLS instance.*
Vlan	The ID of each VLAN in this VPLS instance.
Tagged	The numbers of the tagged ports in each VLAN.
Untagged	The numbers of the untagged ports in each VLAN.
VC-Mode	The VC mode for the VPLS instance: • Raw – The VLAN tag information in the original payload is not carried across the MPLS cloud. • Tagged – The VLAN tag information in the original payload is carried across the MPLS cloud.
Total VPLS peers	The number of VPLS peers this device has for this VPLS instance, as well as the number of these VPLS peers with which this device has an LDP session.
Peer address	The IP address of the VPLS peer.
State	The current state of the connection with the VPLS peer. This can be one of the following states: • **Operational** – The VPLS instance is operational. Packets can flow between the device and the peer • **Wait for functional local ports** – The physical endpoint port that must be connected to the Customer Edge device is down due to a link outage or is administratively disabled • **Wait for LSP tunnel to Peer** – The device cannot find a working tunnel LSP • **Wait for PW Up (Wait for LDP session to Peer)** – The LDP session is not yet ready • **Wait for PW Up (Wait for remote VC label)** – The device has advertised its VC label binding to the VPLS peer, but has not yet received the peer's VC label binding • **Wait for PW Up (VC type mismatched)** – A session is not formed because the VC type does not match with its peer's VC type

TABLE 27 Output from the show mpls vpls **detail** command (continued)

Output field	Description
	• **Wait for PW Up (MTU mismatched)** - The MTU sent to a peer is derived from the device's global setting by the following formula: (system-mtu minus 26 bytes). When a system-mtu value is not configured, a default value of 1500 is sent. • **Wait for PW Up (Wait for LPD session to Peer)** - The LDP session to the peer is down • **Wait for PW Up (No Label Resource)** - When configuring a new VPLS peer, the maximum amount of VC labels that can be supported may exceed 64K, and cause the configuration to be rejected. The maximum amount of VC labels available for VPLS instances is equal to 64K.
Uptime	The time in minutes that the entry has been operational.
Tnnl in use	The tunnel LSP used to reach the VPLS peer. When VPLS traffic to the peer is load balanced across multiple tunnel LSPs, the tunnel LSPs used to reach the peer are displayed.
LDP session	The state of the LDP session between this device and the VPLS peer.
Local VC lbl	The VC label value locally allocated for this peer for this VPLS instance. Packets forwarded from the VPLS peer to this device are expected to contain this label. This is the label that is advertised to the VPLS peer through LDP.
Remote VC lbl	The VC label allocated by the VPLS peer and advertised to this device through LDP. The device applies this label to outbound MPLS packets sent to the VPLS peer.
Local VC MTU	The MTU value locally configured for this peer.
Remote VC MTU	The MTU value configured for the remote VPLS peer.
Local VC-Type	The VC type for this peer.
Remote VC-Type	The VC type for the remote VPLS peer.
CPU-Protection	Whether CPU protection configured on this VPLS instance is ON or OFF. On XMR Series and MLX Series devices only: When CPU protection is enabled on this VPLS instance but is temporarily unavailable due to 100% multicast FID usage, this field includes the message shown above.
Local Switching	Whether local switching behavior on a per-VPLS basis is enabled or disabled.
Extended Counter	Indicates whether or not the extended counter is enabled for the configured VPLS.
Multicast Snooping	Indicates whether the multicast snooping is enabled or disabled.

The Wait for LDP session to Peer state is no longer displayed in the output of the **show mpls vpls detail** command. The Wait for *Pseudo Wire (PW)*, Up (Wait for LDP session to Peer) state is now displayed, and replaces the existing state. The total VC labels allocated field is also removed from the output. In the following example, the LDP session to the remote peer is down. The Local VC 'lbl' field displays N/A (not applicable).

```
device# show mpls vpls detail
VPLS NO_LDP, Id 500, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4101
  Vlan 880 inner-vlan 35
   Tagged: ethe 8/2
 VC-Mode: Tagged
```

```
 Total VPLS peers: 1 (0 Operational)
Peer address: 66.66.66.66, State: Wait for PW Up (Wait for LDP session to Peer)
 Tnnl in use: tnl5(6)
 LDP session: Down, Local VC lbl: N/A
, Remote VC lbl: N/A
 Local VC MTU: 1500, Remote VC MTU: 0,
 LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
CPU-Protection: OFF
 Local Switching: Enabled
Extended Counter: ON
```

The maximum number of VC labels available for VPLS instances is equal to 64K. When configuring a new VPLS peer, the total number of VPLS peers exceeds 64K, and causes the configuration to be rejected. The following error message is displayed on the console.

```
device(config-mpls-vpls-1)# vpls-peer 10.23.23.23
Error - Unable to create vpls peer 10.23.23.23 for VPLS 1 due to no VC label resource.
```

The Wait for PW Up (No label Resource) state is introduced in the output of the **show mpls vpls detail** command. In the following example, the Wait for PW Up (No label Resource) state is highlighted.

```
device# show mpls vpls detail
VPLS waiting_for_remote_label, Id 400, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4100
  Vlan 900 inner-vlan 245
   Tagged: ethe 7/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 10.55.55.55, State: Wait for PW Up (No Label Resource)
 Tnnl in use: tnl4(5)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
Extended Counter: ON
```

When the system runs out of memory, a warning message is displayed on the console. To recover from this state, the user is required to delete the failed peer and reconfigure it. VPLS generates the following warning messages.

```
WARNING: VPLS id 3 Peer IP Address: 10.21.21.21 is placed in VC Bind Failure state due to low system memory.
WARNING: VPLS id 3 Peer IP Address: 10.11.11.11  is placed in VC Withdraw Failure state due to low system
memory.
```

Displaying information about a specified VPLS ID or VPLS name

The **show mpls vpls id** *vpls-id* command displays detailed information about a specified VPLS ID. The **show mpls vpls name** *vpls-name* command displays detailed information about a VPLS name. The output of the **show mpls vpls id** *vpls-id* command, and the output of the **show mpls vpls name** *vpls-name* command display the same information for a configured VPLS instance. The display changes that are described below are applicable to both the **show mpls vpls id** *vpls-id* command, and **show mpls vpls name** *vpls-name* command.

When the remote peer is in an operational state, the total VC labels allocated field no longer displays in the output of the **show mpls vpls id** *vpls-id* command, as shown in the following example.

```
device# show mpls vpls id 3
VPLS name_raw, Id 3, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 3 (3 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4097
  Vlan 300 inner-vlan 500
   Tagged: ethe 3/1 ethe 3/11 ethe 3/13
 VC-Mode: Raw
 Total VPLS peers: 1 (1 Operational)
 Peer address: 10.200.200.200, State: Operational
, Uptime: 1 hr 10 min
```

```
  Tnnl in 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
```

When a VC type mismatch occurs, the output from the **show mpls vpls id** *vpls-id* command displays the Wait for PW Up (VC type mismatched) state. The Wait for VC parameter check (VC type mismatched) state no longer displays The total VC labels allocated field is also removed from the output. In the following example, a VC type mismatch has occurred, and the PW is down. The local VC MTU and the remote VC MTU are not known by VPLS so there is no information to display. The Local VC 'lbl' field, the Remote VC 'lbl' field, and the Remote VC MTU field display N/A (non applicable).

```
device# show mpls vpls id 200
VPLS vc_mismatched, Id 200, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4098
Vlan200 inner-vlan145
Tagged: ethe2/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 10.33.33.33, State: Wait for PW Up (VC type mismatched)
Tnnlin use: tnl0(2)
LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: N/A
LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: Ethernet (0x05)
```

When a MTU mismatch occurs, the output from the **show mpls vpls id** *vpls-id* command displays the Wait for PW Up (MTU mismatched) state. The Wait for VC parameter check (MTU mismatched) state no longer displays. The total VC labels allocated field is also removed from the output. In the following example, a MTU mismatch has occurred, and the PW is down. The Local VC 'lbl' field and the Remote VC lbl field display N/A (not applicable).

> **NOTE**
> When both the VC type and MTU are mismatched, only the output from the VC type mismatch is displayed on the console.

```
device# show mpls vpls id 300
VPLS mtu_mismatched, Id 300, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4099
Vlan100 inner-vlan145
Tagged: ethe1/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 10.44.44.44, State: Wait for PW Up (MTU mismatched)
Tnnlin use: tnl3(3)
LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: 2500,
LOCAL VC-Type: Ethernet Tagged (0x04), Ethernet Tagged (0x04)
```

The Wait for remote VC label from Peer state no longer displays in the output of the **show mpls vpls id** *vpls-id* command. The Wait for PW Up (Wait for remote VC label) state now displays, and replaces the existing state. The total VC labels allocated field is also removed from the output. In the following example, the PW is down and it is waiting for the VC label of the remote peer to advertise to the VPLS peer. The Local VC 'lbl' field and the Remote VC MTU field displays N/A (non applicable).

```
device# show mpls vpls id 400
VPLS waiting_for_remote_label, Id 400, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4100
Vlan900 inner-vlan245
Tagged: ethe7/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 10.55.55.55, State: Wait for PW Up (Wait for remote VC label)
Tnnlin use: tnl4(5)
LDP session: Up, Local VC lbl: N/A
```

```
, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: N/A
,
LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
...
```

The Wait for PW Up (VC Bind in Progress) state is introduced in the output of the **show mpls vpls id** *vpls-id* command. The total VC labels allocated field is removed from the output. In the following example, the PW is down, and local VC binding is still in progress. The Local VC 'lbl' field and the Remote VC MTU field display N/A (non applicable).

```
device# show mpls vpls id 400
VPLS waiting_for_remote_label, Id 400, Max macentries: 8192
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
IFL-ID: 4100
Vlan900 inner-vlan245
Tagged: ethe7/1
VC-Mode: Tagged
Total VPLS peers: 1 (0 Operational)
Peer address: 10.55.55.55, State: Wait for PW Up (VC Bind in Progress)
Tnnlin use: tnl4(5)
LDP session: Up, Local VC lbl: N/A
, Remote VC lbl: N/A
Local VC MTU: 1500, Remote VC MTU: N/A
,
LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
```

The **show mpls vpls id** *vpls-id* command displays the tunnel LSPs that are being used to forward VPLS traffic from the device to the peer. When VPLS traffic to a peer is being load balanced across multiple tunnel LSPs, then the command lists the tunnel LSPs used for load balancing, as shown in the example below.

```
device# show mpls vpls id 5
VPLS test5, Id 5, Max mac entries: 2048
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  Vlan 50
    Tagged: ethe 5/3
 VC-Mode: Raw
 Total VPLS peers: 1 (1 Operational)
 Peer address: 10.5.5.5, State: Operational, Uptime: 28 min
  Tnnl (load balance): tnl0(3) tnl2(3) tnl1(3)
  LDP session: Up, Local VC lbl: 983040, Remote VC lbl: 983040
```

Syntax: show mpls vpls id *vpls-id*

Syntax: show mpls vpls name *vpls-name*

The *vpls-id* variable is the ID of a VPLS instance. The *vpls-name* variable is the name of a VPLS instance.

Displaying VPLS CPU protection configuration status

The **show mpls vpls id** command has changed. The total VC labels allocated field is no longer displayed in the output of the **show mpls vpls id** command.

To see the VPLS CPU protection configuration status for a specified VPLS, use the **show mpls vpls id** command.

```
device(config)# show mpls vpls id 1
VPLS test1, Id 1, Max mac entries: 2048
 Total vlans: 1, Tagged ports: 1 (0 Up), Untagged ports 1 (1 Up)
  Vlan 2
    Tagged: ethe 5/4
    Untagged: ethe 2/2
 Total VPLS peers: 1 (0 Operational)
 Peer address: 10.1.1.1, State: Wait for remote VC label from Peer
  Tnnl: tnl0(3), LDP session: Up, Local VC lbl: 983040, Remote VC lbl: N/A
 CPU-Protection: ON
, MVID: 0x000,  VPLS FID: 0x00000205
```

The CPU protection status is highlighted in the previous example. It can be either on or off. When CPU protection status is enabled on the VPLS but is temporarily off due to unavailable FID resources, the following message is shown in the CPU-Protection field:

```
CPU-Protection: OFF [Resource FID Failure, Retry in 18 seconds (approximate)]
```

Syntax: **show mpls vpls id** [*vpls-id*]

The *vpls-id* variable is the ID of a VPLS instance.

Displaying information about VPLS instances that are not operational

The **show mpls vpls down** command has changed. The Num VC-label field no longer displays in the output of the **show mpls vpls down** command.

To display information about VPLS instances that are not fully operational, enter a command similar to the following:

```
device# show mpls vpls down
The following VPLS's are not completely operational:
                    Num      Num      Ports    Num      Peers
Name          Id    Vlans    Ports    Up       Peers    Up
test1         1     1        1        1        1        0
test2         2     1        2        1        1        0
test3         3     1        1        1        1        0
test4         4     1        2        1        1        0
```

Syntax: **show mpls vpls down**

Displaying the contents of the VPLS MAC database

The VPLS MAC database stores entries associating remote MAC addresses with VC LSPs and local MAC addresses with CE devices. When a PE device receives a Layer 2 frame from an attached CE device with a given destination MAC address, the PE device looks up the MAC address in the VPLS MAC database and assigns the frame to the associated VC LSP. Each VPLS instance configured on the PE device has a separate VPLS MAC database.

> **NOTE**
> It is possible in a loaded system that entries keep aging for few seconds. The age resets to zero (0) after some time and entries remain intact.

Displaying VPLS MAC database entries on the Management Processor

To display the entire VPLS MAC database on the *Management Processor (MP)*, enter the following command.

```
device# show mac vpls
Total VPLS mac entries in the table: 10 (Local: 5, Remote: 5)
                              Vlan:Inner-
VPLS MAC Address     L/R Port Vlan/Peer      Age
==== ============== === ==== =========     ===
1    0000.0000.1601 R   5/1  10.3.3.3        0
1    0000.0000.1003 L   5/3  2               0
1    0000.0000.1603 R   5/1  10.3.3.3        0
1    0000.0000.1005 L   5/3  2               0
1    0000.0000.1002 L   5/3  2               0
1    0000.0000.1605 R   5/1  10.3.3.3        0
1    0000.0000.1602 R   5/1  10.3.3.3        0
1    0000.0000.1004 L   5/3  2               0
1    0000.0000.1001 L   5/3  2               0
1    0000.0000.1604 R   5/1  10.3.3.3        0
1    0000.0001.0201 L   5/4  100:200         0
```

When a given remote VPLS MAC address is learned on multiple uplink interfaces, the Port field in the output of the **show mac vpls** command indicates "Mult." instead of a port number. For example, this abbreviation might appear when all of the following are true:

- The remote PE establishes multiple LSPs to this device
- Packets from a remote VPLS MAC address are load balanced across these LSPs
- The packets arrive on different MPLS uplink interfaces at this device

```
device# show mac vpls
Total VPLS mac entries in the table: 2274 (Local: 8, Remote: 2266)
                                Vlan:Inner-
VPLS   MAC Address      L/R Port  Vlan/Peer       Age
====   ==============   === ====  ============    ===
3      0000.009b.d419   L   4/2   3               0
504    0000.0000.0067   R   Mult. 10.99.42.253    0
504    0000.0033.b24c   R   Mult. 10.99.42.253    0
504    0000.0073.6185   R   1/1   10.99.42.253    375
504    0000.0000.40cf   R   Mult. 10.99.42.253    0
504    0000.0019.d7f4   R   Mult. 10.99.42.253    0
504    0000.0044.d58b   R   Mult. 10.99.42.253    0
504    0000.005c.5a3b   R   Mult. 10.99.42.253    0
504    0000.0044.d696   R   Mult. 10.99.42.253    0
```

To see details for all the ports on which a remote VPLS MAC address has been learned, use the **show mac mpls vpls** *mac-address* command.

To display the VPLS MAC database on the MP for a VPLS instance specified by its VPLS ID, enter the following command.

```
device# show mac vpls 1
Total MAC entries for VPLS 1: 10 (Local: 5, Remote: 5)
                              Vlan:Inner-
VPLS MAC Address      L/R Port Vlan/Peer     Age
==== ==============  === ====  =========     ===
1    0000.0000.1601 R   5/1   10.3.3.3       0
1    0000.0000.1003 L   5/3   2              0
1    0000.0000.1603 R   5/1   10.3.3.3       0
1    0000.0000.1005 L   5/3   2              0
1    0000.0000.1002 L   5/3   2              0
1    0000.0000.1605 R   5/1   10.3.3.3       0
1    0000.0000.1602 R   5/1   10.3.3.3       0
1    0000.0000.1004 L   5/3   2              0
1    0000.0000.1001 L   5/3   2              0
1    0000.0000.1604 R   5/1   10.3.3.3       0
1    0000.0001.0201 L   5/4   100:200        0
```

To display a specific entry in the VPLS MAC database on the MP, enter the following command.

```
device# show mac vpls 1 0000.0000.1601
    VPLS: 1          MAC:  0000.0000.1601    Age:    0
    Remote MAC      Port: ethe 5/1           Peer:   10.3.3.3
Trunk slot mask: 00000000
```

Syntax: show mac vpls [*vpls-id* [*mac-address*]]

The *vpls-id* variable is the ID of a VPLS instance. When the user specifies the VPLS ID, the user can also specify a particular entry in the VPLS MAC database by adding the optional *mac-address* variable.

Table 28 lists the output displayed by the **show mac vpls** command.

TABLE 28 Output from the show mac vpls command

Output field	Description
Total VPLS mac entries in the table	The number of MAC addresses that have been learned in the database.
Local	The number of locally learned entries in the database.
Remote	The number of remotely learned entries in the database.

TABLE 28 Output from the show mac vpls command (continued)

Output field	Description
VPLS	The VC ID of the VPLS instance.
MAC Address	The MAC address of the entry.
L/R	Whether the entry was learned from local endpoints (L), or was learned from a remote VPLS peer (R).
Port	The port number for the entry.
Vlan:Inner-VLAN/Peer	For Local entries, the VLAN ID for the port; for dual-tagged VLANs, the outer VLAN ID followed by the inner VLAN ID; for Remote entries, the IP address of the VPLS peer.
Age	The age of the entry. The value on the MP is zero because the aging occurs on line card processors.

> **NOTE**
> The information displayed in the SA-CAM and DA-CAM index fields is not relevant for day-to-day management of the device. The information is used by engineering and technical support staff for debugging purposes.

Displaying VPLS traffic statistics

The user can display VPLS traffic statistics, to view the forwarding counters for each VPLS configured on the system. The output shows a given port range that receives traffic, how many packets are sent out on local CE device endpoints, and how many are sent out of LSP tunnels to remote PE devices. When the port is a 10G port, a single port is displayed. When the module is a 40x1G module, a range of 10 1G ports is displayed.

> **NOTE**
> When CPU protection is on, flooded traffic received from an endpoint is not accounted by the VPLS statistics for endpoint-out packets even though they are locally switched.

To display all VPLS traffic statistics on an Extreme device, enter a command similar to the following:

```
device# show mpls statistics vpls
VPLS-Name    In-Port(s)     Endpt-Out-Pkts    Tnl-Out-Pkts
--------    ---------     --------------    ------------
test2       e1/1           0                 0
            e1/2           0                 0
            e1/3           0                 0
            e1/4           0                 0
test2       e2/1 - e2/10   0                 0
            e2/11 - e2/20  0                 0
            e2/21 - e2/30  0                 0
            e2/31 - e2/40  0                 0
test3       e1/1           0                 0
            e1/2           0                 0
            e1/3           0                 0
            e1/4           0                 0
test3       e2/1 - e2/10   0                 0
            e2/11 - e2/20  0                 0
            e2/21 - e2/30  0                 0
            e2/31 - e2/40  0                 0
test4       e1/1           0                 0
            e1/2           0                 0
            e1/3           0                 0
            e1/4           0                 0
test4       e2/1 - e2/10   0                 0
            e2/11 - e2/20  0                 0
            e2/21 - e2/30  0                 0
            e2/31 - e2/40  0                 0
test4       e5/1           10354120822       0
            e5/2           0                 0
```

```
           e5/3              0                    2992416134
           e5/4              0                    0
```

NOTE

The VPLS name is repeated for each module from which the statistics are collected, to be displayed on the MP console.

To display VPLS traffic statistics for a VPLS instance specified by its VPLS name, enter a command similar to the following:

```
device# show mpls statistics vpls test4
VPLS-Name       In-Port(s)    Endpt-Out-Pkts    Tnl-Out-Pkts
---------       ----------    --------------    ------------
test4           e1/1          0                 0
                e1/2          0                 0
                e1/3          0                 0
                e1/4          0                 0
test4           e2/1 - e2/10  0                 0
                e2/11 - e2/20 0                 0
                e2/21 - e2/30 0                 0
                e2/31 - e2/40 0                 0
test4           e5/1          10828448712       0
                e5/2          0                 0
                e5/3          0                 3025869251
                e5/4          0                 0
```

To display VPLS traffic statistics for a VPLS instance specified by its VPLS ID, enter a command similar to the following:

```
device# show mpls statistics vpls 4
VPLS-Name       In-Port(s)    Endpt-Out-Pkts    Tnl-Out-Pkts
---------       ----------    --------------    ------------
test4           e1/1          0                 0
                e1/2          0                 0
                e1/3          0                 0
                e1/4          0                 0
test4           e2/1 - e2/10  0                 0
                e2/11 - e2/20 0                 0
                e2/21 - e2/30 0                 0
                e2/31 - e2/40 0                 0
test4           e5/1          10828448712       0
                e5/2          0                 0
                e5/3          0                 3025869251
                e5/4          0                 0
```

Syntax: show mpls statistics vpls [*vpls-name* | *vpls-id*]

The *vpls-name* variable is the configured name for a VPLS instance.

The *vpls-id* variable is the ID of a VPLS instance.

Table 29 lists the output displayed by the **show mpls statistics vpls** command.

TABLE 29 Output from the show mpls statistics vpls command

Output field	Description
VPLS-Name	The configured name of the VPLS instance.
In-Port(s)	The port where the traffic is received.
Endpt-Out-Pkts	The number of packets transmitted out of local endpoints.
Tnl-Out-Pkts	The number of packets transmitted out of LSP tunnels.

Clearing VPLS traffic statistics

To clear the entries stored for all VPLS statistics, enter a command similar to the following:

```
device# clear mpls statistics vpls
```

Syntax: clear mpls statistics vpls [*vpls-name* | *vpls-id*]

The *vpls-name* variable is the configured name for a VPLS instance.

The *vpls-id* variable is the ID of a VPLS instance.

The support enables simplified interactions between MPLS and VPLS with regard to VPLS peer FSM transitions. The LDP integration is supported on all platforms.

VPLS LDP

Displaying the VPLS peer FSM state with LDP support

The user can display the various VPLS peer FSM states with the LDP integration on the device using the **show mpls vpls** commands.

TABLE 30 PEER FSM state description

Peer FSM state name	State description
Wait for functional local ports	No functional local endpoints.
Wait for LSP tunnel to Peer	No LSP tunnels available to reach the remote peer.
Wait for PW UP (Wait for LDP Session)	LDP session to remote peer is down.
Wait for PW UP (Wait for remote VC label)	PW is down (waiting for remote peer's VC label).
Wait for PW UP (VC type Mismatched)	PW is down (VC type mismatched).
Wait for PW UP (MTU Mismatched)	PW is down (MTU mismatched)
Wait for PW UP (VC Bind In Progress)	PW is down (Local VC binding in progress).
Operational	PW is up and operational.
Wait Withdraw Done …	Waiting for VC withdraw completion (internal intermediate states).
VC BIND Failure State	VC binding failed. User intervention required.
VC Withdraw Failure State	VC withdraw failed. User intervention required.

User intervention is required to recover from the VC Bind Failure state, and the VC Withdraw Failure state. To recover, the user must delete the failed peer and then add it back. These failure states may occur during extreme conditions when the system runs out of memory to issue ITC requests. When these failures are detected, VPLS generates the following syslog messages accordingly.

```
WARN: VPLS id X Peer IP Address: aa.bb.cc is placed in VC Bind Failure state due to low system memory.
WARN: VPLS id Y Peer IP Address: dd.ee.ff  is placed in VC Withdraw Failure state due to low system memory.
```

VC type mismatched

The following example shows the output for the LDP integration for a VC type mismatched case.

```
device# show mpls vpls id 200
VPLS vc_mismatched, Id 200, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4098
  Vlan 200 inner-vlan 145
   Tagged: ethe 2/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 10.33.33.33, State: Wait for PW Up (VC type mismatched)
  Tnnl in use: tnl0(2)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: N/A
```

```
    LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: Ethernet (0x05)
   CPU-Protection: OFF
   Local Switching: Enabled
```

The local VC label and remote VC label display is performed only when the Peer is in Operational state. Otherwise, it displays 'N/A' for these fields.

The remote VC Type is the same as the local VC type when the peer state is Operational, or else, it is shown as 'N/A'.

MTU mismatched

The following example shows the output for the LDP integration for a MTU mismatched case.

```
device# show mpls vpls id 300
VPLS mtu_mismatched, Id 300, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4099
  Vlan 100 inner-vlan 145
   Tagged: ethe 1/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 10.44.44.44, State: Wait for PW Up (MTU mismatched)
  Tnnl in use: tnl3(3)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 2500,
  LOCAL VC-Type: Ethernet Tagged (0x04), Ethernet Tagged (0x04)
 CPU-Protection: OFF
Local Switching: Enabled
```

No remote VC label

The following example shows the output for the LDP integration for a no remote VC label case.

```
device# show mpls vpls id 400
VPLS waiting_for_remote_label, Id 400, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4100
  Vlan 900 inner-vlan 245
   Tagged: ethe 7/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 10.55.55.55, State: Wait for PW Up (Wait for remote VC label)
  Tnnl in use: tnl4(5)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
  LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
```

LDP session down

The following example shows the output for the LDP integration for an LDP session down case.

```
device# show mpls vpls detail
VPLS NO_LDP, Id 500, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4101
  Vlan 880 inner-vlan 35
   Tagged: ethe 8/2
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 10.66.66.66, State: Wait for PW Up (Wait for LDP session to Peer)
  Tnnl in use: tnl5(6)
  LDP session: Down, Local VC lbl: N/A, Remote VC lbl: N/A
```

```
     Local VC MTU: 1500, Remote VC MTU: 0,
      LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
     CPU-Protection: OFF
      Local Switching: Enabled
    Extended Counter: ON
```

No local label resource

The following example shows the output for the LDP integration for a no local label resource case.

```
device# show mpls vpls detail
VPLS waiting_for_remote_label, Id 400, Max mac entries: 8192
 Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
  IFL-ID: 4100
  Vlan 900 inner-vlan 245
   Tagged: ethe 7/1
 VC-Mode: Tagged
 Total VPLS peers: 1 (0 Operational)
 Peer address: 10.55.55.55, State: Wait for PW Up (No Label Resource)
  Tnnl in use: tnl4(5)
  LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0,
   LOCAL VC-Type: Ethernet Tagged (0x04), Remote VC-Type: UNKNOWN
 CPU-Protection: OFF
 Local Switching: Enabled
Extended Counter: ON
```

MPLS LDP show commands

When there are issues with the peer VC labels (local or remote), MTU values, or VC type, other than using the **show vpls** command as documented in previous sections, the user can also use the **show mpls ldp** command to compare the PW VC information.

Using the show mpls ldp vc x command

Here is an example of the **show mpls ldp fec vc** command where the remote peer is in the Operational state.

```
device# show mpls ldp fec vc 1
FEC_CB: 0x34ff85a8, idx: 50, type: 128, pend_notif: None
 State: current, Ingr: Yes, Egr: Yes, UM Dist. done: Yes
 VC-Id: 1, vc-type: 4, grp-id: 0
 Local-mtu: 2000, remote-mtu: 1500, MTU enforcement: disabled
 Downstream mappings:
 Local LDP ID       Peer LDP ID       Label     State    CB
 10.21.21.21:0      10.11.11.11:0     983041    Installed 0x34eb2140(-1)
 Upstream mappings:
 Local LDP ID       Peer LDP ID       Label              CB
 10.21.21.21:0      10.11.11.11:0     983040             0x34eb2510(-1)
```

VPLS MAC age timer configuration overview

This documentation is to describe an enhancement to make the age timer configurable for both local and remote entries of VPLS MACs in the software cache.

There is an existing CLI to configure a global timer that controls MAC aging in the system (software cache). However, this configuration is not being applied to the age timers used for MAC entries associated with VPLS instances. Consequently, the age timer for VPLS MAC entries becomes hard-coded.

The VPLS application has separate age timers for different types of entries, local and remote.

It is highly desirable to make age timers fully adjustable through CLI so that you can tune the system to function most effectively based on the deployment and specific configurations.

Issues with timers

There were the following issues with the timers:

- Software MAC age timers for VPLS are *NOT* adjustable.

CLI *mac-age-time* that configures a global timer which controls MAC aging in the software cache for regular L2 is *not* being applied for MAC entries associated with VPLS instances. The age timers for VPLS MAC entries become hard-coded.

- The age timer for VPLS remote entries *is* coupled with the local entries' in values.

VPLS applications actually have separate age timers for different types of entries, local and remote. The timer for the remote entries is calculated as *two x* the age timer of local entries, which may not be desirable.

This feature addresses these issues.

Solution

The benefit of configurable MAC age timers is that you can tune your systems to function most effectively based on the deployment and on a specific configuration.

With this feature:

- The VPLS age timers are fully configurable for both local and remote entries.
- The formula "2 x" between the local timer and the remote timer is removed. Now, you have the flexibility to specify values for the age timers from 60 - 65535 seconds independently for the local and the remote entries.
- The values are bound by the same global system range shared with the regular MAC entries. The default values remains the same, which is 300 seconds for VPLS local entries and 600 seconds for the remote entries.
- Age time "0" disables the software aging. VPLS MAC follows the same format to be consistent. However, the value "0" is hidden as the valid range.
- **show mpls vpls summary** displays the age timers on MP.
- **show mpls vpls** displays the age timers on LP.
- **show running config** displays the age timers when their value becomes none, the default.
- **show mac vpls** displays value "0" for age field that is associated with a MAC entry when you disable the software aging prior to stop the traffic.
- When the software aging is disabled after the hardware aging is activated and the software aging has already started, the age field displays the time value that had been elapsed prior to the aging being disabled.
- When aging is re-enabled after software aging is disabled, the software aging resumes from the age value where it was stopped.

The MAC age timer aging operation

- The aging process only applies to MAC entries that are learned dynamically.
- A SA lookup is always performed on incoming VPLS traffic, a miss on the lookup in hardware triggers SA learning.
- A SA entry is installed in both software and hardware on the LP where it is learned.
- The entry is only installed in software on other LPs.
- It is only programmed in hardware on other LPs when there is a miss on DA lookup for this entry.

- Aging is conducted on both SA and DA entries.
- The VPLS MAC aging is involved with two steps:

1. *Hardware Aging (DA)*– Fixed at 60 seconds, carried out first. The hardware entry becomes invalid once aged out. It remains in software cache.

2. *Software Aging (SA)*– Starts at 60 seconds following the expiration of hardware age timer. The entry is removed from software cache when the configurable software timers expires.

> NOTE
> The configurable timers are only used for the software aging process.

- When an entry is aged out from the hardware, while before being aged out from software, the software age timer stops when the same entry is hit. This is true for both SA and DA.
- If it is a SA and not a station move, or is a DA, the entry is re-installed in the hardware from the local software cache and the hardware timer is re-started again as a newly learned entry.
- When it is a station move (SA), besides stopping the software timer, the MAC is sent to MP following the exact same process as learning a brand new MAC.
- A hit on a lookup in hardware refreshes the hardware timer for that MAC entry in hardware.

Backward compatibility

This feature is an enhancement to the existing functionality and the default values of the aging timers for VPLS remain the same. It is backward compatible.

Upgrade and downgrade considerations

When deploying this feature, follow the standard upgrade procedure for the XMR/MLX platform.

Scaling support

There are no changes to scaling numbers.

VPLS static MAC

The VPLS static MAC feature provides the ability to configure a static MAC address on a PE device and associate it to a VPLS endpoint.

Overview.

The VPLS static MACs captures the functionality and design for providing the ability to configure static MAC addresses at VPLS end points.

The diagram below explains how to configure PE device with static MAC and associate it to a VPLS endpoint.

VPLS static MAC enabled network

Consider MAC station **A** behind the customer edge router CE1. If **A** chooses to only receive and not to transmit any packets, then its MAC address is not learned on the provider edge PE1. When this happens, any traffic received at PE1 for **A** is flooded. If CPU protection is enabled, this flooding happens in the hardware, or else the flooding happens in the LP CPU. Also, if the link to CE1 goes down, the traffic destined to MAC Station A will be unnecessarily flooded across all the end-points.

To help in such situation, VPLS static MAC allows to configure a static MAC on a VPLS endpoint. Therefore, all packets destined to the static MAC station are hardware forwarded instead of flooding the CPU when no CPU protection is enabled. When the link to customer edge router CE1 goes down, the HW entry is reprogrammed to drop the traffic destined to MAC Station A, thereby protecting the CPU and preventing unwanted flooding in the network.

The following actions describe how static MAC is added and removed from the device.

Adding a static MAC

MAC Station **A** can be configured statically by following the configuration steps below. Once configured, the following actions are performed in the system.

1. The configured MAC address is added to the VPLS instance's MAC table in the MP.

2. The entry is also synchronized with the LPs VPLS MAC table.

   NOTE
   The maximum static MAC addresses that can be configured across all VPLS instances in the system is 1000.

Removing a static MAC

When the configuration is removed using the **no** form of the command, the following actions take place.

1. All configured hardware entries corresponding to the static MAC are deleted in the LPs.

2. The software entry is removed from the VPLS MAC table on both MP and LP.

Static MAC limit

The maximum number of static MACs that could be programmed is governed by the size of the VPLS MAC table. There is no other restriction on the number of static MACs that could be programmed.

   NOTE
   Static MACs are counted towards the total MACs learnt by the VPLS instance.

Hardware programming behavior for static MACs

Traffic destined to the statically configured MAC station are initially sent to CPU for forwarding, as there is no CAM entry in the hardware. Here the CPU forwards the packets because the software VPLS MAC table has the MAC.

This event causes the software to program the hardware so that the subsequent packets for the static MAC destination from this port are forwarded in the hardware. Therefore, the hardware is only programmed when a flow is seen for the static MAC. The programming is done only for the port on which the flow is seen to conserve the hardware resources which are used in forwarding. For MLX Series and XMR Series, once the hardware is programmed with the static MAC, it does not age out.

While creating a new hardware entry, the forwarding and dropping action depends on the state of the port on which the static MAC is configured. When the port on which static MAC is configured goes down, all programmed hardware entries are reprogrammed to drop the packets in the hardware. Once the port comes up, the programmed hardware entries are reprogrammed to forward the packets. The hardware entry also follows the STP state of the VPLS endpoint. When the port is blocked, the packets are dropped in the hardware by reprogramming the hardware entries. When the port state changes to forwarding, the hardware entries is reprogrammed to forward the packets.

Source Address learning behavior for static MACs

Learning actions for static MACs are disabled. When traffic is seen on an endpoint, whose source address (SA) matches with that of a configured static MAC, the SA learning event is not processed. At this time, the software will program a special SA CAM entry in the hardware against that port, which prevents subsequent packets from being sent from that port to the CPU for MAC learning. This helps in protecting the CPU from processing unnecessary MAC movement notifications for MACs which have already been configured as a static MAC.

SA learning behavior for MLX Series and XMR Series device

In VPLS, the CPU learns the SA and forwards the packets even if the destination address (DA) in the packet is known and programmed in the hardware. With this behavior, the user can expect packet loss when a new flow of traffic is introduced in the system destined to a static DA which may already be programmed in the hardware and this will continue until the new SA is learnt in the software and programmed in the hardware. This may cause an increment of drop counters in the TM to reflect CPU queue overflow when the rate of incoming traffic is high.

Behavior in CES 2000 Series and CER 2000 Series device

Once a static MAC is configured through the CLI, the FBD in the hardware is updated with this static MAC and will not age out. The only way to remove the hardware entry is by removing the static MAC configuration through CLI. Any traffic destined to this static MAC is always forwarded and not flooded in the hardware, unless the static MAC configuration is removed from the device. Once the static MAC configuration is removed through CLI, the FDB entry is removed and all the traffic destined to the removed static MAC is flooded in the hardware when the VPLS endpoint on which the Static MAC is configured goes down or goes to a blocking state, the FDB is reprogrammed to drop the packets destined for that MAC in the hardware.

> **NOTE**
> VPLS static MAC is supported only on tagged, double tagged, and untagged endpoints.

Forwarding Behavior

1. Local switching and traffic from MPLS uplink

    a. When a flow with the statically configured MAC as DA is seen for the first time on a port, the first few packets are sent to the CPU by the NP for forwarding and DA CAM entry programming.

    b. Once the CAM is programmed, subsequent traffic destined to the statically configured MAC station is forwarded in the hardware.

    c. Traffic destined to the statically configured MAC station is forwarded to the destination port if the port is UP.

    d. If the destination port is down, the flow is dropped in the ingress traffic manager.

Software aging behavior

The statically configured MAC entries in the VPLS MAC table never ages out in both MP and LP.

Hardware aging behavior

1. Entries programmed in the hardware for both VPLS endpoints and MPLS uplinks never age out.

2. The entries can only be deleted by removing the static MAC configuration from the CLI.

Supported end points

The following types of VPLS endpoints are supported:

1. Tagged

2. Untagged

3. Dual tagged

Hitless upgrade consideration

The VPLS subsystem is *not* hitless upgrade capable. There will be traffic loss during hitless upgrade.

Switchover behavior

1. If the standby MP is present while configuring the Static MAC on the active MP, the configuration is synchronized to the standby MP through existing configuration synchronization mechanism.

2. If the standby MP is inserted later after configuring Static MACs, the configuration is synchronized to the standby MP through existing configuration synchronization mechanism.

3. During switchover, the new active MP is always aware of the static MAC configurations made.

4. Switchover is hitless if the underlying protocols switchover without any hit.

Configuring static MAC address at VPLS endpoints

To configure the static MAC address at a VPLS endpoint:

VPLS must be pre-configured on the device before static MAC configuration.

1. Run the **router mpls** command to configure MPLS in the global configuration mode.

2. Run the **vpls** *vpls-id* command to define the VPLS ID.

3. Run the **vlan** command to configure a single tagged VLAN, to configure dual tagged VLAN run **vlan** *inner-vlan-id* command.

4. Run the **static-mac-address** *mac-address* **ethernet** *slot/port* command to configure static MAC on the VPLS endpoints.

Syntax
[no] **static-mac-address** *mac-address* **ethernet** *slot/port*

Description

mac-address specifies the MAC address of the system.

slot/port specifies the slot number or the port ID of the VPLS endpoints.

The following example explains how static MAC address is configured on a VPLS endpoints:

```
device(config)# router mpls
device(config-mpls)# vpls vpls-1 1
device(config-mpls-vpls-1)# vlan 900 inner-vlan 800
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.3333 ethernet 1/20
```

> NOTE
> The **no** form of this command will remove the static MAC configuration on a VPLS
> endpoints.

Limitations

- Static MACs can only be configured on VPLS endpoints.
- Configuring static MAC is not supported on a VPLS uplink.
- Static MACs cannot be configured if the VPLS instance has PBB or MCT configured.

VPLS static MAC error messages

Following are the error message displayed when VPLS static MAC is not supported for different scenarios.

If port not configured as part of the VPLS VLAN:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address
0000.1111.2222 ethernet 1/20

Error: port not part of this VPLS VLAN
```

If port is not part of this VPLS instance:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: port not part of this VPLS instance
```

If port is out of range, empty slot and if module type not configured in the system:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 4/1

Error: interface 4/1 is not an ETHERNET interface

device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/30

Error: invalid interface 1/30, if the interface is out of range.
```

If configuration is done on a secondary port of a LAG:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: Static MAC cannot be configured on a secondary port of a LAG.
```

If VPLS instance has PBB or MCT configured:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: Static MAC support not available for a VPLS with MCT or PBB enabled.
```

If VPLS instance has 802.1ah enabled (for bridging only):

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: Static MAC support not available for a VPLS with 802.1ah enabled
```

If MAC is a Zero MAC:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.0000.0000 ethernet 1/20

Error: Static MAC cannot be zero MAC.
```

If MAC is a Multicast MAC:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0100.1234.5678 ethernet 1/20

Error: Static MAC cannot be a multicast MAC.
```

If MAC is same as one of the local interface MACs:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0009.3400.0001 ethernet 1/20
```

Error: Static MAC cannot be same as interface MACs.

If MAC is Broadcast MAC:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address ffff.ffff.ffff ethernet 1/20
```

Error: Static MAC cannot be broadcast MAC.

If MAC is already configured on another port of the same VPLS instance:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20
```

Error: VPLS 2, MAC 0000.1111.2222 already exists on port 1/23, VLAN 900

If endpoint is double tagged:

Error: VPLS 2, MAC 0000.1111.2222 already exists on port 1/23, VLAN 900, Inner Tag: 1000

If Global VPLS MAC MAC limit reached:

Error: VPLS 1, Global VPLS MAC MAC limit (2048) reached.

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20
```

Error: VPLS 1, Global VPLS MAC MAC limit (2048) reached.

If per VPLS instance MAC MAC limit reached:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20
```

Error: VPLS 1, VPLS instance MAC limit (512) reached

Configuring MPLS *Virtual Leased Line (VLL)*

Overview

This chapter explains how to configure MPLS Virtual *Leased Line (VLL)* on an Extreme device. Virtual Leased Line is also known as Pseudo Wire Emulation as defined by the IETF PWE3 Working Group. MPLS VLL is a method for providing point-to-point Ethernet or VLAN connectivity over an MPLS domain. This functionality is outlined in the IETF documents "draft-ietf-pwe3-control-protocol-14.txt" and "draft-ietf-pwe3-ethernet-encap-08.txt".

This chapter is divided into the following sections:

- How MPLS VLL works on page 334 describes how packets are encapsulated and forwarded over an MPLS VLL.

- Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints on page 341 describes how to set up MPLS VLLs on devices using the *Command Line Interface (CLI)*.

- Displaying MPLS VLL information on page 354 describes the commands used to display information about an MPLS VLL configuration.

- Sample MPLS VLL configuration on page 356 illustrates a sample MPLS VLL configuration and lists the CLI commands used for implementing it.

How MPLS VLL works

The following diagram illustrates how packets are forwarded over an MPLS VLL.

FIGURE 61 Forwarding packets over an MPLS VLL
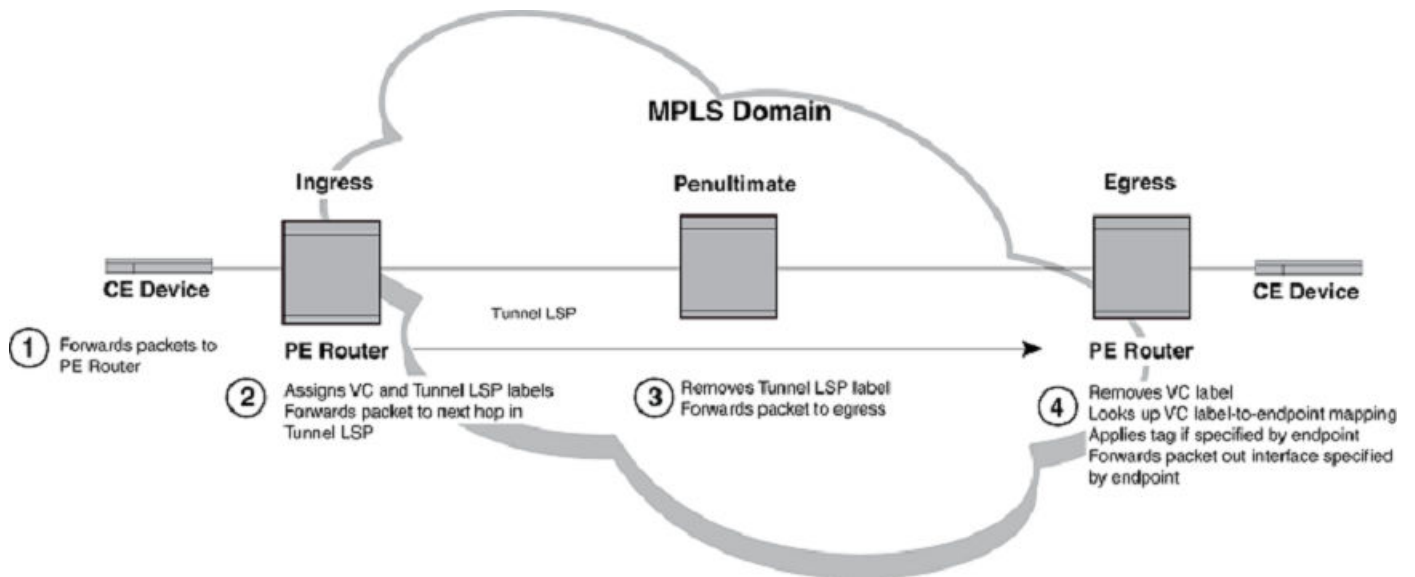
Packets are forwarded over an MPLS VLL as described below.

1. A *Customer Edge (CE)* device forwards a packet to a *Label Edge Router (LER)* serving as a *Provider Edge (PE)* router at the edge of the MPLS domain.

2. The PE router assigns the packet to an RSVP-signaled LSP whose destination is an LER (also serving as a PE router) that is connected to a CE device at the far end of the MPLS domain. The PE router at the other end of the MPLS domain is known as this PE router's VLL peer. The RSVP-signaled LSP used to reach the VLL peer is known as the tunnel LSP. Alternatively, an LDP-signaled, tunneled LSP can be used.

   When a *Class of Service (CoS)* value is set for the VLL, the device selects a tunnel LSP that also has this CoS value, when one is available. When no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VLL). Refer to QoS for VLL traffic on page 336 for more information.

   When there are multiple tunnel LSPs that can be used to reach the VLL peer, the PE router selects one of the tunnel LSPs by using a round-robin method.

   The PE router pushes two labels onto the packet:

   • The inner VC label is used for determining what happens to the packet once it reaches the VLL peer. This label is significant only to the VLL peer.

   • The outer tunnel label is used for forwarding the packet through the MPLS domain. This label corresponds to an RSVP-signaled tunnel LSP.

   Refer to MPLS VLL packet encoding on page 335 for information on the structure of packets forwarded along an MPLS VLL. After applying the two labels to the packet, the PE router forwards it to the next LSR in the tunnel LSP.

3. The penultimate LSR in the tunnel LSP removes the tunnel label and forwards the packet (now with the VC label as the top label) to the PE router at the other edge of the MPLS domain.

4. The VLL peer at the egress of the tunnel LSP examines the VC label. This VC label is mapped to an endpoint for the VLL. The endpoint of a VLL specifies what happens to packets exiting the VLL.

   The endpoint can specify an untagged, dual-tagged, or single-tagged port.

   • For *untagged* ports, the endpoint consists of an interface.

   • For *single-tagged* ports, the endpoint consists of an interface and a VLAN ID.

   • For *dual-tagged* ports, the endpoint consists of an interface and dual (outer and inner) VLAN IDs.

   The egress LER removes the VC label and forwards the packet out the interface specified as the endpoint. When the endpoint is a single-tagged or dual-tagged port, the device transmits the packet with the specified VLAN ID, or with the dual tags, forwarding it out the specified interface to the CE device.

   The two VLL peers advertise VC labels to each other using the *Label Distribution Protocol (LDP)*. Each PE router attempts to initiate an LDP session with its VLL peer. After the LDP session is established, the locally assigned VC label, along with a VLL VC ID, is advertised to the VLL peer. In a similar way, the PE also learns the remotely assigned VC label from the VLL peer. Alternatively, the user can configure static local and remote VC labels manually on both VLL peers; in this case, LDP is not used.

   MPLS VLLs are not involved with spanning tree operations.

   > **NOTE**
   > When MTUs are mismatched on both sides of a VLL session, the session does not come up.

MPLS VLL packet encoding

When a packet is forwarded from the CE device, the PE router encapsulates it as an MPLS packet, applying two labels. The resulting MPLS packet has the following structure.

FIGURE 62 Structure of a packet forwarded over an MPLS VLL
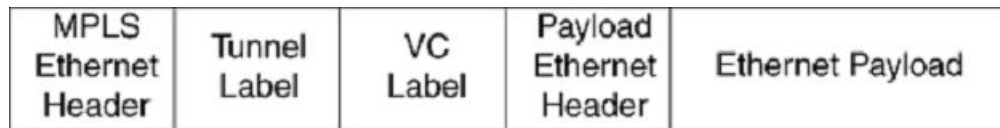
The S bit in the tunnel label is zero, indicating that it is not the bottom of the stack. The VC label is significant only to the PE router at the other end of the VLL.

The Payload Ethernet header may be single-tagged or untagged.

Trunk load balancing of VLL traffic

Load balancing of VLL traffic across trunk ports behaves differently on XMR Series and MLX Series devices compared to CES 2000 Series and CER 2000 Series devices. The following describes the difference:

- On XMR Series and MLX Series devices, VLL traffic load balances across all trunk ports.
- On CES 2000 Series and CER 2000 Series devices, only one of the trunk ports is used for a given VLL instance, depending on the VC label used by the instance.

QoS for VLL traffic

By default, packets traveling through an MPLS domain are treated equally from a QoS standpoint, in a best effort manner. However, when a Layer 2 packet has an internal priority in its 802.1q tag, or the LSP or VLL to which the packet is assigned has a configured *Class of Service (CoS)* value, QoS can be applied to the packet in the MPLS domain. The internal priority or CoS value is mapped to a value in the EXP field of the packet's MPLS header. The value in the EXP field is then mapped to an internal forwarding priority, and the packet is sent to the hardware forwarding queue that corresponds to the internal forwarding priority.

QoS for VLL traffic at the ingress LER

The following methods can be used to provide QoS to packets entering a VLL:

- Use the CoS value assigned to the tunnel LSP used to reach the VLL peer-

When a tunnel LSP has a user-configured CoS value, all packets in all VLLs travailing through the tunnel LSP receive the same QoS.

- Use the CoS value assigned to the VLL-

When a CoS value is set for the VLL, the device selects a tunnel LSP that also has this CoS value, when one is available. When no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VLL).

When the selected tunnel LSP does not have a CoS value, the VLL's configured CoS value is used to provide QoS. The VLLs CoS value is mapped to a value in the EXP field. This allows traffic multiple VLLs using a single tunnel LSP, traffic from each VLL can receive different QoS treatment.

- Use the priority in the packet's 802.1q tag-

When neither the tunnel LSP nor the VLL has a configured CoS value, the device examines the priority in the Layer 2 packet's 802.1q tag, when the packet has one. Consequently, Layer 2 packets with the same 802.1q priority receive the same QoS in the VLL.

- Use the configured priority of the port-

When neither the tunnel LSP nor the VLL has a configured CoS value, and the Layer 2 packet does not have an 802.1q priority, QoS can be provided based on the priority of the incoming port. A port can be assigned a priority from 0 (lowest priority) to 7 (highest priority). The default port priority is 0.

By assigning different priorities to the ports where customer edge (CE) devices are connected (that is, the VLL endpoints), the user can provide QoS to untagged Layer 2 traffic received from different customer locations.

When a packet enters a VLL, the PE router that serves as both the VLL endpoint and the ingress of a tunnel LSP pushes two labels onto the packet the inner VC label and the outer tunnel label. The packet's priority resides in the EXP field of the MPLS label header. The VC label and the tunnel label carry the same value in the EXP field.

The following table lists how a Layer 2 packet's priority is mapped to a value in the EXP field and how the EXP value is mapped to a priority queue.

Tunnel LSP configured CoS or VLL configured CoS or 802.1q priority or Configured port priority	Value placed in the tunnel and VC label EXP field	Priority queue
7	7	qosp7 (highest priority)
6	6	qosp6
5	5	qosp5
4	4	qosp4
3	3	qosp3
2	2	qosp2
1	1	qosp1
0	0	qosp0 (best effort)

QoS for VLL traffic at transit LSRs

At each transit LSR, the device reads the value in the tunnel label's EXP field and places the incoming EXP value in the EXP field of the outbound packet. The outbound MPLS packet is assigned to one of the eight priority queues based on the value in the EXP field. The EXP bits in the MPLS header are used to assign the packet to a priority queue as follows:

EXP Bits in tunnel label	Priority queue
7	qosp7 (highest priority)
6	qosp6
5	qosp5
4	qosp4
3	qosp3
2	qosp2
1	qosp1
0	qosp0 (best effort)

QoS for VLL traffic at the penultimate LSR

When the packet reaches the penultimate LSR in the LSP, its tunnel label is popped, leaving the VC label. The MPLS packet is placed in one of the priority queues using the value in the EXP field of the VC label. Since the VC label has the same EXP value as the tunnel label, the packet is placed in the same queue used for the tunnel LSP.

QoS for VLL traffic at the egress LER

At the VLL endpoint, the VC label is popped and the packet is forwarded as a Layer 2 packet. The packet is placed in one of the priority queues based on the contents of the EXP field in the VC label, as follows:

EXP Bits in VC label	Priority queue
6, 7	qosp3 (highest priority)
4, 5	qosp2
2, 3	qosp1
0, 1	qosp0 (best effort)

CoS behavior for VLL tagged mode and VLL raw mode

This section describes the difference in CoS behavior for VLL traffic when tagged mode or raw mode is in effect.

CoS behavior for VLL tagged mode

> **NOTE**
> This section assumes that the user understands how QoS works.

> **NOTE**
> On CES 2000 Series and CER 2000 Series devices, when VLL is configured in tagged mode, the priority present in inner VLAN tag is not copied to the VLAN being sent out over the endpoint.

Table 31 describes the expected *Class of Service (CoS)* behavior for VLL packets when VLL tagged mode is enabled.

TABLE 31 Expected Class of Service behavior for VLL tagged mode

VLL endpoints	Incoming packet		MPLS cloud		Outgoing packet	
Outer VLAN	Inner VLAN	Tunnel/VClabel (Z)	Payload tag	Outer VLAN	Inner VLAN	
Dual-tagged to dual-tagged	X	Y	V or internal priority	Y	W or Y	Y
Single-tagged to dual-tagged	X	N/A		X	W or X	X
Untagged to dual-tagged	N/A	N/A		0	W or 0	0
Dual-tagged to single-tagged	X	Y		Y	W or Y	N/A

V = Mapped EXP bits from internal priority) using the EXP encode table.

W = Mapped CoS from internal priority (**Z** contributes to internal priority) using the CoS encode table.

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

Z = Incoming EXP bits as described by 'tunnel/VC label' column = **V** internal priority *or* in the 'tunnel/VC' label column differentiating the behavior between when the **qos exp encode** policy is ON (default) or OFF, *or* in the 'outgoing packet outer VLAN' column differentiating the behavior between when the **qos pcp encode** policy is ON (default) or OFF.

NOTE

For more specific examples of CoS behavior for tagged mode, see Example 1: CoS behavior for dual-tagged to dual-tagged VLL endpoints on page 339 and Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints on page 341.

CoS behavior for VLL raw mode

NOTE

This section assumes that the user understands how QoS works.

Table 32 describes the expected *Class of Service (CoS)* behavior for VLL packets when VLL raw mode is in effect.

TABLE 32 Expected Class of Service behavior for VLL raw mode

VLL endpoints	Incoming packet		MPLS cloud		Outgoing packet		
Outer VLAN	Inner VLAN	Tunnel/VClabel (Z)	Payload tag	Outer VLAN	Inner VLAN		
Dual-tagged to dual-tagged	X	Y	V or internal priority	N/A	W or Z	Z	
Single-tagged to dual-tagged	X	N/A				Z	
Untagged to dual-tagged	N/A	N/A				Z	
Dual-tagged to single-tagged	X	Y				N/A	

V = Mapped EXP bits from internal priority (**X** contributes to internal priority) using the EXP encode table.

W = Mapped CoS from internal priority (**Z** contributes to internal priority) using the Cos encode table.

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

Z = Incoming EXP bits as described by 'tunnel/VC label' column = **V** or internal priority *or* in the 'tunnel/VC label' column differentiating the behavior when **qos exp encode** policy is ON (default) or OFF *or* in the 'outgoing packet outer VLAN' column differentiating the behavior when **qos pcp encode** policy is ON (default) or OFF.

NOTE

For more specific examples of CoS behavior for raw mode, see Example 1: CoS behavior for dual-tagged to dual-tagged VLL endpoints on page 339 and Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints on page 341.

Example 1: CoS behavior for dual-tagged to dual-tagged VLL endpoints

Table 33 shows a detailed example of the CoS behavior in a dual-tagged to dual-tagged VLL endpoint configuration. The table shows the difference in behavior for VLL tagged mode (described in CoS behavior for VLL tagged mode on page 338) versus VLL raw mode (described in CoS behavior for VLL raw mode on page 339).

TABLE 33 Example CoS behavior in a dual-tagged to dual-tagged VLL endpoint configuration

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode		
Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN		
LSP CoS 4	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 4	VLAN 400, CoS 0	VLAN 300 CoS 4	VLAN 400 CoS 4	

TABLE 33 Example CoS behavior in a dual-tagged to dual-tagged VLL endpoint configuration (continued)

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode	
VLL CoS 2	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 2	VLAN 400, CoS 0	VLAN 300 CoS 2	VLAN 400 CoS 2
Port priority 6 (with priority force)	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 6	VLAN 400, CoS 0	VLAN 300 CoS 6	VLAN 400 CoS 6
802.1p CoS 6 (outer VLAN) CoS 4 (inner VLAN)	VLAN 100, CoS 6	VLAN 200, CoS 4	VLAN 300, CoS 6	VLAN 400, CoS 4	VLAN 300 CoS 6	VLAN 400 CoS 6
Port priority 6 and VLL CoS 2	VLAN 100, CoS 0	VLAN 200, CoS 0	VLAN 300, CoS 2	VLAN 400, CoS 0	VLAN 300 CoS 2	VLAN 400 CoS 2
Port priority 5 (with priority force)	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 5	VLAN 400, CoS 7	VLAN 300 CoS 5	VLAN 400 CoS 5
Port priority 5 (with priority force), LSP CoS 3	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 3	VLAN 400, CoS 7	VLAN 300 CoS 3	VLAN 400 CoS 3
Port priority 5 (with priority force), LSP CoS 2. VLL CoS 4	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 2	VLAN 400, CoS 7	VLAN 300 CoS 2	VLAN 400 CoS 2
Port priority 5 (with priority force), LSP CoS 0. VLL CoS 4	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 7	VLAN 300 CoS 0	VLAN 400 CoS 0
Port priority 5 (with priority force), LSP no value. VLL CoS 4	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 4	VLAN 400, CoS 7	VLAN 300 CoS 4	VLAN 400 CoS 4
Port priority 5 (with priority force), LSP CoS 0. VLL CoS 4 QoS exp encode policy all-zero (ingress router)	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 7	VLAN 300 CoS 0	VLAN 400 CoS 0
Port priority 5 (with priority force), LSP CoS 3. VLL CoS 4 QoS exp encode policy all-zero (ingress router)	VLAN 100, CoS 7	VLAN 200, CoS 7	VLAN 300, CoS 0	VLAN 400, CoS 7	VLAN 300 CoS 0	VLAN 400 CoS 0
Port priority 5	VLAN 100,	VLAN 200,	VLAN 300,	VLAN 400,	VLAN 300,	VLAN 400,

TABLE 33 Example CoS behavior in a dual-tagged to dual-tagged VLL endpoint configuration (continued)

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode	
(with priority force), LSP CoS 3. VLL CoS 4 QoS PCP encode policy all-zero (egress router)	CoS 7	CoS 7	CoS 0	CoS 7	CoS 0	CoS 3
Port priority 5 (with priority force), LSP CoS 3. VLL CoS 4 QoS PCP decode policy string (ingress router) (mapping of 7 to 1)	VLAN 100, CoS 7	VLAN 200, CoS 6	VLAN 300, CoS 3	VLAN 400, CoS 6	VLAN 300, CoS 3	VLAN 400, CoS 3
QoS PCP decode policy string (ingress router) (mapping of 7 to 1)	VLAN 100, CoS 7	VLAN 200, CoS 6	VLAN 300, CoS 1	VLAN 400, CoS 6	VLAN 300, CoS 1	VLAN 400, CoS 1

Example 2: CoS behavior for dual-tagged to single-tagged VLL endpoints

Table 34 shows a detailed example of the CoS behavior in a dual-tagged to single-tagged VLL endpoint configuration. The table shows the difference in behavior for VLL tagged mode (described in CoS behavior for VLL tagged mode on page 338) versus VLL raw mode (described in CoS behavior for VLL raw mode on page 339).

TABLE 34 Example CoS behavior in a dual-tagged to single-tagged VLL endpoint configuration

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode	
Outer VLAN	Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN	Outer VLAN	
LSP CoS 4,	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 4	NA	VLAN 300 CoS 4	NA
VLL CoS 2	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 2	NA	VLAN 300 CoS 2	NA
Port priority 7 (with priority force)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 7	NA	VLAN 300 CoS 7	NA
802.1p CoS 6 (outer VLAN)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 6	NA	VLAN 300 CoS 6	NA
Port priority 7 and VLL CoS 2	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 2	NA	VLAN 300 CoS 2	NA
Port priority 7 (with priority force), LSP CoS 3	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300 CoS 3	NA

TABLE 34 Example CoS behavior in a dual-tagged to single-tagged VLL endpoint configuration (continued)

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode	
Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300 CoS 3	NA
Port priority 7 (with priority force), LSP CoS 0. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 0	NA	VLAN 300 CoS 0	NA
.1p is 6 for outer VLAN, 5 for inner VLAN Port 3 No LSP CoS VLL CoS 4 (ingress above) Egress below Port is 7 VLL CoS 2	VLAN 100 CoS 6	VLAN 200 CoS 5	VLAN 300 CoS 7	NA	VLAN 300 CoS 7	NA
Port priority 7 (with priority force), LSP no value. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 4	NA	VLAN 300 CoS 4	NA
Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300 CoS 3	NA
Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4 QoS exp encode policy all-zero (ingress router)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 0	NA	VLAN 300 CoS 0	NA
Port priority 7 (with priority force), LSP CoS 3. VLL CoS 4 QoS PCP encode policy all-zero (egress router)	VLAN 100, CoS 6	VLAN 200 CoS 5	VLAN 300, CoS 0	NA	VLAN 300, CoS 0	NA
Port priority 6 (with priority force), LSP CoS 3. VLL CoS 4	VLAN 100, CoS 7	VLAN 200 CoS 5	VLAN 300, CoS 3	NA	VLAN 300, CoS 3	NA

TABLE 34 Example CoS behavior in a dual-tagged to single-tagged VLL endpoint configuration (continued)

Priority	Incoming Packet		Outgoing Packet Tag Mode		Outgoing Packet Raw Mode	
QoS PCP decode policy string (ingress router) (mapping of 7 to 1)						
QoS PCP decode policy string (ingress router) (mapping of 7 to 1)	VLAN 100, CoS 7	VLAN 200 CoS 5	VLAN 300, CoS 1	NA	VLAN 300, CoS 1	NA

Configuring MPLS VLLs

This section explains how to set up MPLS VLLs.

Creating a VLL

The user creates a VLL by entering VLL configuration statements on two PE routers. The two endpoints of a VLL are associated by having the same VLL VC ID on each PE router.

To create an MPLS VLL, enter commands such as the following:

```
device(config-mpls)# vll foundry-sj-to-sf 40000
device(config-mpls-vll)#
```

On the VLL peer (when it is a device), the user would enter commands such as the following:

```
device(config-mpls)# vll foundry-sf-to-sj 40000
device(config-mpls-vll)#
```

Syntax: **vll** *vll-name* | *vll-vc-id* [**CoS** *CoSvalue*] [**raw-mode**]

The *vll-vc-id* corresponds to the user-configurable ID defined in draft-ietf-pwe3-control-protocol-14.txt.

The user can optionally specify a *Class of Service (CoS)* setting for the VLL. When a CoS value is set, the device selects a tunnel LSP that also has this CoS value, when one is available. when no tunnel LSP with this CoS value is available, the device selects a tunnel LSP with the highest configured CoS value (although never higher than the CoS setting for the VLL). The CoS value can be between 0 - 7.

> **NOTE**
> On CES 2000 Series and CER 2000 Series devices, the route-only command must not be configured on untagged MPLS uplinks when using it for VLL. Otherwise, incoming VLL traffic is dropped.

Specifying tagged or raw mode for a VLL

The default treatment for packets that are sent through a VLL is for the ingress router to add a VLAN ID tag to the payload Ethernet header. When the packet arrives at the egress router, the tag is stripped off and the packet is forwarded.

The user can configure a VLL to send all packets in "raw" mode. This means that the ingress router of the VLL does not add a VLAN ID tag to the payload Ethernet header and consequently the egress router does not have to strip it off. Both the ingress and egress routers

must be configured in either default (tagged mode) or raw mode. To configure a router to send or receive packets for a VLL in raw mode, enter the following command.

```
device(config-mpls)# vll <vll-name> <vll-vc-id> raw-mode
```

Syntax: vll *vll-name* | *vll-vc-id* [**raw-mode**]

The *vll-name* is the name of the VLL the user wants to configure raw mode for.

The *vll-vc-id* corresponds to the user-configurable ID defined in draft-ietf-pwe3-control-protocol-14.txt.

When **raw-mode** is specified, the VC type for signaling is five (5), otherwise it is four (4) (for tagged mode).

When **raw-mode** is not specified, the default configuration is for the ingress router to send packets with a tag, and for the egress router to strip it off before forwarding the packets.

> **NOTE**
> when there is a VC type mismatch between VLL peers, a session is not be brought up between them.

Specifying a VLL peer

The VLL peer is the PE router at the other end of the VLL. As part of VLL configuration, the user specifies the IP address of the VLL peer.

Each PE router must have tunnel LSP reachability to its VLL peer. Tunnel LSP reachability is defined as having at least one operational LSP tunnel with the destination (the LSPs "to" address) matching the VLL peer's IP address. An LSP terminating on the VLL peer but configured with a different destination address would not be considered a match.

When a PE router does not have tunnel LSP reachability to its VLL peer, or when the remote VC label is not yet available, packets from the local interface are discarded at the ingress PE router. When the local interface is administratively disabled or goes down, a VC label withdraw message is sent to the VLL peer.

By default, each PE router attempts to initiate an LDP session through extended discovery with its VLL peer, when a session is not already established. The PE router also allocates a VC label from a per-platform label range that is mapped to the local endpoint. Once the LDP session is established, the locally assigned VC label, along with the VLL VC ID is advertised to the VLL peer in a downstream-unsolicited manner. In a similar way, the PE also learns the remotely assigned VC label from the VLL peer.

Alternatively, the user can configure static local and remote VC labels. In this case, no LDP session is established between the VLL peers. Note that when the user makes use of static VC labels, the user must configure them on both VLL peers manually.

The user specify the peer at the other end of the VLL by entering a command such as the following.

```
device(config-mpls-vll)# vll-peer 192.168.2.100
```

The IP address of the peer must match that of a destination for a tunnel LSP configured on the device.

For additional information, see the **vll-peer** command in *NetIron Command Line Interface (CLI) Reference Guide*.

Specifying a VLL endpoint

The endpoint of a VLL specifies what happens to packets exiting the VLL. The user set the endpoint on the local PE router and this endpoint is mapped to a VC label. The VC label is advertised to the remote PE router at the other end of the VLL through LDP. The remote PE router applies this label to packets entering the VLL. When the packet reaches the end of the VLL through the MPLS uplink, the local PE router checks the mapping between the VC label and the endpoint, removes the VC label from the packet, and forwards the packet out the port specified as the endpoint.

All VLL endpoints can be dual-mode ports (tagged-untagged). An untagged endpoint port is removed from the default VLAN and cannot be added back to the default VLAN. A VLL endpoint can be tagged in multiple VLL and L2 VLANs and untagged in one other VLAN.

The *Customer Edge (CE)* device is connected to the PE router over an untagged, dual-tagged, or single-tagged port.

- With a *single-tagged* port, each pair (port, VLAN ID) is identified as a unique endpoint, and the packets are sent in tagged Ethernet format.
- In the case of an *untagged* port, an endpoint is identified by the physical port alone, and the packets are sent in untagged Ethernet format.
- In the case of a *dual-tagged* port, the packets contain both an outer VLAN tag and an inner VLAN tag.

Configuring VLL endpoint over FDP/CDP enabled interface

Configuring VLL endpoint over an FDP/CDP enabled interface will implicitly disable FDP/CDP configuration and also will be enable back implicitly when the VLL endpoint is deleted on that specific interface, considering FDP/CDP is enabled globally.

Info messages will be displayed to notify the user as below for these cases:

For example, when VLL endpoint is created the info messages displayed are as below.

- When only FDP is enabled globally.

```
device(config-mpls-vll-vll1-vlan-100)#tag eth 4/3 eth 4/5 eth 4/7
info- FDP is disabled on port 4/3
info- FDP is disabled on port 4/5
info- FDP is disabled on port 4/7
```

- When only CDP is enabled globally.

```
device(config-mpls-vll-vll1-vlan-100)#tag eth 4/3 eth 4/5 eth 4/7
info- CDP is disabled on port 4/3
info- CDP is disabled on port 4/5
info- CDP is disabled on port 4/7
```

- When both FDP/CDP is enabled globally.

```
device(config-mpls-vll-vll1-vlan-100)#tag eth 4/3 eth 4/5 eth 4/7
info- FDP/CDP is disabled on port 4/3
info- FDP/CDP is disabled on port 4/5
info- FDP/CDP is disabled on port 4/7
```

For example, when VLL endpoint is deleted the info messages are displayed as below.

- When only FDP is enabled globally.

```
device(config-mpls-vll-vll1-vlan-100)#no tag eth 4/3 eth 4/5 eth 4/7
info- FDP is enabled on port 4/3
info- FDP is enabled on port 4/5
info- FDP is enabled on port 4/7
```

- When only CDP is enabled globally.

```
device(config-mpls-vll-vll1-vlan-100)#no tag eth 4/3 eth 4/5 eth 4/7
info- CDP is enabled on port 4/3
info- CDP is enabled on port 4/5
info- CDP is enabled on port 4/7
```

- When both FDP/CDP is enabled globally.

```
device(config-mpls-vll-vll1-vlan-100)#no tag eth 4/3 eth 4/5 eth 4/7
info- FDP/CDP is enabled on port 4/3
info- FDP/CDP is enabled on port 4/5
info- FDP/CDP is enabled on port 4/7
```

NOTE
If the VLL endpoint is configured over a globally enabled FDP/CDP interface, the **show run** command does not display FDP/CDP information for that specific interface.

NOTE
By removing FDP from the configuration, **no fdp enable** stays in the configuration of the VPLS endpoints, which cannot be removed.

Special considerations for VLL dual-tagged endpoints

Before configuring a dual-tagged endpoint, consider the following:

- An *Internal Forwarding Lookup Identifier (IFL-ID)* is be allocated to each MPLS VLL instance that has a dual-tagged endpoint. The ID is displayed in the **show mpls vll detail** command output. For instances that do not have dual-tagged endpoints, the IFL-ID is be displayed as '--'.

- The *Tag Protocol Identifier (TPID)* of the inner VLAN tag must be 0x8100 in order to be classified as dual-tagged and recognized by dual-tagged endpoints. When the TPID is not 0x8100, the packet is be classified as a single-tagged packet.

- The same port, outer VLAN, and inner VLAN combination cannot be specified across MPLS VLL instances. For example, when a dual-tagged endpoint with vlan 100 and inner-vlan 200 is configured on port e 2/1 in MPLS VLL instance 'test', the same endpoint cannot be configured as part of another MPLS VLL instance, say 'test1'. This is also true across applications. That is, when a port, outer VLAN, and inner VLAN combination belongs to a MPLS VLL instance, it cannot simultaneously belong to a Layer 2 VLAN, Local VLL or VPLS.

- To change an existing single-tagged VLL endpoint to a dual-tagged endpoint, first delete the VLAN configuration, then configure the endpoint as dual-tagged.

- A dual-tagged VLL endpoint neither recognizes nor forwards packets that have a single tag. However, a single-tagged endpoint can recognize and forward dual-tagged packets because the endpoint treats the second tag as data.

- The port, outer VLAN, and inner VLAN combination in an incoming dual- tagged packet on a given port is used to do an IFL CAM lookup. This lookup yields an IFL-ID which is used to do a MPLS-VLL CAM lookup. So for dual-tagged endpoints, the regular (port, vlan) lookup is replaced with the (port, IFL-ID) lookup.

- When only the outer VLAN is specified for a given endpoint, it is called a *less-specific* VLAN. When both the outer and inner VLAN are specified, it is called a *more-specific* VLAN (in relation to the outer VLAN).

- When a less-specific VLAN is already configured on a given port, then a more-specific VLAN with the same outer VLAN tag can also be configured on that port. Likewise, when a more-specific VLAN is already configured on a given port, then a less-specific VLAN with the same outer VLAN tag can also be configured on the port.

In the following example, a less-specific tagged endpoint has been configured with vlan 100 on port e 2/1, and a more-specific VLAN with an outer VLAN tag of **100** and an inner vlan tag of **200** has also been configured on port e 2/1.

```
device(config-mpls)# vll test1 1000
device(config-mpls-vll-test1)# vlan 100
device(config-mpls-vll-test1-vlan)# tag e 2/1
device(config-mpls-vll-test1-if-e-2/1)# vll test2 2000
device(config-mpls-vll-test2)# vlan 100 inner-vlan 200
device(config-mpls-vll-test2-vlan)# tag e 2/1
```

This applies even when the less or more-specific VLAN is configured as part of a L2 VLAN, Local VLL or VPLS.

Specifying an untagged endpoint

Untagged ports are not associated with any VLAN. A port must be a member of the default VLAN before it can be used in a VLL configuration as an untagged port. Upon configuration as the endpoint of a VLL, the port is taken out of the default VLAN. This means no local broadcast traffic includes this port. A VLL untagged port does not belong to any VLAN. When the port is currently a member of a regular VLAN or another VLL, the configuration attempt must be rejected.

> **NOTE**
> When a port is added as an untagged port into a VLL, a VLAN must not be defined under the VLL. When a VLAN is configured under the VLL, the configuration to add an untagged port is rejected.

To specify an untagged endpoint for a VLL.

```
device(config-mpls-vll)# untagged e 2/1
```

Syntax: untagged [**ethernet**] *portnum*

> **NOTE**
> *Foundry Discovery Protocol (FDP)* must not be enabled on an untagged VPLS or VLL endpoint.

Specifying a single-tagged endpoint

Tagged ports are configured under a VLAN ID. This VLAN ID is only meaningful for this tagged port. Another tagged port may use the same VLAN ID but the two ports are not under the same VLAN.

For tagged ports, a *vlan-id* , *port* pair constitutes a VLL endpoint. One VLL may configure a VLAN ID on one port and another VLL may reuse the same VLAN ID on another port. This capability is know as VLAN ID reuse.

As with regular VLANs, when a port is currently a member of a non-default VLAN as an untagged port, it must be returned to the default VLAN before it can be assigned to a VLL as a tagged port.

To specify a tagged endpoint for a VLL.

```
device(config-mpls-vll)# vlan 200
device(config-mpls-vll-vlan)# tagged e 3/11
```

Syntax: vlan *num*

Syntax: tagged [**ethernet**] *slot/port*

> **NOTE**
> A tagged port can be a part of one or more VLLs, and at the same time be part of one or more regular VLANs and one or more VPLSs as a tagged member.

Specifying a dual-tagged endpoint

Dual-tagged packets contain both an outer VLAN tag and an inner VLAN tag. Dual-tagged VLL endpoints enable MPLS VLL to recognize packets with two tags and make forwarding decisions based on them. A dual-tagged endpoint can receive packets with two tags and forward them to the other endpoint either untagged, single-tagged, or dual-tagged.

> **NOTE**
> Before configuring a dual-tagged endpoint, see

To specify a dual-tagged endpoint for a VLL instance, enter commands such as the following:

```
device(config-mpls)# vll test 100
device(config-mpls-vll-test)# vlan 200 inner-vlan 300
device(config-mpls-vll-test-vlan)# tagged e 3/11
```

Syntax: [no] vlan *vlan-id* inner-vlan *vlan-id*

Syntax: [no] tagged ethernet *slot/port*

The **vlan** *vlan-id* , which is the outer VLAN ID, can be in the range from 1 to 4094 and excludes the default VLAN.

The **inner-vlan** *vlan-id* can be in the range from 1 to 4095 and includes the default VLAN.

Use the [no] form of the command to remove the dual-tagged VLL VLAN configuration and its associated endpoints. For example, the command **no vlan** *200* **inner-vlan** *300* removes the dual-tagged VLAN and associated endpoints. The single-tagged VLAN, **vlan 200**, is not deleted. Similarly, the command **no vlan** *200* removes the single-tagged VLAN, **vlan 200**, and associated endpoints. The dual-tagged VLAN, **vlan** *200* **inner-vlan** *300* , is not deleted.

> **NOTE**
> A tagged port can be a part of one or more VLLs, and at the same time be part of one or more regular VLANs and one or more VPLSs as a tagged member.

Viewing the VLL dual-tagged configuration

Use the **show running config** and **show mpls vll** commands to view the VLL dual-tagged configuration. The following shows an example **show running config** output. For examples and details of the **show mpls vll** commands, see Displaying information about MPLS VLLs on page 354.

```
device# show run
....
router mpls
vll test 100
  vlan 100 inner-vlan 45
  tag e2
```

Specifying a LAG group as the endpoint of a VLL

The endpoint of a VLL can be a LAG group. When the endpoint of a VLL is a LAG group, the VLL traffic load is distributed to the *Customer Edge (CE)* device across all of the LAG group's ports, using a hashing mechanism.

Figure 63 illustrates a sample configuration where a LAG group of two ports serves as the endpoint of a VLL.

FIGURE 63 Specifying a LAG group as the endpoint of a VLL
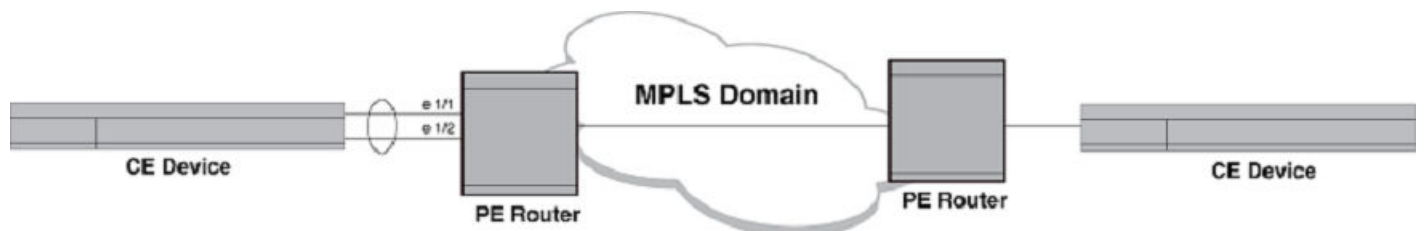
To configure a LAG group like the one in Figure 63, enter commands such as the following.

```
device(config)# lag red dynamic
device(config-lag-red)# ports ethernet 1/1 to 1/2
device(config-lag-red)# primary port 1/1
```

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

To configure a VLL like the one in Figure 63, enter commands such as the following.

```
device(config)# router mpls
device(config-mpls)# vll test2 40000
device(config-mpls-vll)# vll-peer 10.10.10.10
device(config-mpls-vll)# untagged e 1/1
```

> **NOTE**
> When the user first creates a LAG and then configure a VLL instance, the port the user specifies as the VLL endpoint must also be the port the user specified as the primary port of the LAG group:

- When the user later deletes the LAG from the configuration, only the primary port is still a port of the VLL and all secondary ports become normal ports.

- When the user specified a tagged endpoint for the VLL instance, all of the ports in the LAG must be tagged.

- Traffic received from any port in the LAG is forwarded to the VLL instance. All traffic is matched to its VLAN.

- Both static and dynamic LAGs are supported.

Enabling VLL MTU enforcement (optional)

The user can selectively enable local and remote VLL MTU mismatch checking using the following command.

```
device(config)# router mpls
device(config-mpls)# vll-mtu-enforcement
```

Syntax: [no] vll-mtu-enforcement

By default, MTU checking is off. The user can use the [**no**] form of the command to disable VLL MTU checking when it is on.

> **NOTE**
> The user must save the configuration and reload the software for this command to take
> effect.

Specifying a VLL MTU

The user can use the **vll-mtu** command that allows the user to specify an MTU value per-VLL. When an MTU value is not specified for a VLL, the router continues to use the default max-frame-size for establishing the LDP session with the peer. The MTU value configured per-VLL can be changed dynamically and takes effect immediately. Consequently, the label (when already advertised) is withdrawn from the peer and re-advertised using the new MTU. This occurs irrespective of the state of the VLL. When MTU enforcement checks are enabled and when the MTUs don't match, the VLL stays down with the reason code "MTU mismatch".

> **NOTE**
> This parameter is not enforced on the data plane. Consequently, the user can still send and receive packets larger than the configured MTU.

> **NOTE**
> The global MTU must be changed before the command is
> effective.

To configure a new MTU value for a VLL, use the **vll-mtu** command as shown in the following.

```
device>enable
device# configure terminal
```

```
device(config)# router mpls
device(config-mpls)# vll foundry 40000
device(config-mpls-vll-foundry)# vll-mtu 1000
```

Syntax: [no] vll-mtu *mtu-value*

The *mtu-value* variable can be set to any value between 64 - 9190.

Generating traps for VLLs

The user can enable and disable SNMP traps for VLLs. VLL traps are enabled by default.

To enable VLL traps after they have been disabled, enter a command similar to the following:.

```
device(config)# snmp-server enable traps mpls vll
```

Syntax: [no] snmp-server enable traps mpls vll

Use the [no] form of the command to disable VLL traps.

Transparent forwarding of L2 and L3 protocols on a VLL for CES and CER

Use the **forward-all-protocol** command to add per port Layer2 and Layer3 (L2/L3) protocols ACL filters for the VLL end-point port. The command **no forward-all-protocol** removes the L2/L3 protocols ACL filters for the VLL end point port.

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

To implement per port Layer2 and Layer3 (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
   This command cannot be used on an MPLS interface as it will break existing neighbor relationships.

VLL extended counters

With the support of ingress and egress port VLAN counters on the MLX Series series 8x10G module, the port VLAN counters are enabled by default for all the *Layer 2 Virtual Private Network (L2VPN)* instances (VPLS, VLL, and Local VLL). As a result, the user can count the number of packets and bytes that are received and sent on a particular endpoint or all the endpoints of the L2VPN instances. The L2VPN extended counters can count all the types of packets including IPv4, IPv6, MPLS, MPLS over tagged, *Generic Routing Encapsulation (GRE)*, GRE over tagged, unicast, and multicast. The user can also count per-priority statistics on each endpoint by enabling per-VLAN, port, and priority-based accounting mode on the ingress and egress counters at the global configuration level.

   NOTE
   The extended counters for dual tag endpoints are not supported both on the ingress and egress ports.

The port VLAN counters are enabled by default for all the VLL instances. To disable the extended counters globally for all the VLL instances, enter the following command.

```
device(config-mpls)# vll-policy
device(config-mpls-vll-policy)# no extended-counters
```

Syntax: [no] extended-counters

When the extended counters are disabled globally, the user can enable the extended counters for a particular VLL instance by entering the following command.

```
device(config-mpls-vll-test10)# extended-counters on
```

Syntax: [no] extended-counters [on | off]

The **on** option enables extended counters for a particular VLL instance. The **off** option disables extended counters for a particular VLL instance.

Displaying VLL extended counters

When extended counters are enabled for a particular VLL instance either by default or explicit configuration, the user can display the number of bytes and packets received and sent on a particular endpoint or all the endpoints of that particular VLL instance. The counters are displayed whether or not the per-VLAN, port, and priority-based accounting mode is enabled at the global configuration level.

When the per-VLAN, port, and priority-based accounting mode is enabled at the global configuration level, the following output is displayed for the **show mpls statistics vll extended-counters** command.

```
device# show mpls statistics vll vll78 extended-counters
VLL vll78, VLL-ID 78: Extended Counters (only applicable for G2 modules)
VLAN   Port      RxPkts    TxPkts    RxBytes    TxBytes
74     5/2       0         0         0          0
       p0        0         0         0          0
       p1        0         0         0          0
       p2        0         0         0          0
       p3        0         0         0          0
       p4        0         0         0          0
       p5        0         0         0          0
       p6        0         0         0          0
       p7        0         0         0          0
```

When the per-VLAN, port, and priority-based accounting mode is disabled, the following output is displayed for the **show mpls statistics vll extended-counters** command.

```
device# show mpls statistics vll vll78 extended-counters
VLL vll78, VLL-ID 78: Extended Counters (only applicable for G2 modules)
VLAN   Port      RxPkts    TxPkts    RxBytes    TxBytes
74     5/2       0         0         0          0
```

Syntax: **show mpls statistics vll** [*vll-name* | *vll-id* [**extended-counters** [[**vlan** *vlan-id*] [**ethernet** *port-id*]]]]

The *vll-name* parameter specifies the configured name for a VLL instance.

The *vll-id* parameter specifies the ID of a VLL instance.

The **extended-counters** keyword enables the extended counters for a particular VLL instance.

The **vlan** *vlan-id* parameter specifies the ID of the configured VLAN.

The **ethernet** *port-id* parameter specifies the port ID of the interface for which the user wants to display the counters.

Table 35 describes the output parameters of the **show mpls statistics vll extended-counters** command.

TABLE 35 Output of the **show mpls statistics vll extended-counters** command

output field	Description
VLL	The configured name for a VLL instance.
VLL-ID	The ID of the VLL instance.
VLAN	The ID of the configured VLAN.
Port	The port ID of the interface for which the user wants to display the counters.
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.

TABLE 35 Output of the **show mpls statistics vll extended-counters** command (continued)

output field	Description
TxBytes	The number of bytes transmitted from the specified port.

Clearing VLL extended counters

To clear all the port VLAN counters for a particular VLL instance, enter a command similar to the following:

```
device# clear mpls statistics vll vll78 extended-counters
```

To clear all the port VLAN counters for a particular VLL instance and port under a specific VLL VLAN, enter the following command. This command is supported only for a single VLAN instance and is not supported for dual tag endpoints.

```
device# clear mpls statistics vll vll78 extended-counters vlan 74
```

To clear all the port VLAN counters for all the endpoints of a particular VLL instance, enter the following command. When the VLL endpoint is a *Link Aggregation Group (LAG)*, then the counters only for the given physical port are cleared.

```
device# clear mpls statistics vll vll78 extended-counters vlan 74 ethernet 5/2
```

To clear all the port VLAN counters for the given priority of a particular VLL endpoint, enter the following command.

```
device# clear mpls statistics vll vll78 extended-counters vlan 74 ethernet 5/2 p1
```

Syntax: clear mpls statistics vll [*vll-name* | *vll-id* [**extended-counters** [[**vlan** *vlan-id*] [**ethernet** *port-id* [**priority** *pri*]]]]]

The *vll-name* parameter specifies the configured VLL name for which the user wants to clear the counters.

The *vll-id* parameter specifies the ID of a VLL instance for which the user wants to clear the counters.

The **vlan** *vlan-id* parameter specifies the ID of the configured VLAN for which the user wants to clear the counters.

The **ethernet** *port-id* parameter specifies the port ID of the interface for which the user wants to clear the counters.

The **priority** *pri* parameter specifies a priority queue for a particular VLL endpoint for which the user wants to clear the counters.

MPLS VLL behavior with other features

This section describes the interaction of MPLS VLL with the features sFlow, IFL CAM, and Layer 2 ACLs.

sFlow

sFlow sampling is supported for VLL packets received from customer interfaces. sFlow is not supported for packets received from MPLS uplinks. The following describes the behavior for sFlow for VLL packets received from customer endpoints:

- When the endpoint is untagged, the default VLAN ID in the sFlow sample is used for both the incoming and outgoing VLAN fields in the sFlow sample collection.
- When the endpoint is tagged, the tag is used as both the incoming and outgoing VLAN in the sFlow sample collection.
- When the endpoint is dual-tagged, 4096 is used as the incoming VLAN to indicate that the packet is dual-tagged. When the VLL is in raw-mode, the outer VLAN of the packet is used as the outgoing VLAN ID in the sFlow sample collection. When the VLL is in tagged-mode, the inner VLAN ID of the packet is used as the outgoing VLAN ID in the sFlow sample collection.

Note that when the endpoint is dual-tagged, the sFlow packet does not contain VLL- or MPLS-specific information.

Table 36 illustrates the above points.

TABLE 36 Source and destination VLAN in an sFlow sampled VLL packet

Endpoint	Source VLAN	Destination VLAN
Untagged	Default VLAN	Default VLAN
Single-tagged	Incoming VLAN	Incoming VLAN
Dual-tagged	4096	Raw mode: Incoming outer VLAN
		Tagged mode: Incoming inner VLAN

IFL CAM

For dual-tagged VLL instances, IFL CAM entries are used for the service lookup. The default system value for IFL CAM is 8192, which can be modified up to a maximum of 81920 entries using the CLI command **system-max ifl-cam** *number*.

Layer 2 ACLs

When the port and VLAN combination of a Layer 2 ACL matches with any VLL endpoint, the ACL is applied. For dual-tagged VLL endpoints, the Layer 2 ACL is applied based on the port and outer VLAN combination, when it is configured.

Displaying MPLS VLL information

The user can display the following information about the MPLS VLL configuration on the device:

- Information about individual MPLS VLLs configured on the device
- Information about detailed MPLS VLLs configured on the device
- Information about LDP sessions between VLL peers
- Information about VLL Endpoint Statistics
- Information about packets sent between VLL endpoints and MPLS uplinks

Displaying information about MPLS VLLs

Use the following command to display information about MPLS VLLs.

```
device# show mpls vll
Name    VC-ID   Vll-peer     End-point                        State   Tunnel-LSP
test    10      10.11.11.11  tag vlan 2     e 1/10            UP      tn17
test2   100     --           tag vlan 100 inner-vlan 45  e2/1  DOWN
```

Syntax: show mpls vll

> **NOTE**
> Show commands have been enhanced to include the full MPLS name. Previously, the MPLS name was truncated because it exceeded the character length. Now, the MPLS name is text wrapped to display the full name.

For each MPLS VLL on the device, the following information is displayed.

TABLE 37 Output from the show mpls vll command

Output field	Description
Name	The configured name of the VLL.
VC-ID	The user-configurable ID as defined in draft-ietf-pwe3-control-protocol-14.txt.
Vll-peer	The remote PE router. This must be the same as the LSP destination for the LSPs that the VLL is transported over.
End-point	How packets are forwarded once they reach the egress LER. This can be one of the following: • "untagged" *portnum*" – Forward the packet out the specified port as untagged. • "tag" **vlan** *vlan-id / portnum*" – Tag the packet with the specified VLAN ID and forward the packet out the specified port. • "tag" **vlan** *vlan-id* **inner-vlan** *vlan-id* – Tag the packet with the specified outer and inner VLAN IDs and forward the packet out the specified port. • "undefined" – An endpoint has not been configured for this VLL.
State	The current state of the VLL. This can be either UP or DOWN. Data can be forwarded over the VLL only when the state is UP.
Tunnel-LSP	The name of the RSVP-signaled LSP that has been selected to carry the VLL traffic through the MPLS domain.

Displaying LDP information

To display information about the state of the LDP connection between the device and VLL peers, use the **show mpls ldp peer** command. For additional information on this CLI command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

For each established LDP session, use the command **show mpls ldp session** command. For additional information on this command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

To display information about LDP sessions between a specified router and VLL peers, use the **show mpls ldp session filtered** command. For additional information on this command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying VLL endpoint statistics

The user can display VLL Endpoint traffic statistics to see the forwarding counters for each VLL configured on the system. The display is shown so that, for a given port range that receives traffic, it shows the number of packets arriving from the customer endpoint and the number of packets arriving from the MPLS core and going to the customer interface.

To display all VLL traffic statistics on an Extreme device, use the **show mpls statistics vll** command. For additional information regarding this command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

> **NOTE**
> The VLL name is repeated for each module from where the statistics are collected, is displayed on the Management console.

To display VLL traffic statistics for a VLL instance specified by its VLL name, use the **show mpls statistics vll** *vll-name* command. For additional information regarding this command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

To display VLL traffic statistics for a VLL instance specified by its VLL ID, the **show mpls statistics vll** *vll_id* command. For additional information regarding this command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Clearing Local VLL traffic statistics

To clear all the statistics for all the Local VLL instances, enter a command similar to the following:

```
device# clear mpls statistics vll-local
```

To clear all the statistics for a particular Local VLL instance, enter a command similar to the following:

```
device# clear mpls statistics vll-local loc8
```

Syntax: clear mpls statistics vll-local [*vll-name* | *vll-id*]

The *vll-name* parameter specifies the configured name for a Local VLL instance.

The *vll-id* parameter specifies the ID of a Local VLL instance.

Sample MPLS VLL configuration

Figure 64 depicts a sample VLL configuration.

FIGURE 64 MPLS VLL configuration
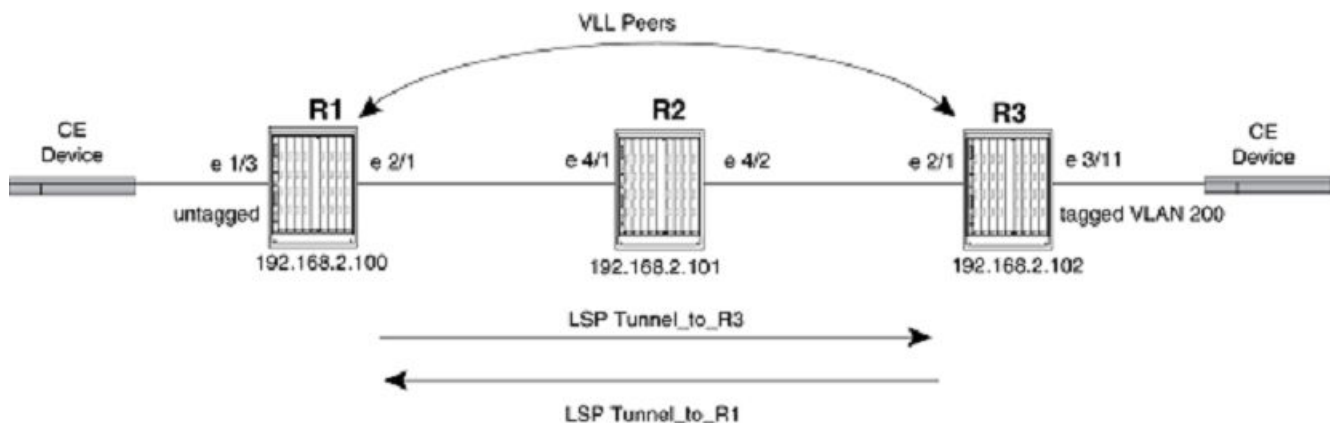
In this example, routers R1 and R3 are *Provider Edge (PE)* routers configured as VLL peers. R1 and R3 have established an LDP session to exchange VLL label information. When the LDP session is established, each router advertises its locally assigned VC label and VC ID to its VLL peer.

RSVP-signaled (tunnel) LSPs have been established in each direction between the two routers. When the CE device forwards a Layer 2 packet to R1, the router assigns the packet to an RSVP-signaled LSP whose destination is R3. R1 encapsulates the packet as an MPLS packet, adding a tunnel label and the VC label advertised to the router by R3. The MPLS packet is then forwarded over the outbound interface indicated by the tunnel label to the next hop in the LSP.

When the MPLS packet reaches R2, the penultimate LSR in the tunnel LSP, R2 pops the tunnel label, leaving the packet with only the VC label, then forwards the packet to R3.

R3 examines the VC label in the packet. On R3, the VC label is mapped to the user-specified endpoint for the VLL. In this example, the endpoint consists of VLAN ID 200 and interface 3/11. R3 then pops the VC label, tags the Layer 2 packet with VLAN 200, then forwards the packet out interface 3/11.

In the opposite direction, R3 assigns traffic received from the CE device to an RSVP-signaled LSP destined for R1, pushes tunnel and VC labels onto the packets, and forwards them to the next hop in LSP. When the packets reach R1, the router pops the VC label and

forwards the Layer 2 packets out the interface indicated by the VLL endpoint. In this example, the endpoint consists of interface 1/3, so the packets are forwarded untagged out interface 1/3 to the CE device.

Router device1

The following commands configure Router device1 in

```
device1(config)# ip router-id 192.168.2.100
device1(config)# router ospf
device1(config-ospf-router)# area 0
device1(config-ospf-router)# exit
device1(config)# router mpls
device1(config-mpls)# mpls-interface e 2/1
device1(config-mpls)# policy
device1(config-mpls-policy)# traffic-engineering ospf
device1(config-mpls-policy)# exit
device1(config-mpls)# lsp Tunnel_To_R3
device1(config-mpls-lsp)# to 192.168.2.102
device1(config-mpls-lsp)# enable
device1(config-mpls-lsp)# exit
device1(config-mpls)# vll VLL_to_R3 40000
device1(config-mpls-vll)# vll-peer 192.168.2.102
device1(config-mpls-vll)# untagged e 1/3
device1(config-mpls-vll)# exit
device1(config)# interface loopback 1
device1(config-lbif-1)# port-name Generic All-Purpose Loopback
device1(config-lbif-1)# ip address 192.168.2.100/32
device1(config-lbif-1)# ip ospf area 0
device1(config-lbif-1)# exit
device1(config)# interface e 1/3
device1(config-if-e100-1/3)# port-name VLL_endpoint
device1(config-if-e100-1/3)# enable
device1(config-if-e100-1/3)# exit
device1(config)# interface e 2/1
device1(config-if-e1000-2/1)# port-name Connection_to_R2
device1(config-if-e1000-2/1)# enable
device1(config-if-e1000-2/1)# ip address 192.168.37.1/30
device1(config-if-e1000-2/1)# ip ospf area 0
device1(config-if-e1000-2/1)# exit
```

Router device2

The following commands configure Router device2 in

```
device2(config)# ip router-id 192.168.2.101
device2(config)# router ospf
device2(config-ospf-router)# area 0
device2(config-ospf-router)# exit
device2(config)# router mpls
device2(config-mpls)# mpls-interface e 4/1 to e 4/2
device2(config-mpls)# policy
device2(config-mpls-policy)# traffic-engineering ospf
device2(config-mpls-policy)# exit
device2(config)# interface e 4/1
device2(config-if-e1000-4/1)# enable
device2(config-if-e1000-4/1)# ip address 192.168.37.2/30
device2(config-if-e1000-4/1)# ip ospf area 0
device2(config-if-e1000-4/1)# exit
device2(config)# interface e 4/2
device2(config-if-e1000-4/2)# enable
device2(config-if-e1000-4/2)# ip address 192.168.41.2/30
device2(config-if-e1000-4/2)# ip ospf area 0
device2(config-if-e1000-4/2)# exit
device2(config)# interface loopback 1
device2(config-lbif-1)# port-name Generic All-Purpose Loopback
device2(config-lbif-1)# ip address 192.168.2.101/32
```

```
device2(config-lbif-1)# ip ospf area 0
device2(config-lbif-1)# exit
```

Router device3

The following commands configure Router device3 in Sample MPLS VLL configuration on page 356.

```
device3(config)# ip router-id 192.168.2.102
device3(config)# router ospf
device3(config-ospf-router)# area 0
device3(config-ospf-router)# exit
device3(config)# router mpls
device3(config-mpls)# mpls-interface e 2/1
device3(config-mpls)# policy
device3(config-mpls-policy)# traffic-engineering ospf
device3(config-mpls-policy)# exit
device3(config-mpls)# lsp Tunnel_To_R1
device3(config-mpls-lsp)# to 192.168.2.100
device3(config-mpls-lsp)# enable
device3(config-mpls-lsp)# exit
device3(config-mpls)# vll VLL_to_R1 40000
device3(config-mpls-vll)# vll-peer 192.168.2.100
device3(config-mpls-vll)# vlan 200
device3(config-mpls-vll-vlan)# tagged e 3/11
device3(config-mpls-vll-vlan)# exit
device3(config-mpls-vll)# exit
device3(config)# interface loopback 1
device3(config-lbif-1)# port-name Generic All-Purpose Loopback
device3(config-lbif-1)# ip address 192.168.2.102/32
device3(config-lbif-1)# ip ospf area 0
device3(config-lbif-1)# exit
device3(config)# interface e 3/11
device3(config-if-e100-3/11)# port-name VLL_endpoint
device3(config-if-e100-3/11)# enable
device3(config-if-e100-3/11)# exit
device3(config)# interface e 2/1
device3(config-if-e1000-2/1)# port-name Connection_to_R2
device3(config-if-e1000-2/1)# enable
device3(config-if-e1000-2/1)# ip address 192.168.41.1/30
device3(config-if-e1000-2/1)# ip ospf area 0
device3(config-if-e1000-2/1)# exit
```

Local VLL

Local VLL is used to create a *Virtual Leased Line (VLL)* circuit with endpoints in the same Extreme device. A Local VLL can be configured between two ports in a router, two LAGs in a router or between a port and a LAG as shown in Figure 65. Each entity (port or LAG) is identified as either "Endpoint 1" or Endpoint 2".

> NOTE
> LAGs supported include server LAGs and per-packet server LAGs. LACP LAGs are not supported.

FIGURE 65 Local VLL port and LAG configurations
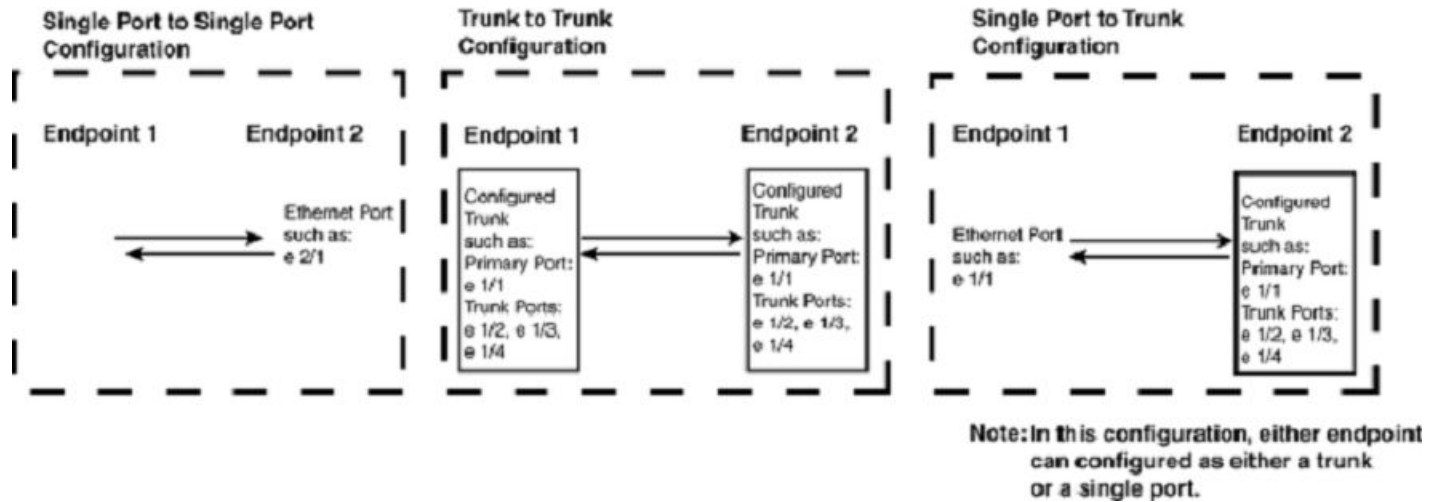
> **NOTE**
> When configuring a LAG as an endpoint, only the primary port of the LAG is specified in the Local VLL configuration.

> **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.

The endpoints connected to the Local VLL can be untagged or tagged as members of the same or different VLANs. Using this function of Local VLL, a router can receive packets with a particular tag or no tag on one endpoint and forward them to the Local VLLs other endpoint which may be untagged or tagged with a different VLAN tag. When so configured, the tags within the packets are changed to reflect the configuration of the egress port as they leave the router.

Local VLL configuration examples

Local VLL supports traffic flows between any combination of single-tagged, untagged, and dual-tagged ports. Some scenarios are described and illustrated in the following configuration examples.

Example of a Local VLL configured for single-tagged VLAN traffic on both ports

In Figure 66 the Local VLL named "Test1" contains Ethernet ports 1/1 and 2/1. Port 1/1 is a member of VLAN 100 and port 2/1 is a member of VLAN 200. Because both ports belong to Local VLL "Test1" traffic tagged with VLAN 100 is able to reach nodes within VLAN 200 and traffic tagged with VLAN 200 is able to reach nodes within VLAN 100. Traffic that ingresses on port 1/1 must have a tag with the value "100" and egresses on port 2/1 with a tag value of "200". Traffic that ingresses on port 2/1 must have a tag with the value "200" and egresses on port 2/1 with a tag value of "100".

FIGURE 66 Local VLL "Test1" with two tagged VLANs


```
device(config)# router mpls
device(config-mpls)# vll-local test1
device(config-mpls-vll-lo-test1)# vlan 100
device(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/1
device(config-mpls-vll-lo-test1-vlan)# vlan 200
device(config-mpls-vll-lo-test1-vlan)# tagged ethernet 2/1
```

Example of a Local VLL configured for dual-tagged and single-tagged VLAN traffic

The user can configure a Local VLL with ports that are configured for dual tags. In a dual-tagged configuration, the packets contain an outer tag and an inner tag. One or both of the VLANs configured for the Local VLL has an inner VLAN configured in addition to the default outer VLAN.

Under this configuration, a router can receive packets with a two tags on one endpoint and forward them to the Local VLLs other endpoint either untagged, tagged with a single tag or tagged with an inner and outer VLAN tag. Where dual-tagging is used within a Local VLL, the system allocates an *Internal Lookup Identifier (IFL-ID)* for the Local VLL instance.

In Figure 67 the Local VLL named "Test2" contains Ethernet ports 1/1 and 2/1. Port 1/1 is a member of VLAN 100 and is configured to accept packets with an inner VLAN tag value of 300. Port 2/1 is a member of VLAN 200. Because both ports belong to Local VLL "Test2" traffic tagged with outer VLAN tag 100 and inner VLAN tag 300 is able to reach nodes within VLAN 200 and traffic tagged with VLAN 200 is able to reach nodes within VLAN 100. Traffic that ingresses on port 1/1 must have an outer tag with the value "100" and an inner tag with the value "300" and egresses on port 2/1 with a tag value of "200". Traffic that ingresses on port 2/1 must have a tag with the value "200" and egresses on port 1/1 with an inner tag value of "100" and an outer tag value of "300".

FIGURE 67 Local VLL "Test2" with one single-tagged VLAN and one dual-tagged VLAN


```
device(config)# system-max ifl-cam 16384
device(config)# router mpls
device(config-mpls)# vll-local test2
```

```
device(config-mpls-vll-lo-test1)# vlan 100 inner-vlan 300
device(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/1
device(config-mpls-vll-lo-test1-vlan)# vlan 200
device(config-mpls-vll-lo-test1-vlan)# tagged ethernet 2/1
```

A shown in the following example, the user can use the **show mpls vll-local detail** command to see that an IFL-ID has been created for this Local VLL instance.

```
device# show mpls vll-local detail
VLL test2 VLL-ID 1 IFL-ID 4096
  State: UP
  End-point 1:     tagged  vlan 100   inner-vlan 300    e 1/1
                       COS: --
  End-point 2:     tagged  vlan 200   e 2/1
                   COS: --
```

Example of a Local VLL configured for single-tagged and untagged VLAN traffic

In Figure 68 the Local VLL named "Test3" contains Ethernet ports 1/1 and 2/1. Port 1/1 is a member of VLAN 100 and port 2/1 is untagged. Because both ports belong to Local VLL "Test3, traffic tagged with VLAN 100 is able to reach nodes attached to the untagged port and traffic from the untagged port is able to reach nodes within VLAN 100.

FIGURE 68 Local VLL "Test3" with one tagged VLAN and one untagged port
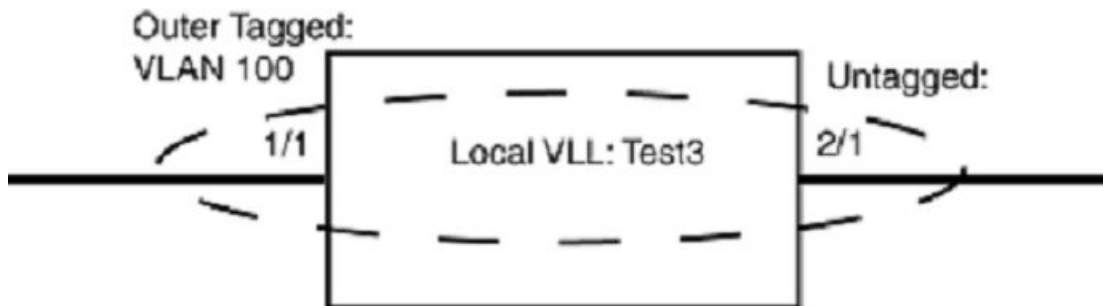

```
device(config)# router mpls
device(config-mpls)# vll-local test3
device(config-mpls)# untagged ethernet 2/1
device(config-mpls-vll-lo-test1)# vlan 100
device(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/1
```

Local VLL QoS

The user can optionally specify *Class of Service (CoS)* on a *per-endpoint (EP)* basis. This CoS value applies to inbound traffic on the endpoint. When a CoS value is not specified, the port's configured priority and the packet's 802.1p priority are used to determine the internal priority, as described for the following traffic flows.

Untagged Endpoint 1 (EP1) to Untagged Endpoint 2 (EP2).

1. When available, use the configured CoS value on untagged EP1 otherwise go to step 2.

2. When there is a configured port priority on untagged EP1 use that priority; otherwise go to step 3.

3. When there is neither a CoS value or priority configured (as described in steps 1 and 2), the default "best effort" priority is used.
    Tagged Endpoint 1 (EP1) to Tagged Endpoint 2 (EP2).

CoS behavior for Local VLL

> **NOTE**
> This section assumes that the user understands how QoS works.

Table 38 describes the expected *Class of Service (CoS)* behavior for VLL packets when Local VLL is in effect.

TABLE 38 Expected Class of Service behavior for Local VLL

Local VLL endpoints	Incoming packet		Outgoing packet	
Outer VLAN	Inner VLAN	Outer VLAN	Inner VLAN	
Dual-tagged to dual-tagged	X	Y	X' or X	Y
Single-tagged to dual-tagged	X	N/A	X' or X	X
Untagged to dual-tagged	N/A	N/A	X' or 0	0
Dual-tagged to single-tagged	X	Y	X' or Y	N/A

X = Original outer VLAN CoS.

Y = Original inner VLAN CoS.

X' = Mapped CoS from internal priority (**X** contributes to internal priority) using CoS encode table.

Configuring Local VLL

Configuring Local VLL uses the following procedures:

- Local VLL configuration on page 362
- Specifying Local VLL endpoints on page 362
- Configuring Local VLL QoS (optional) on page 364

Local VLL configuration

Local VLL is configured under router mpls as shown.

```
device(config)# router mpls
device(config-mpls)# vll-local test1
```

Syntax: [no] vll-local *vll-name*

Specifying Local VLL endpoints

Local VLL can be configured between any combination of untagged, single-tagged, and dual-tagged endpoints.

The following sections describe how to configure VLL Endpoints:

Configuring an untagged endpoint

To configure untagged port 1/1 into Local VLL instance "test1" use the following commands:

```
device(config)# router mpls
device(config-mpls)# vll-local test1
device(config-mpls-vll-test1)# untagged ethernet 1/1
```

Syntax: [no] untagged ethernet *slot/port*

Configuring a single-tagged endpoint

Tagged ports are configured under a VLAN ID. This ID is only meaningful for the tagged port.

For tagged ports, a *vlan-id port* pair constitutes a VLL endpoint. When a port is currently a member of a non-default VLAN as an untagged port, it must be returned to the default VLAN before it can be assigned to a VLL as a tagged port.

To configure tagged port 1/2 with VLAN 200 into Local VLL instance "test1" use the following commands.

```
device(config)# router mpls
device(config-mpls)# vll-local test1
device(config-mpls-vll-lo-test1)# vlan 200
device(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/2
```

Syntax: vlan *VLAN-ID*

The range for *VLAN ID* is 1 - 4094. (This parameter range excludes the default VLAN ID.)

Syntax: [no] tagged ethernet *slot/port*

Configuring a dual-tagged endpoint

Dual-tagged ports are configured under a VLAN ID. The main difference between single and dual tagged configuration is that in the dual-tagged configuration, a parameter for **inner-vlan** is configured.

Considerations when configuring the Local VLL with dual tagged endpoints

Before configuring a Local VLL to operate with Dual Tagged Endpoints, the user must consider the following:

- The System Max value for IFL CAM must be changed from its default value of zero (which does not support this feature) to a higher value.
- Only one dual-tag endpoint can exist on the same port per instance.
- The inner VLAN of the dual-tag endpoint cannot be configured dynamically. In other words, an existing single-tag VLL endpoint cannot be changed to a dual tag VLL endpoint on the fly. The user must delete the single-tag VLL endpoint before configuring a dual-tag endpoint.
- A dual tag VLL endpoint neither recognizes nor forwards packets that have a single tag. However, a single-tag endpoint can recognize and forward dual tag packets because the endpoint treats the second tag as data.
- When only the outer VLAN is specified for a given endpoint, the VLAN is called a less-specific VLAN. Similarly, when both the outer and inner VLANs are specified, the VLAN is called a more-specific VLAN (in relation to the outer VLAN).
- When a less-specific VLAN is already configured on a given port, then a more-specific VLAN with the same outer VLAN tag can be configured on that port. In the following example, a less-specific, tagged endpoint has been configured with VLAN 100 on port e 2/1, and a more-specific endpoint with outer-VLAN value of "100" and an inner-VLAN value of "200" is configured on port e 2/1.

```
device(config-mpls)# vll-local test1
device(config-mpls-vll-lo-test1)# vlan 100
device(config-mpls-vll-lo-test1-vlan)# tag e 2/1
device(config-mpls-vll-lo-test1-if-e-2/1)# vlan 100 inner-vlan 200
device(config-mpls-vll-lo-test1-vlan)# tag e 2/1
device(config-mpls-vll-lo-test1-vlan)#
```

The result of this example is that single-tagged packets received on port 2/1 with VLAN ID value of "100" and double-tagged packets with an outer-VLAN value of "100" and inner-VLAN of any value other than "200" are sent back out from port 2/1 with outer-VLAN

value of "100" and an inner-VLAN value of "200". Dual-tagged packets received on port 2/1 with an outer-VLAN value of "100" and an inner-VLAN value of "200" are sent back out from port 2/1 as single-tagged packets with a VLAN value of "100".

- In any given Local VLL instance, two dual-tag endpoints on the same port are not allowed. The Error messages displayed in **bold** in the following two configuration examples describe this restriction.

```
device(config-mpls)# vll-local test3
device(config-mpls-vll-lo-test3)# vlan 100 inner-vlan 400
device(config-mpls-vll-lo-test3-vlan)# tag e 2/3
device(config-mpls-vll-lo-test3-if-e-2/3)# vlan 100 inner-vlan 500
device(config-mpls-vll-lo-test3-vlan)# tag e 2/3
Error - VLL test3 already has a dual tag end-point on port 2/3 - another dual tag endpoint on the same port
not allowed.
device(config-mpls)# vll-local test4
device(config-mpls-vll-lo-test3)# vlan 1000 inner-vlan 400
device(config-mpls-vll-lo-test3-vlan)# tag e 2/3
device(config-mpls-vll-lo-test3-if-e-2/3)# vlan 2000 inner-vlan 500
device(config-mpls-vll-lo-test3-vlan)# tag e 2/3
Error - VLL test4 already has a dual tag end-point on port 2/3 - another dual tag endpoint on the same port
not allowed.
```

To support dual tags, the VLAN CLI command in the Local VLL configuration mode has a parameter that lets the user configure dual tag endpoints.

```
device(config)# router mpls
device(config-mpls)# vll-local test1
device(config-mpls-vll-lo-test1)# vlan 200 inner-vlan 300
device(config-mpls-vll-lo-test1-vlan)# tagged ethernet 1/2
```

Syntax: [no] vlan *VLAN-ID* inner-vlan *Inner-VLAN-ID*

The range for *VLAN-ID* is 1 - 4094. (This parameter range excludes the default VLAN ID.)

The range for *inner-VLAN-ID* is 1 - 4095. (This parameter range does not exclude the default VLAN ID.)

Configuring Local VLL QoS (optional)

The user can configure a *Class of Service (CoS)* value for either a tagged or untagged port. when the CoS value is configured, it is used to determine traffic priority as described in .

To set a CoS value for an untagged port use the following command.

```
device(config)# router mpls
device(config-mpls)# vll-local test1
device(config-mpls-vll-test1)# untagged ethernet 1/1
device(config-mpls-if-e1000-1/1)# CoS 3
```

To set a CoS value for an tagged port, use a command similar to the following:

```
device(config)# router mpls
device(config-mpls)# vll-local test1
device(config-mpls-vll-test1)# vlan 200
device(config-mpls-vll-vlan)# tagged 1/2
device(config-mpls-if-e1000-1/2)# CoS 4
```

Syntax: [no] cos *cos-value*

The *cos-value* can be set to a priority between 0 - 7.

Local VLL extended counters

With the support of ingress and egress port VLAN counters on the MLX Series series 8x10G module, the port VLAN counters are enabled by default for all Local VLL instances.

The user can disable the extended counter functionality globally for all the Local VLL instances by entering a command similar to the following:

```
device(config-mpls)# vll-local-policy
device(config-mpls-vll-local-policy)# no extended-counters
```

Syntax: [no] extended-counters

When the extended counters are disabled globally, the user can enable the extended counters for a particular Local VLL instance to display the number of bytes and packets received and sent on a particular endpoint or all the endpoints of that Local VLL instance.

To enable the extended counters for a particular Local VLL instance, enter a command similar to the following:

```
device(config-mpls-vll-local-test10)# extended-counters on
```

Syntax: [no] extended-counters [on | off]

The **on** option enables extended counters for a particular Local VLL instance. The **off** option disables extended counters for a particular Local VLL instance.

Displaying Local VLL extended counters

When extended counters are enabled for a particular Local VLL instance either by default or explicit configuration, the user can display the number of bytes and packets received and sent on a particular endpoint or all the endpoints of that Local VLL instance. The counters are displayed whether or not the per-VLAN, port, and priority-based accounting mode is enabled at the global configuration level.

For additional information, go to the **show mpls statistics vll-local** CLI command in *NetIron Command Line Interface (CLI) Reference Guide*.

Clearing Local VLL extended counters

To clear all the port VLAN counters for a particular Local VLL instance, enter a command similar to the following:

```
device# clear mpls statistics vll-local loc8 extended-counters
```

To clear all the port VLAN counters for a particular Local VLL instance and port under a specific Local VLL VLAN, enter the following command. This command is supported only for a single VLAN instance and is not supported for dual tag endpoints.

```
device# clear mpls statistics vll loc8 extended-counters vlan 94
```

To clear all the port VLAN counters for all the endpoints of a particular Local VLL instance, enter a command similar to the following:

```
device# clear mpls statistics vll loc8 extended-counters vlan 94 ethernet 5/2
```

To clear all the port VLAN counters for the given priority of a particular Local VLL endpoint, enter a command similar to the following:

```
device# clear mpls statistics vll loc8 extended-counters vlan 94 ethernet 5/2 p1
```

Syntax: clear mpls statistics vll-local [*vll-name* | *vll-id* [**extended-counters** [[**vlan** *vlan-id*] [**ethernet** *port-id* [**priority** *pri*]]]]]

The *vll-name* parameter specifies the configured Local VLL name for which the user wants to clear the counters.

The *vll-id* parameter specifies the ID of a Local VLL instance for which the user wants to clear the counters.

The **vlan** *vlan-id* parameter specifies the ID of the configured VLAN for which the user wants to clear the counters.

The **ethernet** *port-id* parameter specifies the port ID of the interface for which the user wants to clear the counters.

The **priority** *pri* parameter specifies a priority queue for a particular Local VLL endpoint for which the user wants to clear the counters.

Displaying Local VLL information

The user can display the following information about the Local VLL configuration on a **show mpls vll** device:

*   Information about individual Local VLLs configured on the router
*   Information about VLL Endpoint Statistics

Displaying information about Local VLLs

To display brief information about Local VLLs use the **show mpls vll-local** command. Addition information regarding this command is located in *NetIron Command Line Interface (CLI) Reference Guide* .

To display detailed information about a specific Local VLL configured on the device, see *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying Local VLL endpoint statistics

To view the forwarding counters for each Local VLL configured on the system, the user can display Local VLL Endpoint traffic statistics. The display is shown such that for a given port range that receives traffic, how many packets are arriving from the customer endpoint.

> **NOTE**
> When the forwarding cam is full, the vll-local software forwarded packets are not accounted in vll-local statistics.

To display all VLL traffic statistics on an Extreme device, enter a command similar to the following:

```
device# show mpls stat vll-local
VLL-Name      End-point 1/2    VLL Port(s)    VLL-Ingress-Pkts
--------      -------------    -----------    ----------------
test          End-point1       e2/3-2/4       835192705
              End-point2       e2/3-2/4       838181595
test1         End-point1       e2/3-2/4       544017
              End-point2       e2/3-2/4       544017
test3         End-point1       e2/1           544022
              End-point2       e2/2           544022
```

To display VLL traffic statistics for a VLL instance specified by its VLL name, enter a command similar to the following:

```
device# show mpls stat vll-local test
VLL-Name      End-point 1/2    VLL Port(s)    VLL-Ingress-Pkts
--------      -------------    -----------    ----------------
test          End-point1       e2/3-2/4       0
              End-point2       e2/3-2/4       0
```

To display Local VLL traffic statistics for a VLL instance specified by its VLL ID, enter a command similar to the following:

```
device# show mpls stat vll-local 4
VLL-Name      End-point 1/2    VLL Port(s)    VLL-Ingress-Pkts
--------      -------------    -----------    ----------------
test3         End-point1       e2/1           0
              End-point2       e2/2           0
```

Syntax: **show mpls statistics vll-local** [*vll-name* | *vll-id*]

The *vll-name* variable is the configured name for a Local VLL instance. The *vll-id* variable is the ID of a VLL instance.

The following information is displayed in the **show mpls statistics vll** command:

TABLE 39 Output from the show mpls vll-local command

Output field	Description
VLL-Name	The configured name of the Local VLL instance.
End-point 1/2	Either the End-point1 or End-point2 of the Local VLL instance.
VLL Ports	The port or ports that are assigned to the end point. When there are multiple ports, they are members of a trunk.
VLL-Ingress-Pkts	Packets arriving on the specified end point from outside the Local VLL.

Clearing the VLL traffic statistics

To clear the entries stored for all Local VLL statistics, enter a command similar to the following:

```
device# clear mpls statistics vll-local
```

Syntax: **clear mpls statistics vll-local**

Enabling MPLS Local VLL traps

The user can enable trap notification to be sent for Local VLLs by entering a command similar to the following:

```
device(config)# snmp-server enable trap mpls vll-local
```

Syntax: **[no] snmp-server enable trap mpls vll-local**

Refer to the *Unified IP MIB Reference* for MPLS VLL trap notifications.

Disabling Syslog messages for MPLS VLL-local and VLL

Transitions of VLL local instances from an up state to a down state and vice versa are logged by default. The user can disable the logging of these events by entering a command similar to the following:

```
device(config)# no logging enable mpls
```

Syntax: **[no] logging enable mpls**

Similarly, the generation of Syslog message for MPLS VLL events are enabled by default. The user can disable the logging of these event by entering a command similar to the following:

```
device(config)# no logging enable mpls vll
```

Syntax: **[no] logging enable mpls vll**

VLL raw-pass-through overview

The raw-mode and tagged-mode supports are for both CES 2000 Series and XMR Series platforms. In the raw-pass-through mode, VLL instance behaves similar to either tagged-mode or raw-mode based on the VLL endpoint configuration and similar to tagged-mode for a tagged endpoint and raw-mode for an untagged endpoint.

Packet formats and VC mode definitions:

The expected raw-mode packet and tagged-mode packet formats are shown below.

Expected Raw Mode Packet format:

| RDA | RSA | Etype | Label(s) | C-DA | C-SA | Etype (2B) | IP header and payload |

Expected Tagged Mode Packet format:

| RDA | RSA | Etype | Label(s) | C-DA | C-SA | Etype (2B) | Payload Tag | Etype | IP header and payload |

Packet handling behavior

Depending on the type of endpoints configured on the VLL instance, VLL instance has the packet processing behavior listed in the table below.

TABLE 40 Packet tag insertion and stripping decision

Local endpoint type	Packet received from local endpoint destined towards remote peer (MPLS uplink).	Packet received from local remote peer (MPLS uplink) destined towards local endpoint.
Untagged endpoint	No additional tag is inserted in the packet. Behaves the same as the "raw-mode".	No tag stripping from the received MPLS uplink packet. Behaves the same as the "raw-mode".
Tagged endpoint	Insert the received VLAN Tag into the MPLS uplink packet with the PCP value preserved. Behaves the same as the "tagged-mode".	Strip out the tag from the received MPLS uplink packet. Behaves the same as the "tagged-mode".

The following table reviews the VC mode definitions.

TABLE 41 VC mode definitions

VC Mode	PW VC type signaled to remote peer through LDP
Raw-mode	0x5 (Ethernet-type)
Tagged-mode	0x4 (Ethernet tagged)
Raw-pass-through	0x5 (Ethernet-type) - for untagged endpoint
	0x4 (Ethernet tagged) - for tagged endpoint

The following figure describes the behavior with "raw-pass-through-mode".
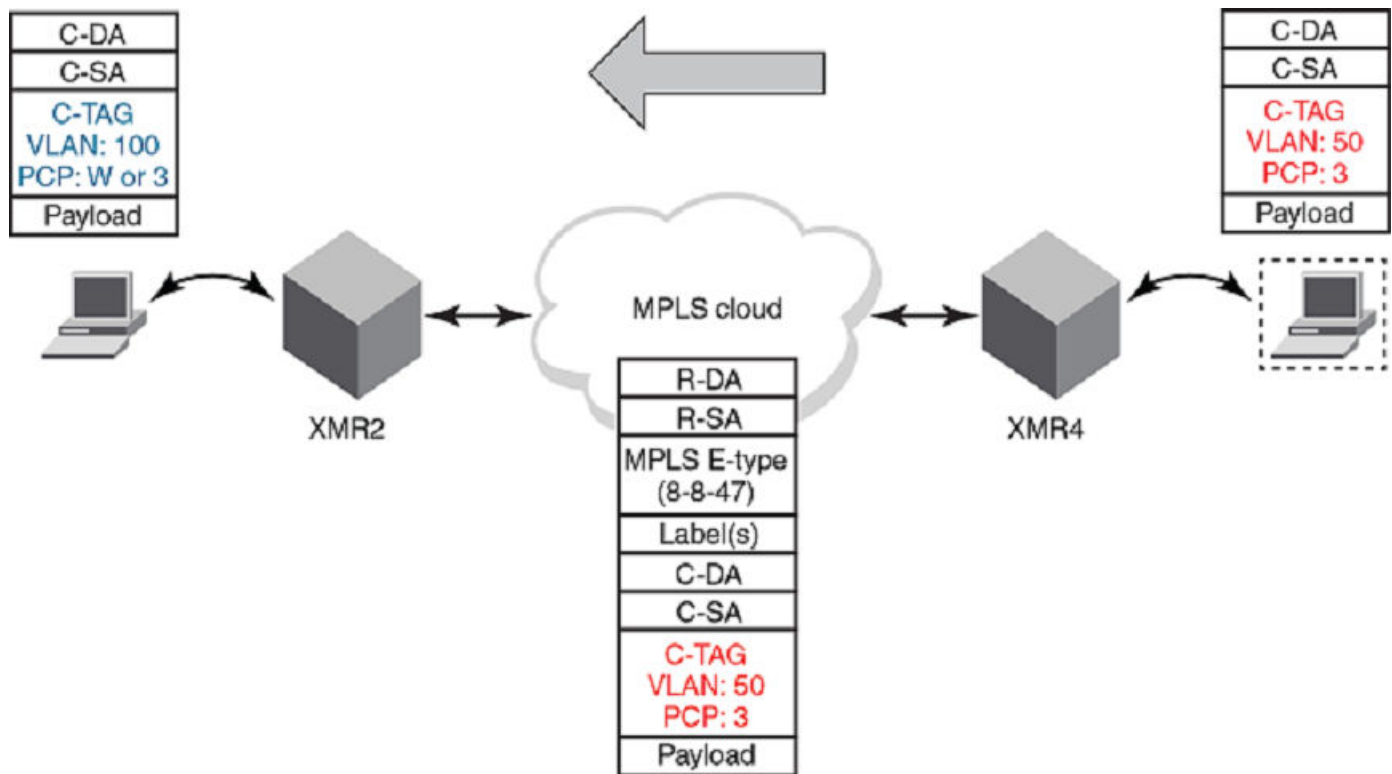
Backward compatibility

This feature is backward compatible.

Upgrade and downgrade considerations

When deploying this feature, follow the standard upgrade procedure for the XMR Series and MLX Series platforms.

Scaling support

There are no changes to scaling numbers.

Customer requirements

- Configured local endpoint as untagged
    - Pass the received tags ASIS from local endpoint towards remote peer.
    - Treat payload tag received from MPLS Uplink (remote peer) as part of the payload and do not take its PCP into account for QoS decision making.
- Configured local endpoint as tagged
    - Pass the received tags ASIS from local endpoint towards remote peer.
    - Strip the payload tag received from MPLS uplink (remote peer) but preserve its PCP value and take into account for QoS decision making based on customer configuration.

VLL mapping to specific LSPs

The VLL mapping to specific LSPs feature allows the user to assign specific LSPs to VLL and configure up to eight RSVP-TE LSPs for the VLL peer.

Supporting hardware

Hardware requirements for the implementation of this feature.

TABLE 42 Supporting hardware

XMR Series	MLX Series	CES 2000 Series BASE package	CES 2000 Series ME_PREM package	CES 2000 Series ME_PREM package	CER 2000 Series BASE package	CER 2000 Series Advanced Services package
Y	Y	N	Y	N	N	Y

> **NOTE**
> Does not support Gen-1 interface cards.

Feature specification

The feature allows the user to configure up to eight RSVP-TE LSPs for the VLL peer. The VLL then selects the first operationally UP LSP from the configured list. The remaining LSPs are used in the event of the first chosen LSP goes down or is unconfigured. If none of the configured LSPs are operationally UP, the VLL is treated as down.

> **NOTE**
> There is no priority among the LSPs that are configured. The algorithm followed is "use the first operationally UP LSP from the assigned LSP list". For example, if four LSPs, lsp1, lsp2, lsp3, and lsp4 are assigned to a VLL, all LSPs are operationally UP. In this case, lsp1 is chosen, and in the event of lsp1 going down, the first operationally UP LSP from remaining LSPs (lsp2, lsp3 ,lsp4) is chosen. If lsp1 comes back UP, the VLL is not moved back to lsp1. It continues to use the LSP that was previously chosen as the replacement.

Glossary of terms

TABLE 43 Glossary of terms

Term	Meaning
VLL	Virtual Leased Line
VPLS	Virtual Private LAN Services
LSP	Label Switched Path
MCT	Multi-Chassis Trunking
MCT-VLL	Multi-Chassis Trunking end-points for Virtual Leased Line
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
LDP	Label Distribution Protocol
PW	Pseudo Wire
CoS	Class of Service

Limitations and pre-requisites

- There is no LSP assignment support for MCT spoke PW.
- The RSVP LSP target address must match the VLL peer address or the configuration is rejected.
- The VLL CoS configuration is ignored when used with the *VLL mapping to specific LSPs* feature.
- The LDP LSPs cannot be mapped to a VLL. (LDP LSPs do not have a name association.)

Upgrade and downgrade considerations

Upgrade information

The user will not see any behavioral change after the upgrade (example, VLL will use any UP LSP towards the VLL peer).

When the user upgrades to a newer version which supports LSP mapping to VLL, the user must issue the additional configuration command **vll-peer** *ip_address* **lsp** lasp1 lsp2...

However, the **show running config** command will only show the minimal configuration.

Example:

```
device (config-mpls)#show mpls config vll
vll test 45000
  vll-peer 11.11.11.11 lsp lsp1 lsp2 lsp3 lsp4
  vlan 1000
   tagged e 4/5
```

Feature downgrade

This feature is not available when downgraded.

Customer use scenarios for VLL mapping to specific LSPs

The following figure represents the network topology that is referenced throughout this section.
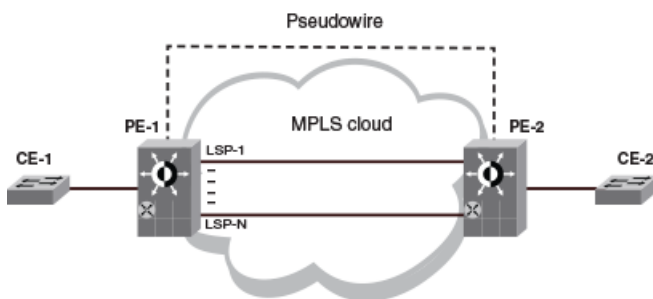
Customer use scenario 1: VLL mapped to a set of LSPs.

Pre-condition	Multiple RSVP LSPs are setup between the VLL Peers.
Trigger	User assigns a set of RSVP LSPs to VLL.
Basic Flow	VLL module creates an VLL using the first operationally UP LSP. The command **show mpls vll detail** displays if LSPs are mapped and the selected LSP.

Alternate Flow	• LSPs are mapped to a VLL after the VLL becomes operational, the VLL is moved from the LSP selected by default to first assigned operating UP LSP. • All assigned LSPs are already DOWN, VLL does not come UP. • A new LSP is appended to list of LSPs already assigned to a VLL, and added to the list. There is no update to the forwarding module and no traffic impact.

Customer use scenario 2: VLL mapped to a set of LSPs after the VLL was UP.

Pre-condition	Multiple RSVP LSPs are setup between the VLL Peers.
Trigger	User configures a VLL, and it comes UP. User now assigns a set of LSPs to VLL.
Basic Flow	The VLL which was using a LSP selected by default (LDP/RSVP/CoS based) is moved to one of the new assigned LSP (if at least one assigned LSP is UP). No new VC label signaling takes place. The new LSP information is updated to the forwarding module. This causes minimal traffic impact.
Alternate Flow	All assigned LSPs are already DOWN, VLL is brought DOWN.

Customer use scenario 3: Un-configure LSP assigned to a VLL from the assigned LSP list.

Pre-condition	Multiple RSVP LSPs are setup between the VLL Peers. These RSVP LSPs are assigned to the VLL.
Trigger	User un-configures the LSP being used by VLL from the VLLs assigned LSP list.
Basic Flow	VLL module picks the next operationally UP LSP from the configured list. When none of the configured LSPs are operationally UP, the VLL is brought DOWN.
Alternate Flow	• Un-configure LSP from the assigned LSP list which is not in use by the VLL. No update to the forwarding module. No traffic impact. • Un-configure all LSPs on the assigned list. A new LSP is selected (based on CoS, if configured).

Customer use scenario 4: Assigned LSP goes DOWN.

Pre-condition	Multiple RSVP LSPs are setup between the VLL Peers. User assigns a set of RSVP LSPs to VLL.
Trigger	The interface used by assigned LSP goes DOWN.
Basic Flow	VLL module picks the next operationally UP LSP from the configured list. When none of the configured LSPs are operationally UP, the VLL is brought DOWN. Traffic impact: Yes
Alternate Flow	• When there is a single RSVP LSP assigned to a VLL, then the VLL is brought DOWN a soon as the LSP goes DOWN. • Not in use LSP from the assigned LSP list goes DOWN. No effect on VLL status.

Customer use scenario 5: Assigned LSP from the list comes UP.

Pre-condition	• Multiple RSVP LSPs are setup between the VLL Peers. • User assigns a set of RSVP LSPs to VLL. • The first RSVP LSP from the assigned list is DOWN. • The VLL is UP using another LSP from the assigned list.
Trigger	The first RSVP LSP from the assigned list comes UP.
Basic Flow	The VLL module updates the LSP operational status. There is no re-programing involving the LSP that just came UP. Traffic impact: None.
Alternate Flow	-

Customer use scenario 6: MPLS updates to assigned LSP.

Pre-condition	• Multiple RSVP LSPs are setup between the VLL Peers. • User assigns a set of RSVP LSPs to VLL. • VLL is UP using the first operationally UP LSP from the assigned list.
Trigger	Nexthop change affecting RSVP LSP in use by VLL.
Basic Flow	The VLL module updates the LSP based on the updates from MPLS module and programs forwarding accordingly.
Alternate Flow	• MPLS updates like switchover to FRR, primary to secondary fallback, adaptive LSP update. All such updates cause minimal traffic loss due to re-programming in the forwarding module. • MPLS updates to LSP which are configured, but not used, by the VLL. Updates saved locally in the VLL module.

Customer use scenario 7: Assigning LSP to VLL standby or backup peer in MCT and PW redundancy respective scenarios.

Pre-condition	Multiple RSVP LSPs are setup between the both Standby, Backup and Active VLL Peers.
Trigger	User assigns a set of RSVP LSPs to Standby and Backup VLL Peer.
Basic Flow	VLL is UP, using the first operationally UP LSP from the assigned list. Use the **show mpls vll detail** command to see the LSP that is being used. Refer to the **show mpls vll** command for additional information.
Alternate Flow	Scenarios from Customer use scenario 2 and Customer use scenario 5 are applicable here.

L2 VLL over IPsec

To enhance security, VLL traffic can be sent over IPsec.

> **NOTE**
> The encryption of Layer 2 (L2) VLL traffic over IPsec is only supported on the 10GX4-IPSEC-M module interface card.

FIGURE 69 L2 VLL traffic over IPsec
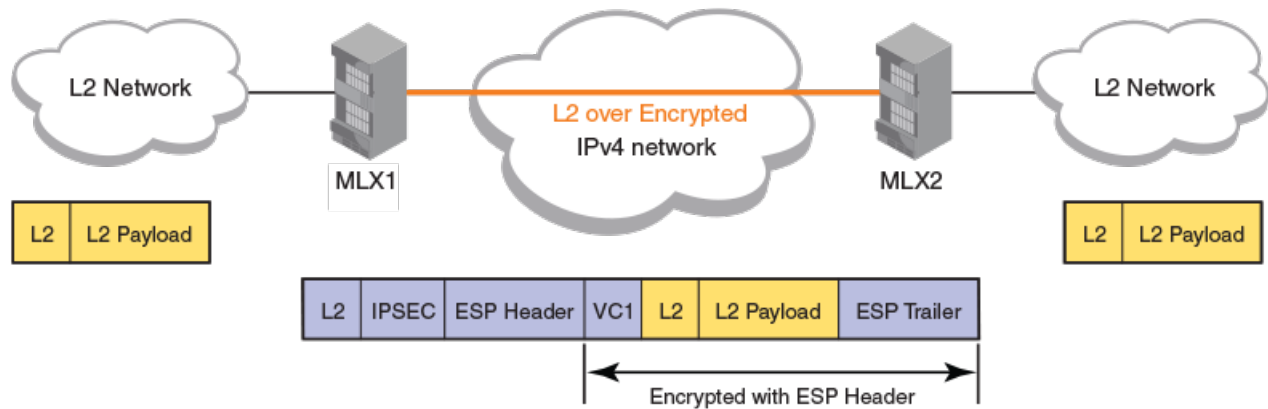
The preceding figure shows how L2 VLL traffic is encrypted and encapsulated by using an LDP tunnel over IPsec.

At MLX1, an L2 VLL packet comes in from a Layer 2 network. MPLS processing then adds a VC label to the original Layer 2 packet. VC label encapsulation enables an IPsec tunnel to carry traffic from multiple customers; the VC label is used to translate the original customer VLAN address. IPsec then encrypts and encapsulates the packet. An outer IP header is added and the packet is sent out over an IP network.

At MLX2, IPsec encapsulation is removed and the packet is decrypted at the termination point of the IPsec tunnel. MPLS processing then translates the VC label to the customer VLAN address and the packet is sent out onto a Layer 2 network.

To route VLL traffic over an IPsec tunnel, the following tasks must be completed.

1. Configure an IPsec tunnel interface.
2. Configure the IPsec interface as an MPLS interface and enable LDP on the interface.
3. Configure VLL traffic to route over the IPsec tunnel.

Limitations for L2 VLL over IPsec

Limitations apply to the configuration of Layer 2 VLL over IPsec:

- VLL over IPsec is only supported for IPv4 IPsec tunnels.
- IP over MPLS does not route over LDP when the outgoing interface is an IPsec tunnel.
- sFlow is not supported for an L2 VLL packet going onto an IPsec tunnel; there is no change to sFlow functionality at the VLL endpoint.
- A maximum of 1500 IPsec tunnels are supported across the system.
- VLL can only be configured over an IPsec tunnel by using LDP; VLL cannot be configured over IPsec by using RSVP.
- The routing protocol that runs on a logical IPsec tunnel interface may not be the same as the routing protocol that runs on the underlying physical destination port for the IPsec tunnel.
- VLL multi-chassis trunking (MCT) over IPsec is not supported.

Enabling LDP on an IPsec tunnel interface

LDP must be enabled on an MPLS interface that is used to route Layer 2 traffic over IPsec.

Before enabling LDP, an IPsec tunnel interface must be set up and configured as an MPLS-capable interface. There is a configuration example at the end of the following task that shows all the configuration steps in order.

Perform the following task to enable LDP on an IPsec tunnel interface.

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

   ```
   device# configure terminal
   ```

2. Enter MPLS configuration mode.

   ```
   device(config)# router mpls
   ```

3. Define an IPsec tunnel interface as an MPLS-capable interface and enter MPLS tunnel configuration mode.

   ```
   device(config-mpls)# mpls-interface tunnel 1
   ```

   In this example, an IPsec tunnel interface with the tunnel ID 1 is defined as an MPLS-capable interface.

4. Enable LDP on the interface.

   ```
   device(config-mpls-if-tnif-1)# ldp-enable
   ```

The following example shows how to configure an IPsec tunnel interface and then enable LDP on the interface.

```
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode ipsec ipv4
device(config-tnif-1)# tunnel protection ipsec profile ipsec-vrf
device(config-tnif-1)# tunnel source 10.22.22.1
device(config-tnif-1)# tunnel destination 10.23.23.2
device(config-tnif-1)# ip address 10.1.1.1/24
device(config-tnif-1)# exit

device(config)# router mpls
device(config-mpls)# mpls-interface tunnel 1
device(config-mpls-if-tnif-1)# ldp-enable
device(config-mpls-if-tnif-1)# end
```

Configuring VLL to route over an IPsec tunnel

VLL traffic can be routed over an IPsec tunnel.

Before configuring VLL for a route over an IPsec tunnel, the IPsec tunnel interface is configured as an MPLS interface and LDP is enabled on the interface. There is a configuration example at the end of the following task that shows all the configuration steps in order.

Perform the following task to configure VLL to route over an IPsec tunnel.

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

   ```
   device# configure terminal
   ```

2. Enter MPLS configuration mode.

   ```
   device(config)# router mpls
   ```

3. Define a VLL service and enter VLL service configuration mode.

```
device(config-mpls)# vll l2oipsec 1
```

4. Configure the IP address of the far-end router (VLL peer).

```
device(config-mpls-l2oipsec)# vll-peer 10.10.1.1
```

5. Define the endpoint of the VLL.

```
device(config-mpls-l2oipsec)# vlan 500
```

In this example, VLAN 500 is defined as the VLL endpoint and you enter the configuration mode for VLAN 500.

6. Define a tagged port for the VLAN.

```
device(config-mpls-l2oipsec-vlan-500)# tagged ethernet 4/1
```

The following example shows how to configure an IPsec tunnel interface, enable LDP on the interface, and configure VLL to route over the IPsec tunnel.

```
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode ipsec ipv4
device(config-tnif-1)# tunnel protection ipsec profile ipsec-vrf
device(config-tnif-1)# tunnel source 10.22.22.1
device(config-tnif-1)# tunnel destination 10.23.23.2
device(config-tnif-1)# ip address 10.1.1.1/24
device(config-tnif-1)# exit

device(config)# router mpls
device(config-mpls)# mpls-interface tunnel 1
device(config-mpls-if-tnif-1)# ldp-enable
device(config-mpls-if-tnif-1)# exit

device(config-mpls)# vll l2oipsec 1
device(config-mpls-vll-l2oipsec)# vll-peer 10.10.1.1
device(config-mpls-vll-l2oipsec)# vlan 500
device(config-mpls-vll-l2oipsec-vlan-500)# tagged ethernet 4/1
device(config-mpls-vll-l2oipsec-vlan-500)# end
```

Configuring VLL to route over a specific IPsec tunnel

VLL traffic can be routed over a specific IPsec tunnel by configuring a static route for the destination IP address.

Before configuring VLL to route over a specific IPsec tunnel, you must bring up multiple LDP tunnels between provider edge (PE) devices. You can bring up multiple tunnels by configuring multiple loopback interfaces.

> **NOTE**
> You should not configure dynamic routing protocols on the loopback interfaces because VLL traffic is routed over a specific IPsec tunnel by using a static route.

When multiple LDP tunnels are configured, perform the following task to configure VLL to route over a specific IPsec tunnel.

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

```
device# configure terminal
```

2. Enter MPLS configuration mode.

```
device(config)# router mpls
```

3. Define a VLL service and enter VLL service configuration mode.

```
device(config-mpls)# vll l2oipsec 1
```

4. Configure the IP address of the far-end router (VLL peer) and specify an LDP tunnel for the peer.

```
device(config-mpls-l2oipsec)# vll-peer 10.10.1.1 ldp 10.3.3.3
```

> **NOTE**
> Only one LDP tunnel can be assigned to a VLL peer.

5. Define the endpoint of the VLL.

```
device(config-mpls-l2oipsec)# vlan 500
```

In this example, VLAN 500 is defined as the VLL endpoint and you enter the configuration mode for VLAN 500.

6. Define a tagged port for the VLAN and return to privileged EXEC mode.

```
device(config-mpls-l2oipsec-vlan-500)# tagged ethernet 4/1
device(config-mpls-l2oipsec-vlan-500)# end
```

7. Enter global configuration mode.

```
device# configure terminal
```

8. Configure VLL to route over the specific IPsec tunnel by configuring a static route.

```
device(config)# ip route 10.3.3.3./32 10.1.1.2
```

The following example shows how to configure VLL to route over a specific IPsec tunnel.

```
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode ipsec ipv4
device(config-tnif-1)# tunnel protection ipsec profile ipsec-vrf
device(config-tnif-1)# tunnel source 10.22.22.1
device(config-tnif-1)# tunnel destination 10.23.23.2
device(config-tnif-1)# ip address 10.1.1.1/24
device(config-tnif-1)# exit

device(config)# router mpls
device(config-mpls)# mpls-interface tunnel 1
device(config-mpls-if-tnif-1)# ldp-enable
device(config-mpls-if-tnif-1)# exit

device(config-mpls)# vll l2oipsec 1
device(config-mpls-vll-l2oipsec)# vll-peer 10.10.1.1 ldp 10.3.3.3
device(config-mpls-vll-l2oipsec)# vlan 500
device(config-mpls-vll-l2oipsec-vlan-500)# tagged ethernet 4/1
device(config-mpls-vll-l2oipsec-vlan-500)# end

device# configure terminal
device(config)# ip route 10.3.3.3/32 10.1.1.2
```

Displaying information about VLL over IPsec

Various show commands can be used to display information about VLL over IPsec configurations.

VLL over IPsec must be configured before displaying this information.

Use any of the following commands to display information about VLL over IPsec configurations:

- Enter the **show mpls vll brief** command to display summary information about VLL configurations.

```
device# show mpls vll brief

* - Active VLL Peer; U - UP; D - DOWN
                                                      Vll-Peer        MCT
Name       VC-ID  End-Point                           (State)         State
----       -----  ---------                           --------        -----
l2oipsec2 2      tag vlan 500  e 4/1(U)                10.10.1.1(U)*   None
l2oipsec4 4      tag vlan 200 inner-vlan 500 e 4/1 (U) 10.10.1.1 (U)*  None
```

- Enter the **show mpls vll** command with the VLL ID to display information about a specific VLL configuration.

```
device# show mpls vll 2

VLL l2oipsec2, VC-ID 2, VLL-INDEX 1

  End-point        : tagged  vlan 500   e 4/1
  End-Point state  : Up
  MCT state        : None
  IFL-ID           : --
  Local VC type    : tag
  Local VC MTU     : 9190
  COS              : --
  Extended Counters: Enabled

  Vll-Peer          : 10.10.1.1
    State           : UP
    Remote VC type : tag                 Remote VC MTU  : 9190
    Local label    : 851968              Remote label   : 851968
    Local group-id : 0                   Remote group-id: 0
    Tunnel LSP     : LDP (tnl2)
    MCT Status TLV : --
    LSPs assigned  : No LSPs assigned
```

- Enter the **show mpls ldp tunnel** command to display LDP tunnel information.

```
device# show mpls ldp tunnel

Total number of LDP tunnels : 8
              Oper      Tunnel    Outbound
To            State     Intf      Intf
10.10.1.1     UP        tnl1      ipsec-tnl1
10.3.3.3      UP        tnl2      ipsec-tnl3
10.5.5.5      UP        tnl3      ipsec-tnl2
```

- Enter the **show nht-table** command to display next-hop table information.

```
device# show nht-table

NHT IP    Index MAC Address     VLAN Out I/F  Out Port  LABEL  EXP/SPI-ID  EXP/ENCAP/SPI-ID
10.0.5.1  1    0024.387c.d100  6    v6      4/2       6      1
10.0.6.1  2    0024.387c.d100  7    v7      4/2       7      3
10.1.5.1  3    0000.0000.0000  1    tn3     tn3       0      1
10.1.6.1  4    0000.0000.0000  1    tn2     tn2       0      3
```

Configuration example for mapping VLL to an IPsec tunnel

When VLL is configured over an IPsec tunnel, the tunnel is only used by VLL traffic.

To configure VLL over an IPsec tunnel, the following tasks must be completed on the devices at each end on the tunnel.

1. Configure an IPsec tunnel interface.

2. Enable LDP on the IPsec tunnel interface.

3. Configure VLL to route over the IPsec tunnel.

The following example shows how to configure VLL over IPsec on the devices Router1 and Router2. In this example, tunnel 1 is configured as an IPsec tunnel by setting the mode of the tunnel to **ipsec**. When the configuration steps are completed on both devices, tunnel 1 is only used by VLL traffic.

Router1

```
Router1(config)# interface tunnel 1
Router1(config-tnif-1)# tunnel mode ipsec ipv4
Router1(config-tnif-1)# tunnel protection ipsec profile ipsec-vrf
Router1(config-tnif-1)# tunnel source 10.22.22.1
Router1(config-tnif-1)# tunnel destination 10.23.23.2
Router1(config-tnif-1)# ip address 10.1.1.1/24
Router1(config-tnif-1)# exit

Router1(config)# router mpls
Router1(config-mpls)# mpls-interface tunnel 1
Router1(config-mpls-if-tnif-1)# ldp-enable
Router1(config-mpls-if-tnif-1)# exit

Router1(config-mpls)# vll l2oipsec 1
Router1(config-mpls-vll-l2oipsec)# vll-peer 10.10.1.1
Router1(config-mpls-vll-l2oipsec)# vlan 500
Router1(config-mpls-vll-l2oipsec-vlan-500)# tagged ethernet 4/1
Router1(config-mpls-vll-l2oipsec-vlan-500)# end
```

Router2

```
Router2(config)# interface tunnel 1
Router2(config-tnif-1)# tunnel mode ipsec ipv4
Router2(config-tnif-1)# tunnel protection ipsec profile ipsec-vrf
Router2(config-tnif-1)# tunnel source 10.23.23.2
Router2(config-tnif-1)# tunnel destination 10.22.22.1
Router2(config-tnif-1)# ip address 10.1.1.2/24
Router2(config-tnif-1)# exit

Router2(config)# router mpls
Router2(config-mpls)# mpls-interface tunnel 1
Router2(config-mpls-if-tnif-1)# ldp-enable
Router2(config-mpls-if-tnif-1)# exit

Router2(config-mpls)# vll l2oipsec 1
Router2(config-mpls-vll-l2oipsec)# vll-peer 10.2.2.2
Router2(config-mpls-vll-l2oipsec)# vlan 500
Router2(config-mpls-vll-l2oipsec-vlan-500)# tagged ethernet 3/1
Router2(config-mpls-vll-l2oipsec-vlan-500)# end
```

Configuration example for mapping VLL to a specific IPsec tunnel

When multiple LDP tunnels are configured, VLL can be mapped to a specific IPsec tunnel.

> **NOTE**
> VLL can only be mapped to a specific IPsec tunnel when multiple LDP tunnels exist. Multiple LDP tunnels can be brought up by configuring multiple loopback interfaces between provider edge (PE) devices.

To map VLL to a specific IPsec tunnel, the following tasks must be completed on the devices at each end on the tunnel:

1. Configure an IPsec tunnel interface.

2. Enable LDP on the IPsec tunnel interface.

3. Configure a static route to map VLL traffic to the specific IPsec tunnel.

The following example describes how to map VLL to a specific IPsec tunnel on the devices Router1 and Router2.

Router1

```
Router1(config)# interface tunnel 1
Router1(config-tnif-1)# tunnel mode ipsec ipv4
Router1(config-tnif-1)# tunnel protection ipsec profile ipsec-vrf
Router1(config-tnif-1)# tunnel source 10.22.22.1
Router1(config-tnif-1)# tunnel destination 10.23.23.2
Router1(config-tnif-1)# ip address 10.1.1.1/24
Router1(config-tnif-1)# exit

Router1(config)# router mpls
Router1(config-mpls)# mpls-interface tunnel 1
Router1(config-mpls-if-tnif-1)# ldp-enable
Router1(config-mpls-if-tnif-1)# exit

Router1(config-mpls)# vll l2oipsec 1
Router1(config-mpls-vll-l2oipsec)# vll-peer 10.10.1.1 ldp 10.3.3.3
Router1(config-mpls-vll-l2oipsec)# vlan 500
Router1(config-mpls-vll-l2oipsec-vlan-500)# tagged ethernet 4/1
Router1(config-mpls-vll-l2oipsec-vlan-500)# exit
Router1(config-mpls-vll-l2oipsec)# exit
Router1(config-mpls)# end

Router1# configure terminal
Router1(config)# ip route 10.3.3.3/32 10.1.1.2
```

Router2

```
Router2(config)# interface tunnel 1
Router2(config-tnif-1)# tunnel mode ipsec ipv4
Router2(config-tnif-1)# tunnel protection ipsec profile ipsec-vrf
Router2(config-tnif-1)# tunnel source 10.23.23.2
Router2(config-tnif-1)# tunnel destination 10.22.22.1
Router2(config-tnif-1)# ip address 10.1.1.2/24
Router2(config-tnif-1)# exit

Router2(config)# router mpls
Router2(config-mpls)# mpls-interface tunnel 1
Router2(config-mpls-if-tnif-1)# ldp-enable
Router2(config-mpls-if-tnif-1)# exit

Router2(config-mpls)# vll l2oipsec 1
Router2(config-mpls-vll-l2oipsec)# vll-peer 10.2.2.2 ldp 10.4.4.4
```

```
Router2(config-mpls-vll-l2oipsec)# vlan 500
Router2(config-mpls-vll-l2oipsec-vlan-500)# tagged ethernet 3/1
Router2(config-mpls-vll-l2oipsec-vlan-500)# exit
Router2(config-mpls-vll-l2oipsec)# exit
Router2(config-mpls)# end

Router2# configure terminal
Router2(config)# ip route 10.4.4.4/32 10.1.1.1
```

VLL load Balancing

VLL load balancing

Virtual Leased Line (VLL) load balancing provides VLL LSP load balancing with a maximum of eight LSPs or LDPs. It also provides VLL LSP load balancing with the assigned LSP.

When more than one tunnel exists from the device to a VLL peer, the device can select multiple tunnels to forward VLL traffic to the peer.

When configuring an LSP assignment, the device ignores the configured CoS of the LSP and ignores the VLL to select an LSP for the VLL peer. However, traffic sent on the LSP uses the CoS of the LSP. When enabling LSP load balancing for a VLL peer, traffic is load balanced.

For VLL load balancing, select an LSP based on a hash-index. The LSP in turn, calculates the following:

- Layer 2, non-IPv4, and IPv6 packets: Source MAC address and destination MAC address.

- IPv4, non-TCP/UDP packets: Source MAC address and destination MAC address, source IP address and destination IP address.

- IPv4 TCP packets: Source MAC address and destination MAC address, source IP address and destination IP address, and TCP source port and TCP destination port.

- IPv4 UDP packets: Source MAC address and destination MAC address, source IP address and destination IP address, and UDP source port and UDP destination port.

- IPv6 non-TCP/UDP packets: Source MAC address and destination MAC address, source IP address and destination IP address.

- IPv6 TCP packets: Source MAC address and destination MAC address, source IP address and destination IP address, and TCP source port and TCP destination port.

- IPv6 UDP packets: Source MAC address and destination MAC address, source IP address and destination IP address, and UDP source port and UDP destination port.

  NOTE
  For VLL traffic, source and destination MAC addresses come from the inner Ethernet
  header.

The eligibility of a tunnel LSP for load balancing depends on whether CoS values are defined for the VLL instance and the tunnel LSP:

- When the VLL instance does not have a CoS value defined, then all tunnel LSPs to the peer are eligible for load balancing.

- When a VLL instance has a CoS value defined, and at least, one tunnel LSP to the peer has a CoS value less than or equal to the VLL instance CoS value, then all tunnel LSPs with the highest CoS value less than or equal to the VLL instance CoS value are eligible for load balancing.

- When a VLL instance has a CoS value defined, and none of the tunnel LSPs to the peer has a CoS value less than or equal to the VLL instance CoS value, then all tunnel LSPs to the peer that do not have a CoS value are eligible for load balancing.

VLL load balancing assumptions and dependencies

This section describes assumptions, dependencies, and limitations for VLL load balancing.

- The hashing decision is based on the fields in each packet received.

- VLL can use both LDP and RSVP tunnels for load balancing as long as they all match the CoS criteria. A tunnel reachable by a peer with a valid CoS value is all that is required to be used as a candidate for VLL tunnel load balancing. There are no preferences for a particular type of tunnel dependencies.

- The LSPs chosen for load balancing must have the same CoS values. For example, when CoS of LSP1 = 4, LSP2 = 4, LSP3 = 2, LSP4 =2, LSP5 = 1, and VLL instance CoS = 3, then traffic is load balanced with LSP3 and LSP4, which have the same CoS values.

- The hashing technique to load balance is not consistent.

- A maximum of eight LSPs can be used for load balancing.

- The maximum VLL instance is 16000.

IP over MPLS

One of the benefits that MPLS offers service providers is the ability to take advantage of MPLS traffic engineering capabilities to efficiently utilize the service provider network bandwidth, to control traffic placement, and to achieve fast network resilience. This is accomplished through IP-over-MPLS features.

The following sections describe some of the procedures and considerations required when configuring a device to carry IP traffic over an MPLS network:

* BGP shortcuts on page 383 - This feature directs BGP to resolve a route nexthop to a MPLS LSP when one is available.

* LDP route injection on page 388 – This feature allows the user to make selected customer routes available though LDP created LSP tunnels.

* Using traffic-engineered LSPs within an AS on page 393 - This section describes how CoS values are determined for packets through an LSP.

  NOTE
  Do not forward packets from one type of tunnel to another type of tunnel in XPP. Packets may not be routed properly.

BGP shortcuts

In a typical configuration, BGP considers only IP routes when building a routing table. When an MPLS network uses BGP to propagate routes, BGP must consider whether the MPLS tunnels are viable routes. The BGP shortcut feature forces BGP to use an MPLS tunnel as the preferred route to a destination network when one is available. The user can also force BGP to include LSP metrics for best-route computations.

When configured on an MPLS edge router, BGP computes routes to destinations available through other edge routers. When BGP determines that a route is available through an edge router that is reachable through an MPLS tunnel, a BGP shortcut directs BGP to place the MPLS tunnel in the routing table as the preferred BGP route.

The user can globally enable the BGP shortcut feature and optional inclusion of LSP metrics on an Extreme device. With the BGP shortcut feature enabled, the Extreme device first attempts to resolve BGP routes with an MPLS tunnel, and can optionally consider LSP metrics. When the BGP attempt at route resolution is unsuccessful, the Extreme device defaults to the IPv4 routing table.

Key algorithms

This section describes the behavior of the system in three contexts.

* **Next-hop MPLS disabled:** Only IP routing tables are used to resolve routes for the routing table.

* **Next-hop MPLS enabled:** LSP with a fixed metric of one is used to resolve the routes. For routes that cannot be resolved, the system uses the routing table.

- **Next-hop MPLS with LSP metric consideration:** When BGP resolves the next hop with LSP, it uses the LSP metric as the IGP cost for that next hop. When any of the possible paths are through an LSP, then only LSPs are chosen. The IGP cost of each next hop is then compared, and only IGP cost paths with the lowest values are considered for ECMP.

Native IP forwarding

When next-hop MPLS is disabled, BGP uses the default BGP decision process and native IP forwarding to build BGP EMCP routes.

Next-hop MPLS

For each unique BGP next hop, when next-hop MPLS is enabled, BGP first determines when an LSP can be used to resolve the route. When BGP can resolve the route, it does not check the native IP routing table.

For each BGP next hop, when the route is resolved by LSP, then all possible LSPs with the same lowest-metric value are selected. After this selection, BGP internally sets this next hop IGP cost to one (rather than the true LSP metric) to force it to be the preferred hop over a hop resolved by native IP.

For each BGP next hop, the IGP cost is compared, and the least-value IGP cost for the next hop or hops are used to install them in the routing table.

When the Extreme device installs a BGP route in the RTM, it uses a BGP metric, not the IGP metric (IGP cost.)

Next-hop MPLS comparing LSP metrics

With the option enabled to compare LSP metrics, after BGP resolves a next hop with LSP, it uses the LSP metric as the IGP cost for that next hop. Thereafter, all of the next hops IGP costs are compared, and only the IGP cost paths with the lowest values are considered for ECMP. When any of these paths is an LSP, then only LSP paths are taken.

The user have the flexibility to choose a native IP path over an LSP path when they have different BGP next-hops, and the native IP path has a lower IGP cost.

> NOTE
> Enabling or disabling the LSP metric option takes effect immediately: BGP automatically recalculates the existing BGP routes.

To configure BGP shortcuts and optionally compare LSP metrics, use the **next-hop-mpls** command in BGP configuration mode, as in the following example.

```
device(config)# router bgp
device(config-bgp)# next-hop-mpls compare-lsp-metric
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

For the **next-hop-mpls** command, when the user employs the [**no**] form with the optional **compare-lsp-metric** parameter, only this optional parameter is deleted, so the global next-hop MPLS enable remains the same. To disable both the optional LSP-metric compare and the global next-hop MPLS, use the [**no**] form of the command but without the optional **compare-lsp-metric** parameter.

Examples of next-hop MPLS

This section illustrates how to configure a BGP shortcut by enabling next-hop MPLS. It also illustrates the optional parameter -- the consideration of LSP metrics:

- Enabling next-hop MPLS (LSP metric becomes fixed at 1).
- Enabling compare-LSP-metric (so IGP metric is compared with user-configurable LSP metric).

- Disabling next-hop MPLS.

Enable next-hop MPLS using the **next-hop-mpls** command, as the following example illustrates. The follow-up **show** command of the running configuration indicates the global enabling of this feature.

```
device(config-bgp)# next-hop-mpls
device(config-bgp)# show ip bgp config
Current BGP configuration:
router bgp
 local-as 10
 neighbor 10.1.1.2 remote-as 20
 neighbor 10.10.1.2 remote-as 20
 address-family ipv4 unicast
 next-hop-mpls
 exit-address-family
 address-family ipv4 multicast
 exit-address-family
 address-family ipv6 unicast
 exit-address-family
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

Enable the Extreme device to use the compare LSP metric. The running configuration reflects the global configuration on one line.

```
device(config-bgp)# next-hop-mpls compare-lsp-metric
device(config-bgp)# show ip bgp config
Current BGP configuration:
router bgp
 local-as 10
 neighbor 10.1.1.2 remote-as 20
 neighbor 10.10.1.2 remote-as 20
 address-family ipv4 unicast
 next-hop-mpls compare-lsp-metric
 exit-address-family
 address-family ipv4 multicast
 exit-address-family
 address-family ipv6 unicast
 exit-address-family
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

This series of examples shows how an IP-only routing table resolution for BGP is affected first by the enabling of next-hop MPLS and then by the enabling of LSP-metric comparison. The tasks for these examples are:

- Specify metrics for three LSPs. The existing LSPs in this example are to2, to22, and to2_sec. As a precondition for this example, their metrics are changed to 10, 20, and 10.

- Enable BGP ECMP, then check the routing table. The destination IP address for this example is 10.8.8.1/32. The routing table shows that native IP-forwarding is used.

- Enable next-hop MPLS and observe the effect on the route to 10.8.8.1/32.

- Enable LSP-metric comparison and note that, because of the metric for LSP to22, it has no effect on the routing table.

- Change the metric for an LSP (to2 in this example).

- Disable LSP-metric compare and check the consequences.

- Disable global next-hop MPLS.

Specifying metrics

This step specifies metrics for three LSPs.

```
device(config-bgp)# router mpls
device(config-mpls)# lsp to2
device(config-mpls-lsp-to2)# disable
device(config-mpls-lsp-to2)# to 10.1.1.2
```

```
device(config-mpls-lsp-to2)# from 10.1.1.1
device(config-mpls-lsp-to2)# metric 10
device(config-mpls-lsp-to2)# enable
Connecting signaled LSP to2
exit
....
device(config-mpls)# lsp to22
device(config-mpls-lsp-to22)# disable
device(config-mpls-lsp-to22)# to 10.1.1.2
device(config-mpls-lsp-to22)# from 10.1.1.1
device(config-mpls-lsp-to22)# metric 20
device(config-mpls-lsp-to22)# enable
Connecting signaled LSP to22
exit
....
device(config-mpls)# lsp to2_sec
device(config-mpls-lsp-to2_sec)# diable
device(config-mpls-lsp-to2_sec)# to 10.10.1.2
device(config-mpls-lsp-to2_sec)# from 10.10.1.1
device(config-mpls-lsp-to2_sec)# metric 10
device(config-mpls-lsp-to2_sec)# enable
Connecting signaled LSP to2_sec
exit
device(config-mpls)# show mpls lsp
                         Admin Oper  Tunnel  Up/Dn Retry Active
Name            To       State State Intf    Times No.   Path
to2             10.1.1.2    UP    UP  tnl0    1     0     --
to2_sec         10.10.1.2   UP    UP  tnl2    1     0     --
to22            10.1.1.2    UP    UP  tnl1    1     0     --
```

Syntax: **[no] metric** *num*

Enable BGP ECMP

This example shows BGP ECMP being enabled and the check of the *Routing Table Manager (RTM)* by the **show ip route** command. The destination for this example is 10.8.8.8/32, and native IP forwarding is in effect.

```
device(config-mpls)# router bgp
device(config-bgp)# maximum 5
device(config-bgp)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
    Destination    Gateway      Port        Cost  Type Uptime
1   10.1.1.1/32    DIRECT       loopback 1  0/0   D    9m46s
2   10.2.3.3/32    DIRECT       loopback 2  0/0   D    9m46s
3   10.5.5.5/32    10.1.1.10    eth 1/1     1/1   S    9m35s
4   10.8.8.1/32    10.1.1.2     eth 1/1     20/0  B    0m1s
    10.8.8.1/32    10.10.1.2    eth 1/2     20/0  B    0m1s
5   10.8.8.2/32    10.1.1.2     eth 1/1     20/0  B    0m1s
    10.8.8.2/32    10.10.1.2    eth 1/2     20/0  B    0m1s
```

Enable next-hop MPLS

In this example, the next-hop MPLS is enabled, and the **show ip route** command is used to check the RTM.

```
device(config-bgp)# next-hop-mpls
device(config-bgp)# show ip route
Total number of IP routes: 4
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
    Destination    Gateway      Port        Cost  Type Uptime
1   10.1.1.1/32    DIRECT       loopback 1  0/0   D    10m4s
2   10.2.3.3/32    DIRECT       loopback 2  0/0   D    10m4s
3   10.5.5.5/32    10.1.1.10    eth 1/1     1/1   S    9m53s
```

```
4   10.8.8.1/32     10.1.1.2    lsp to2     20/0  B    0m1s
    10.8.8.1/32     10.10.1.2   lsp to2_sec 20/0  B    0m1s
5   10.8.8.2/32     10.1.1.2    lsp to2     20/0  B    0m1s
    10.8.8.2/32     10.10.1.2   lsp to2_sec 20/0  B    0m1s
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

Enable LSP-metric comparison

For this example, LSP-metric comparison is enabled and the consequences are checked in the RTM. In this case, LSPs to2 and to2_sec already provide the best route, so this display does not differ from the example in which next-hop MPLS is enabled. Note that to22 is not displayed because its metric is 20, but the metric of to2 (to the same destination) is only 10 and so represents the chosen LSP.

```
device(config-bgp)# next-hop-mpls compare-lsp-metric
device(config-bgp)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
    Destination   Gateway     Port        Cost    Type Uptime
1   10.1.1.1/32   DIRECT      loopback 1  0/0     D    11m30s
2   10.2.3.3/32   DIRECT      loopback 2  0/0     D    11m30s
3   10.5.5.5/32   10.1.1.10   eth 1/1     1/1     S    11m19s
4   10.8.8.1/32   10.1.1.2    lsp to2     20/0    B    0m1s
    10.8.8.1/32   10.10.1.2   lsp to2_sec 20/0    B    0m1s
5   10.8.8.2/32   10.1.1.2    lsp to2     20/0    B    0m1s
    10.8.8.2/32   10.10.1.2   lsp to2_sec 20/0    B    0m1s
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

Changing the metric for an LSP

In the next example, the metric for LSP to2 is changed to a value (20) that causes the system to remove it from the routing table, so only LSP to2_sec to 10.8.8.1/32 remains. This output illustrates this result.

```
device(config-mpls)# lsp to2
device(config-mpls-lsp-to2)# disconnect
Disconnecting signaled LSP
device(config-mpls-lsp-to2)# metric 20
device(config-mpls-lsp-to2)# enable
Connecting signaled LSP to2
device(config-mpls)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
    Destination   Gateway     Port        Cost    Type Uptime
1   10.1.1.1/32   DIRECT      loopback 1  0/0     D    12m23s
2   10.2.3.3/32   DIRECT      loopback 2  0/0     D    12m23s
3   10.5.5.5/32   10.1.1.10   eth 1/1     1/1     S    12m12s
4   10.8.8.1/32   10.10.1.2   lsp to2_sec 20/0    B    0m6s
5   10.8.8.2/32   10.10.1.2   lsp to2_sec 20/0    B    0m6s
```

Disabling LSP-metric compare and checking the consequences

For the last example related to next-hop MPLS, disable LSP-metric compare using the [no] form of the **next-hop-mpls** command and include the **compare-lsp-metric** option.

> NOTE
> When the user employs the [no] form with the optional **compare-lsp-metric** parameter for the **next-hop-mpls** command, only this optional parameter is deleted, so global next-hop-mpls enable remains the same. To disable both the optional LSP-metric compare and the global next-hop-mpls, use the [no] form of the **next-hop-mpls** command without the optional parameter.

Because global next-hop MPLS remains enabled and the LSP metrics are no longer a factor, all the LSPs are displayed in the routing table because BGP considers them to have equal cost.

```
device(config-bgp)# no next-hop-mpls compare-lsp-metric
device(config-bgp)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Typ 1 2:External Type 2 s:Sham Link
    Destination    Gateway     Port        Cost   Type Uptime
1   10.1.1.1/32    DIRECT      loopback 1  0/0    D    12m58s
2   10.2.3.3/32    DIRECT      loopback 2  0/0    D    12m58s
3   10.5.5.5/32    10.1.1.10   eth 1/1     1/1    S    12m47s
4   10.8.8.1/32    10.1.1.2    lsp to2     20/0   B    0m1s
    10.8.8.1/32    10.1.1.2    lsp to22    20/0   B    0m1s
    10.8.8.1/32    10.10.1.2   lsp to2_sec 20/0   B    0m1s
5   10.8.8.2/32    10.1.1.2    lsp to2     20/0   B    0m1s
    10.8.8.2/32    10.1.1.2    lsp to22    20/0   B    0m1s
    10.8.8.2/32    10.10.1.2   lsp to2_sec 20/0   B    0m1s
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

Disabling global next-hop MPLS

Disable global next-hop MPLS and check the RTM to see that native IP-forwarding is restored.

```
device(config-bgp)# no next-hop-mpls
device(config-bgp)# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
    Destination    Gateway     Port        Cost   Type Uptime
1   10.1.1.1/32    DIRECT      loopback 1  0/0    D    9m46s
2   10.2.3.3/32    DIRECT      loopback 2  0/0    D    9m46s
3   10.5.5.5/32    10.1.1.10   eth 1/1     1/1    S    9m35s
4   10.8.8.1/32    10.1.1.2    eth 1/1     20/0   B    0m1s
    10.8.8.1/32    10.10.1.2   eth 1/2     20/0   B    0m1s
5   10.8.8.2/32    10.1.1.2    eth 1/1     20/0   B    0m1s
    10.8.8.2/32    10.10.1.2   eth 1/2     20/0   B    0m1s
```

Syntax: [no] next-hop-mpls [compare-lsp-metric]

LDP route injection

An MPLS edge router is typically connected to a customer network that is not configured for MPLS. When the edge router is then connected to the MPLS core through LSP tunnels that have been created by LDP, only routes to the loopback address of the edge router are available for routing through the LSP tunnels. In practice, this means that routes to and from the customer network are unavailable to the MPLS network.

The LDP route injection feature allows the user to make routes available from the customer network through LSPs that have been created by LDP. The user can filter routes that the user wants to allow through the MPLS network using an ACL, and then apply that ACL to the **advertise-labels for** command. The routes injected are then accessible over the MPLS network.

To direct the device to inject non-loopback routes into LDP while restricting the routes injected through reference to an ACL, enter the following command.

```
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# advertise-labels for 30
```

Syntax: advertise-labels for *access-list*

The *access-list* variable refers to the number of the access list that filters for the routes the user wants to use for label binding.

LDP route injection improvements

This document describes the design details for the LDP route injection feature.

The route injection feature contains the following improvements and enhancements:

- Improvements in LDP CLI commands.
- Syslog enhancement for LDP system DOWN.
- Ability to Clear or Teardown an RSVP session.
- Improvements in LDP display outputs.

LDP route injection specifications

Describes the specifications for the LDP route injection feature.

Improvements in LDP CLI commands

A change is made to the area in LDP:

- Use of PREFIX-LIST instead of ACL in LDP.

LDP, by default, advertises all /32 prefixes that are learned from all the loopback interfaces to all other LDP peers. To enable LDP to advertise other prefixes that are learned by IGP, the "LDP route injection" feature is used. In this feature, an ACL is created to specify the prefixes to be permitted or denied and it is applied in the **advertise-labels for** command available in LDP.

There are two other FEC filtering mechanisms in LDP for inbound and outbound FEC filtering. These use prefix-lists instead of ACL. Prefix-lists are more compact, flexible and perform better than ACLs. The configuration command for LDP route injection is changed to use a prefix-list instead of ACL for above mentioned advantages. This makes the configuration more homogeneous in FEC learning and distribution control mechanisms available in LDP.

To achieve this objective, the existing command using ACLs is deprecated and the updated command is implemented using prefix-list. All other customer use-cases for the MPLS RAS feature are unchanged.

Syslog enhancement for LDP session DOWN

This RAS enhancement addresses this shortcoming by adding the session DOWN reason to the syslog output.

Clear or teardown of an RSVP session

A command exists for clearing an LSP from the ingress router (**clear mpls** [**lsp** | **bypass-lsp**]). This feature introduces a command to use to tear down an RSVP session from any of the ingress, transit, or egress routers.

On executing the command provided, a PATHERR and a RESVTEAR is sent upstream to the ingress of the session. In response, the ingress router initiates PATHTEAR and the LSP is torn down. This process tears down all the sessions for the LSP including the backup sessions for that LSP.

The secondary LSP is assigned a different tunnel ID than the primary LSP. So, clearing the primary LSP does not teardown the secondary LSP, and vice versa.

Considerations when using LDP route injection

1. The user can directly change the LDP route injection filter without deleting a previously configured one. The change automatically applies and triggers LDP route re-injections.
2. Any change to a referenced ACL automatically applies to LDP route injection filtering and triggers LDP route re-injection.
3. When no LDP route injection filter is configured, by default LDP acquires all local loopback addresses.
4. When the ACL referenced by the LDP route injection filter is not configured, it is an implicit deny. All local routes are denied.
5. Both number-based and name-based ACLs can be used. Because only prefix-based filtering is applied, use of a standard ACL is preferred.
6. The LDP route injection filter is only applied on local route injection. Learned remote binding is not filtered.

Feature requirements

Requirements for the route injection feature.

Requirement for Improvements in LDP CLI commands

Usage of prefix-list instead of ACL in LDP.

Requirement for Syslog enhancement for LDP session DOWN

LDP session DOWN syslog also displays the session DOWN reason.

Requirement for the clear or teardown of an RSVP session

A CLI is provided to clear an RSVP session from the transit or egress router.

Requirement for the show mpls ldp tunnel command output sorting

The existing CLI in the customer use cases of the LDP route injection feature, except that it uses a prefix-list instead of an ACL.

LDP route injection example

This example describes how to use LDP route injection to inject routes 10.2.2.2/32 and 10.5.5.2/32 into the LDP label information database.

1. The **show ip interface** command displays IP addresses of loopback interfaces in Router 1.

```
device# show ip interface
Interface       P-Address   OK?  Method   Status   Protocol   VRF
eth 1/1         10.0.0.1    YES  NVRAM    up       up         default
eth 1/2         10.0.1.1    YES  NVRAM    up       up         default
loopback 3      10.3.3.3    YES  NVRAM    up       up         default
loopback 5      10.5.5.5    YES  manual   up       up         default
```

2. By default, the LDP label information database only contains labels learned for IP addresses of loopback interfaces, as demonstrated in this example, where only prefixes 10.3.3.3/32 and 10.5.5.5/32 are displayed by the **show mpls ldp database** command.

```
device# show mpls ldp database
Session 10.3.3.3:0 - 10.5.5.2:0
 Downstream label database:
   Label      Prefix           State
   1024       10.3.3.3/32      Retained
 Upstream label database:
   Label      Prefix
   3          10.3.3.3/32
   3          10.5.5.5/32
```

3. The **show ip route** command displays routes available to ports on Router 1.

```
device# show ip route
Total number of IP routes: 9
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
     Destination     Gateway      Port        Cost    Type
1    10.2.2.2/32     10.0.0.2     eth 1/1     1/1     S
2    10.3.3.3/32     DIRECT       loopback 3  0/0     D
3    10.5.5.0/24     10.0.0.2     eth 1/1     1/1     S
4    10.5.5.1/32     10.0.0.2     eth 1/1     1/1     S
5    10.5.5.2/32     10.0.0.2     eth 1/1     110/2   O
6    10.5.5.5/32     DIRECT       loopback 5  0/0     D
7    10.5.6.2/32     10.0.0.2     eth 1/1     1/1     S
8    10.0.0.0/24     DIRECT       eth 1/1     0/0     D
9    10.10.0.0/24    DIRECT       eth 1/2     0/0     D
```

4. In this example, a filter is configured to inject route 10.2.2.2/32.

```
device(config)# access-list 30 permit 10.2.2.2/32
device(config)# router mpls
device(config-mpls)# ldp
device(config-mpls-ldp)# advertise for 30
```

5. As shown, the 10.2.2.2/32 has been injected into the LDP Label information database.

```
device# show mpls ldp database
Session 10.3.3.3:0 - 10.5.5.2:0
 Downstream label database:
   Label      Prefix           State
 Upstream label database:
   Label      Prefix
   3          10.2.2.2/32
```

6. In this example a second filter is configured to inject route 10.5.5.2/32.

```
device(config)# access-list 30 permit 10.5.5.2/32
```

7. As shown, route 10.5.5.2/32 has been injected into the LDP label information database.

```
device(config)# show mpls ldp database
Session 10.3.3.3:0 - 10.5.5.2:0
 Downstream label database:
   Label      Prefix           State
 Upstream label database:
   Label      Prefix
   3          10.2.2.2/32
   3          10.5.5.2/32
```

Customer use cases

Customer use case for syslog enhancement

Customer use case for Improvement in CLI command

There is no change in the customer use cases of LDP route injection feature, except that it uses a prefix-list instead of an ACL.

Customer use case for syslog enhancement for LDP session DOWN.

LDP session DOWN	
Pre-condition	LDP session established between two routers R1 and R2.
Trigger	Multiple triggers on Router R1for bringing the LDP session down. 1. Clear LDP neighbor. 2. Disable LDP on the interface. 3. Disable OSPF on the LDP interface. 4. Disable loopback interface. 5. Disable OSPF on the loopback interface. 6. Disable the physical interface. 7. Add ACL to filter LDP session keepalive messages. (With or without GR enabled.) 8. Add ACL to filter LDP hello messages. (With or without GR enabled.) 9. Un-configure the MPLS interface. 10. Un-configure 'router mpls'.
Basic flow	LDP session comes down as indicated by the syslog message which contains the appropriate session down reason code. Verify using syslogs on both R1 and R2.
Alternate flow	--

Upgrade and downgrade compatibility

The existing ACL based configuration in LDP is already supported if previously configured an works seamlessly after upgrade. However, the ACL command in LDP is depreciated by making it hidden and giving a warning prompt telling the operator to use the prefix-list base configuration command.

Syslog enhancement for LDP session DOWN

When using this feature, the syslog for the LDP session DOWN displays the reason for the session DOWN.

The show ldp tunnel output sorting

When using this feature, the output of the **show mpls ldp tunnel** is always sorted.

Backward compatibility

This feature is not backward compatible because of the change in the CLI command.

Displaying routes through LSP tunnels

Once a network has been enabled to allow routes through LSP tunnels, the routes appear in the IP routing table. In the following example, the **show ip route** command displays a table that contains routes through LSP tunnels. In this example, routes 7 - 8 and 10 - 14 are LDP tunnels.

```
device# show ip route
Total number of IP routes: 1027
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost -
Dist/Metric
    Destination     Gateway        Port          Cost       Type
1   10.1.1.1/32     DIRECT         loopback 1    0/0        D
2   10.1.2.1/32     DIRECT         loopback 2    0/0        D
3   10.1.3.1/32     DIRECT         loopback 3    0/0        D
4   10.2.2.2/32     10.0.0.2       eth 1/1       110/10     O
5   10.3.3.3/32     10.0.0.2       eth 1/1       110/12     O
    10.3.3.3/32     10.8.0.2       eth 1/4       110/12     O
6   10.4.4.4/32     10.8.0.2       eth 1/4       110/10     O
7   10.5.1.5/32     10.5.5.5       lsp(LDP)      200/0      B
8   10.5.3.5/32     10.5.5.5        lsp(LDP)      200/0       B
9   10.5.5.5/32     10.0.0.2       eth 1/1       110/13     O
    10.5.5.5/32     10.8.0.2       eth 1/4       110/13     O
10  10.6.1.6/32     10.6.6.6        lsp(LDP)      200/0       B
11  10.6.2.6/32     10.6.6.6        lsp(LDP)      200/0       B
12  10.6.3.6/32     10.6.6.6        lsp(LDP)      200/0       B
13  10.6.4.6/32     10.6.6.6       lsp(LDP)      200/0      B
14  10.6.5.6/32     10.6.6.6       lsp(LDP)      200/0      B
15  10.6.6.6/32     10.0.0.2       eth 1/1       110/14     O
    10.6.6.6/32     10.8.0.2       eth 1/4       110/14     O
```

ACL to prefix-list conversion in LDP

There are two other FEC filtering mechanisms in LDP for inbound and outbound FEC filtering. These use prefix-lists instead of ACL, as prefix-lists are more compact, flexible, and perform better than ACLs.

The configuration command for LSD router injection uses prefix-lists instead of ACLs . This also makes the configuration more homogeneous in FEC learning and distribution control mechanisms available in LDP.

Using traffic-engineered LSPs within an AS

In addition to traffic destined to travel outside an AS, Extreme devices can forward internal AS traffic into LSP tunnels. This feature allows the user to configure a signaled LSP to serve as a shortcut between nodes in an AS. In a shortcut LSP, OSPF includes the LSP in the SPF calculation. When OSPF determines that the LSP shortcut is the best path to a destination, it installs a route into the IP routing table, specifying the LSP tunnel interface as the outbound interface, as well as the cost of the LSP. Only LSPs configured to router IDs can be considered as shortcuts. When the LSP goes down or is administratively disabled, the LSP tunnel route is removed from the main routing table.

The cost of the LSP is the user-configured metric for the LSP. When there is no user-configured metric, the underlying IP cost of the LSP is used. For example, when the IP cost of the best underlying path between two routers is 2, and there is an LSP configured between these two routers, the cost of the LSP is 2. Once an LSP is used as a next hop for a destination, the cost of the LSP can be used to calculate other destinations that can use the LSP egress node as next hop. This allows traffic for addresses downstream from the LSP egress node (including prefixes of the egress node) to use the LSP shortcut.

When OSPF is already using an LSP tunnel route to an *Area Border Router (ABR)*, all inter-area routes through that ABR use the LSP as the next hop, provided there are no other better paths to the destination (paths through other ABRs). An LSP to a destination outside an area is not used by OSPF in the calculation of inter-area routes.

Only signaled LSPs can be used as OSPF shortcuts. RSVP packets, used to establish and maintain signaled LSPs, are never forwarded into LSP tunnels.

Refer to Creating OSPF shortcuts over an LSP tunnel on page 395 for more information.

BGP MPLS metric follow IGP

This feature is to enable BGP to rely purely on IGP metric to the BGP next hop to determine the best path for IP over MPLS and Layer-3 VPN cases.

In Extreme device BGP implementation, MPLS metric value is always used as IGP cost when resolving BGP routes with MPLS tunnel. This feature is to have a way to use the IGP metric to the BGP next hop to be used as IGP cost.

When using this feature, the MPLS metric cost is completely ignored in the BGP decision process. So user must be aware and ensure that their IGP costs are consistent across their network and that they indeed want to rely only on the IGP cost to determine where to send the traffic. This works the best when the MPLS LSP metrics follow the IGP cost, thus they have the full advantage of both routing protocols and MPLS to select the best path.

The advantage of doing this is that the BGP decision process on IGP cost is purely based on the IGP cost to different next hop. MPLS tunnel is treated as a "relay" service. Once BGP determined which IGP is a better way to go, the system checks when there is a MPLS tunnel for that path. This design has the advantage in customer network where the IGP cost is significant throughout their local domain and all routing protocols, and they want to use that as a tie-breaker rather than use MPLS specific metric value.

A few things must be clarified:

1. This feature does NOT override MPLS LSP metric value. MPLS internal metric value is the same.
2. When this feature is turned on, BGP ignores all MPLS LSP metrics, whether it is a default LSP metric value or a configured specific value. This is applicable to IPv4, IPv6 and Layer-3 VPN BGP next hop resolutions.
3. This document details this feature for BGP. Static route miss some feature (**compare-lsp-metric**), so this feature is not added to the static route for now.

    The *RFE 3053* is mainly for BGP to set MED value as IGP cost and advertise out. This feature has the foundation for this for BGP routes resolving next hop to MPLS LSP tunnel. Then, a route-map is required to set BGP MED value to IGP metric by set metric-type internal.

Feature information

*   The two options **compare-lsp-metric** and **follow-igp** are mutually exclusive. Because one option uses MPLS metric value, the other uses IGP metric.
*   This command takes effect immediately and automatically. No clear BGP session or clear routes command is required.
*   The **show ip bgp nexthop** command displays the current BGP next hop IGP cost. When that is using MPLS LSP as outgoing interface, the value reflects MPLS LSP metric when using **compare-lsp-metric** and IGP metric when using **follow-igp.**
*   Combined with another BGP command, **install-igp-cost** under **router bgp**, you can see that IGP cost is installed with BGP routes in RTM.
*   Combined with BGP outbound policy for **set metric-type internal**, you can now set Layer-3 VPN and IP over MPLS routes using IGP metric to send out as MED.

Limitations

- When the user is running IGP throughout the network, and the IGP metric is trustable in the entire domain, the user may want to rely on this IGP metric to make a best path and forwarding decision, regardless of whether the forwarding happens in native IP or MPLS encapsulation.

- The current implementation on MPLS metric is manually configured in each LSP. There is no dynamic way to tie MPLS metric with IGP metric. Thus, when using MPLS LSP as BGP route outgoing interface, the user loses the ability to tie the forwarding decision with unified IGP metric.

Configuring BGP next-hop IGP cost

When this command is issued, BGP decides purely based on IGP cost to BGP next hop. After the decision is made, BGP checks when an MPLS LSP is present, and totally ignores the LSP metric.

```
compare-lsp-metric Compare metric value among LSP ECMP paths
follow-igp Use IGP metric and ignore LSP metric
```

Syntax: [no] next-hop-mpls follow-igp

This command becomes mutually exclusive with **next-hop-mpls compare-lsp-metric** command, since it is designed by having BGP to ignore all LSP metric value, whether it is specified or by default. The reason is simple, when this is configured, LSP metric is totally ignored; and when **compare-lsp-metric** is configured, the LSP metric cannot be ignored.

Also, this new CLI command takes effect immediately and automatically. A **clear** command is not required, and BGP automatically re-scans its next hop and changes the decision process outcome accordingly.

Displaying show commands

These commands display the configuration.

```
device(config)# show running config
device(config)# show ip bgp config
```

To check BGP next hop resolution and the IGP cost for the next hop, use this show command.

```
device(config)# show ip bgp next-hop
```

This command checks the RTM entry cost value to determine whether BGP next hop resolution takes the IGP cost value, compare to MPLS LSP metric value.

```
device(config)# show ip route
```

Creating OSPF shortcuts over an LSP tunnel

This feature allows the user to forward traffic to destinations within an OSPF routing domain through an LSP tunnel, which optimizes available bandwidth by choosing LSPs where multiple paths exist to the same OSPF destination. When an LSP is configured as an OSPF shortcut, OSPF includes the LSP in the SPF calculation. When OSPF determines that the LSP shortcut is the best path to a destination, it adds a route to the IP routing table, specifying the LSP tunnel interface as the outbound interface, along with the cost of the LSP. Only LSPs configured to router ID are considered as shortcuts. When the LSP goes down or is administratively disabled, or the **shortcuts ospf** command is removed from the configuration, the LSP tunnel route is removed from the main routing table.

LSPs used for this feature must originate and terminate within the same OSPF area. When configured, OSPF directs routes that are reachable from the egress router of a shortcut-enabled LSP to an LSP tunnel as the outgoing interface.

To configure this feature, point the LSP to the router ID of the egress router where traffic is forwarded. The user must also configure the LSP with the **shortcuts ospf** command.

The following configuration of LSP "tunnel1" specifies the egress router with a router ID of 10.2.2.2 and enables it for OSPF shortcuts.

```
device(config)# router mpls
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp)# to 10.2.2.2
device(config-mpls-lsp)# shortcuts ospf
device(config-mpls-lsp)# enable
```

Syntax: [no] shortcuts ospf

This feature points OSPF routes to routes from the configured egress router of the LSP tunnel. By way of the LSP interface, the ingress router points to routes on the egress router (including downstream external or summary routes). To view these routes, enter the **show ip route** command as shown in the following example.

```
device# show ip route
Total number of IP routes: 5
Type Codes - B:BGP D: Connected I:ISIS S: Static R:RIP O: OSPF;Cost - Dist/Metric
     Destination      Gateway     Port        Cost      Type
1    10.2.2.0/24      10.2.2.2    lsp tunnel1 110/10    O2
2    10.5.5.0/24      10.1.1.2    eth 1/1     110/2     O
3    10.15.15.15/32   10.3.3.3    lsp l1      110/10    O2
4    10.0.0.0/8       10.1.1.2    eth 1/1     110/10    O2
5    192.85.1.0/24    10.1.1.2    eth 1/1     110/2     O
```

In this example, Type "O2" routes are OSPF routes from outside the OSPF area.

The user can set the next hop for a static route to the egress router of an LSP tunnel when the destination route is contained in the MPLS routing table.

IS-IS shortcuts

This section describes IS-IS shortcuts and how to configure them on an MPLS router with *Traffic Engineering (TE)* capabilities.

Overview

The IS-IS shortcuts feature enables an MPLS TE path (LSP tunnel) to serve as a shortcut through the network to a destination based on the cost of the path (metric). Traffic is forwarded through the LSP tunnel to destinations within the IS-IS routing domain. This feature helps optimize available bandwidth by choosing paths using LSPs where multiple paths exist to the same destination.

When IS-IS shortcuts are enabled on an LSP tunnel, IS-IS includes the LSP in the SPF calculation. When IS-IS determines that the LSP shortcut is the best path to a destination, it adds the route to the IP routing table, specifying the LSP tunnel interface as the outbound interface, including the cost of the LSP. Only LSPs configured to a router ID are considered as shortcuts. When the LSP goes down or is administratively disabled, or when the **shortcuts isis** command is removed from the configuration, the IS-IS LSP tunnel routes are removed from the main routing table.

Determining the cost of an IS-IS shortcut

IS-IS uses the following information to determine the cost of an IS-IS shortcut:

- The **announce metric**, when announce is enabled.

- When announce is not enabled and **ignore-lsp-metric** is not configured, IS-IS uses the LSP metric configured under the LSP configuration mode, for example:

```
device(config-mpls-lsp)# metric value
```

- When no LSP metric is configured, IS-IS uses the native IGP cost, plus or minus the **relative metric**.
- When there is no relative metric, IS-IS uses the native IGP cost.

The announce metric and relative metric are described in detail in the following sections. Use the **show isis shortcuts** command to display the metric used to determine the cost.

The announce metric

When IS-IS shortcuts are enabled on an LSP tunnel, the MPLS router does not announce (advertise) the IS-IS shortcuts unless specifically configured to do so. When **announce** is enabled, the user can optionally specify an announce metric, which is used to compute the LSP cost of the IS-IS shortcut. When an announce metric is not explicitly configured, IS-IS uses a default metric value of 10.

To configure an announce metric for IS-IS shortcuts, refer to Configuring the announce metric on page 399.

The relative metric

When announce is not enabled and an LSP metric is not explicitly configured under the LSP configuration mode of the CLI, the **relative metric** is used to compute the LSP cost, which is the native IGP cost, plus or minus the relative metric.

The relative metric is optionally specified when IS-IS shortcuts are enabled, and is used to make an LSP tunnel less or more preferred over other paths. The default relative metric value is zero (0), but can be configured to be a positive or negative number. A positive number disables an LSP tunnel from participating in the SPF calculation. A negative number ensures that the LSP tunnel is preferred over native IGP paths in the SPF calculation. Figure 70 shows an example of this configuration.

FIGURE 70 SPF calculation adjustment using the relative metric
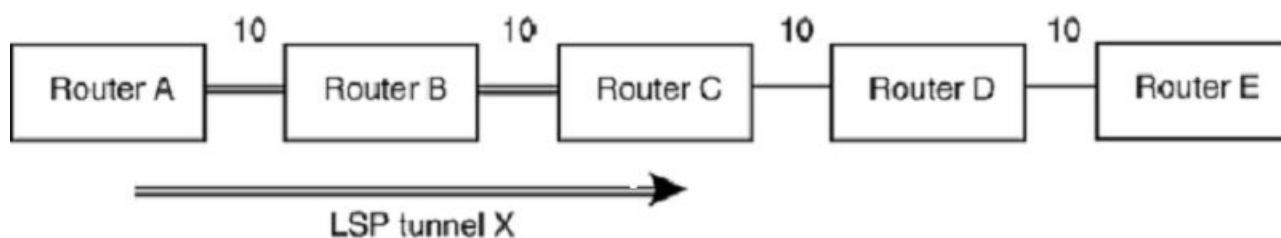
In this example, when there are no IS-IS shortcuts, Router A adds routes in the routing table for routers C, D, and E with the metrics 20, 30, and 40, respectively. When an IS-IS shortcut is configured on LSP tunnel X and the relative metric is -5 (minus 5), Router A installs the same routes in the routing table with the metrics 15, 25, and 35, respectively, over the LSP tunnel X.

To configure the device to use a relative metric value other than zero (0), refer to Configuring the relative metric on page 399.

IS-IS shortcuts over an LSP tunnel

Refer to IS-IS shortcuts on page 396 for details about creating IS-IS shortcuts over an LSP tunnel.

Why LSP tunnels may be excluded from the SPF calculation

LSP tunnels may be excluded in SPF calculations in the following cases:

- The system did not find mapping between the LSP tunnel destination (the 'To' address) and the IS-IS system ID
- There is no IS-IS native route to the LSP tunnel destination
- The IS-IS native route has a better metric than the LSP tunnel
- Another shortcut has a better metric than the LSP tunnel

Configuration notes

Consider the following configuration notes:

- IS-IS shortcuts require MPLS and IS-IS *Traffic Engineering (TE)* to be enabled
- IS-IS does not use an LSP tunnel as a shortcut when the 'To' address of the tunnel is not the router ID of the destination router
- Where multiple IS-IS shortcuts have the same cost, IS-IS installs LSP tunnel-based ECMP routes

Configuration tasks

It is recommended that the user performs the configuration tasks in the order listed in Table 44.

TABLE 44 Configuration tasks for IS-IS shortcuts

Configuration task	Default behavior	See...
1. Enable IS-IS shortcuts	Disabled	Enabling and disabling IS-IS shortcuts on page 398
2. Optionally enable announce on the LSP	Disabled	Enabling IS-IS shortcut advertisements on page 399
3. When announce is enabled, optionally configure the announce metric	When announce is enabled, the system uses either the default metric value of 10, or the explicitly-configured announce metric value	Configuring the announce metric on page 399
4. Optionally configure the relative metric	The default value is zero (0).	Configuring the relative metric on page 399

After performing the configuration steps listed in Table 44, the user can observe the IS-IS routes that use IGP shortcuts. For more information, refer to Show command support on page 402.

Enabling and disabling IS-IS shortcuts

To enable IS-IS shortcuts on an LSP tunnel, enter commands such as the following, starting at the MPLS level of the CLI.

```
device(config-mpls)# lsp tomu3
device(config-mpls-lsp-tomu3)# shortcuts isis level2
device(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

These commands enable IS-IS shortcuts on the **tomu3** LSP tunnel.

For additional information regarding this command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Enabling IS-IS shortcut advertisements

When announce is enabled, the tunnel information is advertised in an IS neighbor TLV, which is stored in the IS-IS database.

To enable announce, enter the following command on an LSP that is not yet enabled.

```
device(config-mpls-lsp-tomu3)# shortcuts isis level2 announce
```

When the tunnel is enabled, disable it before enabling announce, then re-enable the tunnel. For example:

```
device(config-mpls-lsp-tomu3)# disable
Disconnecting signaled LSP tomu3
device(config-mpls-lsp-tomu3)# shortcuts isis level2 announce
device(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

These commands enable the system to advertise IS-IS shortcuts. Since an announce metric is not explicitly specified in this example, IS-IS uses the default announce metric of 10. To configure an announce metric other than 10, refer to Configuring the announce metric on page 399.

Syntax: [no] shortcuts isis [level1 | level2] announce

Enter the [**no**] form of the command to disable advertisement of IS-IS shortcuts. IS-IS shortcuts are still enabled, but are no longer advertised in the IS-IS database.

Configuring the announce metric

The announce metric is described in The announce metric on page 397.

To configure an announce metric, enter a command such as the following at the MPLS LSP level of the CLI.

```
device(config-mpls-lsp-tomu3)# shortcuts isis level2 announce announce-metric 20
```

Syntax: [no] shortcuts isis [level1 | level2] announce announce-metric *num*

Enter the [**no**] form of the command to return to the default announce metric value of 10. IS-IS shortcuts are still enabled, however the [**no**] form of the command simply reverts to the default announce metric.

For *num* , enter a value from 1 - 16777215. The default is 10.

The announce metric is displayed in the output of the **show isis shortcuts** command. When the LSP tunnel is not announced, a '-' (dash) is displayed in the announce metric field.

Configuring the relative metric

The relative metric is described in The relative metric on page 397.

When announce is not enabled and a metric is not explicitly configured under the LSP configuration mode of the CLI, the **relative metric** is used to compute the shortcut cost.

To configure the relative metric, enter a command such as the following at the MPLS LSP level of the CLI.

```
device(config-mpls-lsp-tomu3)# shortcuts isis level2 relative-metric -5
```

This command sets the relative metric value to -5 (minus 5). The LSP cost is determined by subtracting 5 from the native IGP cost to reach the tunnel destination. Using this example, when the native IGP cost is 10, the relative metric value -5 sets the LSP cost to 5.

> **NOTE**
> The shortcut cost is never a value less than one. For example, when the native IGP cost is 10 and the relative metric is -15, the shortcut cost is one, not -5.

Syntax: [no] shortcuts isis [level1 | level2] relative-metric [+ | -] *num*

Enter the [**no**] form of the command to return to the default native IGP path metric. IS-IS shortcuts are still enabled. The [**no**] form of the command simply removes the relative-metric value from the configuration.

The '**+**' or '**-**' sign is required. The '**+**' denotes a positive number. '**-**' denotes a negative number.

For *num* , enter a value from 1 - 16777215. The default is zero (0).

The metric used in the SPF calculation is displayed in the output of the **show isis shortcuts** command. When the LSP tunnel is not used in the SPF calculation, a '**-**' (dash) is displayed in the SPF metric field.

Example configurations

This section includes example configurations and relevant show command outputs both before and after IS-IS shortcuts are enabled.

The following display shows an IS-IS route configuration *before* IS-IS shortcuts are installed.

```
device(config-mpls-lsp-tomu3)# show ip route isis
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
    Destination    Gateway    Port      Cost     Type Uptime
1   0.0.0.0/0      10.1.1.2   eth 1/1   115/21   IL2  0m23s
2   10.2.1.0/24    10.1.1.2   eth 1/1   115/20   IL2  0m23s
3   10.3.1.0/24    10.1.1.2   eth 1/1   115/10   IL2  0m23s
4   10.4.1.1/32    10.1.1.2   eth 1/1   115/10   IL2  0m23s
5   10.2.2.2/32    10.1.1.2   eth 1/1   115/10   IL2  0m23s
6   10.1.0.0/16    DIRECT     drop      115/10   IL1  3m37s
7   10.2.1.1/32    10.1.1.2   eth 1/1   115/20   IL2  0m23s
```

The following example shows IS-IS shortcut configuration.

```
device(config-mpls)# lsp tomu3
device(config-mpls-lsp-tomu3)# metric 1
device(config-mpls-lsp-tomu3)# shortcuts isis level2
device(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

The following display shows the IS-IS route configuration *after* the shortcut configuration is applied. The bold text indicates that the routes are now using shortcuts. Compare this output with the output generated before the shortcut configuration was applied.

```
device1(config-mpls)# show ip route isis
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
    Destination    Gateway    Port       Cost      Type Uptime
1   0.0.0.0/0      10.1.1.1   lsp tomu3  115/2   IL2  0m2s
2   10.2.1.0/24    10.1.1.1   lsp tomu3  115/11  IL2  0m2s
3   10.3.1.0/24    10.1.1.1   lsp tomu3  115/1   IL2  0m2s
4   10.4.1.1/32    10.1.1.1   lsp tomu3  115/1   IL2  0m2s
5   10.2.2.2/32    10.1.1.1   lsp tomu3  115/1   IL2  0m2s
6   10.1.0.0/16    DIRECT     drop       115/10  IL1  3m54s
7   10.2.1.1/32    10.1.1.1   lsp tomu3  115/1   IL2  0m2s
```

The following example shows an IS-IS shortcut configuration with route advertisements enabled.

```
device(config-mpls)# lsp tomu3
device(config-mpls-lsp-tomu3)# disable
Disconnecting signaled LSP tomu3
device(config-mpls-lsp-tomu3)# shortcuts isis level2 announce
device(config-mpls-lsp-tomu3)# enable
Connecting signaled LSP tomu3
```

In the output for this configuration, the bold text indicates that the device uses the extended TLV fields to advertise the shortcut in an IS adjacency TLV. when route advertisements are not enabled, this text would not appear in the output.

```
device(config-mpls)# show isis database mu1.00-00 detail
IS-IS Level-2 Link State Database
LSPID         Seq Num     Checksum  Holdtime  ATT/P/OL
mu1.00-00*    0x00000010  0xd938    35        1/0/0
   Area Address:  47
   NLPID:  IPv6 IP
   Hostname:  mu1
   TE Router ID: 10.1.1.1
   Metric:  10    IP-Extended 10.1.1.0/24      Up: 0 Subtlv: 0
   Metric:  10    IP-Extended 10.2.1.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.2.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.3.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.4.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.5.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.6.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.7.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.8.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.9.0/24      Up: 0 Subtlv: 0
   Metric:  1     IP-Extended 10.1.10.0/24     Up: 0 Subtlv: 0
   Metric:  10    IP-Extended 10.1.0.0/16      Up: 0 Subtlv: 0
   Metric:  10    IPv6 Reachability 1000::/32  Up: 0 Subtlv: 0
   Metric:  10    IPv6 Reachability 2000::/32  Up: 0 Subtlv: 0
   Metric:  10    IS-Extended mu1.02
     Admin Group: 0x00000000
     Interface IP Address: 10.1.1.1
     Link BW: 10000000 kbits/sec
     Reservable BW: 10000000 kbits/sec
     Unreserved BW:
       [0]   10000000 kbits/sec  [1]   10000000 kbits/sec
       [2]   10000000 kbits/sec  [3]   10000000 kbits/sec
       [4]   10000000 kbits/sec  [5]   10000000 kbits/sec
       [6]   10000000 kbits/sec  [7]   10000000 kbits/sec
     Admin Group: 0x00000000
     Interface IP Address: 10.1.1.1
     Neighbor IP Address: 10.1.1.2
     Link BW: 10000000 kbits/sec
     Reservable BW: 8000000 kbits/sec
     Unreserved BW:
       [0]   8000000 kbits/sec  [1]    8000000 kbits/sec
       [2]   8000000 kbits/sec  [3]    8000000 kbits/sec
       [4]   8000000 kbits/sec  [5]    8000000 kbits/sec
       [6]   8000000 kbits/sec  [7]    8000000 kbits/sec
   Metric:  10       IS-Extended mu3.00
```

Clearing IS-IS shortcuts

When the user clears IS-IS shortcuts, IS-IS attempts to re-map the LSP To address to IS-IS system ID. Clearing shortcuts is useful when the mapping between the To address and System ID must be refreshed once the LSP tunnel is being used in the SPF calculation.

> NOTE
> This is not a common operation.

To clear IS-IS shortcuts from the configuration, use one of the following CLI commands at any level of the CLI:

- **clear isis shortcut** - This command clears all IS-IS shortcuts from the configuration.
- **clear isis shortcut lsp** *lsp-name* - This command clears IS-IS shortcuts for the specified LSP.

Syntax: **clear isis shortcut** [**lsp** *lsp-name*]

Ignore LSP metric

The Ignore LSP Metric feature, when enabled, forces IGP protocols not to use configured LSP metric values for IS-IS and OSPF shortcuts when performing SFP calculations. Enabling this feature causes the shortcut's effective metric to be derived by summing up all the path's cost spanned over by the shortcut. By ignoring the LSP metric, IGP such as IS-IS, can just run Incremental Shortcut SPF instead of full SPF calculation when an LSP goes up or down since the network topology is not changed.

This feature can only be enabled on routers when the IGP shortcut is configured. The feature is disabled by default.

> **NOTE**
> The LSP must be disabled before enabling or disabling this feature.

To enable the Ignore LSP metric feature, disable the LSP and enter a command such as the following at the MPLS LSP level of the CLI.

For IS-IS:

```
device(config-mpls-lsp-tomu3)# shortcut isis level2 ignore-lsp-metric
```

Syntax: [no] shortcut isis [level1 | level2] [ignore-lsp-metric] [announce [announce-metric *value*]] [relative-metric [+ | -]] *value*

The '**+**' or '**-**' sign is required. The '**+**' denotes a positive number. The '**-**' denotes a negative number.

For *num* , enter a value from 1 - 16777215. The default is zero (0).

Enter the [**no**] form of the command to return to using the configured LSP metric as the shortcut's cost when performing IS-IS SFP calculation.

For OSPF:

```
device(config-mpls-lsp-tomu3)# shortcut ospf ignore-lsp-metric
```

Syntax: [no] shortcut ospf [ignore-lsp-metric]

Enter the [**no**] form of the command to return to using the configured LSP metric as the shortcut's cost when performing OSPF SFP calculation.

The following example shows a sample configuration of the Ignores LSP metric feature:

```
device2(config-mpls)# lsp lsp100
device2(config-mpls-lsp-lsp100)# disable
Disconnecting signaled LSP lsp100
device2(config-mpls-lsp-lsp100)# shortcut isis level2 ignore-lsp-metric
device2(config-mpls-lsp-lsp100)# enable
Connecting signaled LSP lsp100
device2(config-mpls)# show isis shortcut detail
Configured:1 Up: 1, Announced: 0
L2 lsp lsp100
 To 10.1.1.3, Used by SPF (10), Not Announced (Announce not configured)
 ISIS System Id for 10.1.1.3 is R3.00-00
 LSP Metric: 8(Ignored), Relative Metric: 0, Announce Metric: -
 Last notification from MPLS received 763d17h53m ago
device2(config-mpls)#
```

Show command support

Use the following show commands to display information about IS-IS shortcuts:

*   **show isis shortcuts** - Displays information about all IS-IS shortcuts configured on the device.
*   **show isis shortcuts lsp** *lsp-name* - Displays information about all IS-IS shortcuts configured for a specified LSP.
*   **show isis shortcuts detail** - Displays detailed information about all IS-IS shortcuts that are UP, such as the system ID and matching To address of the tunnel, configured metric values, and the time period for which the LSP has been an IS-IS shortcut.

- **show isis shortcuts lsp** *lsp-name* **detail** - Displays detailed information about all IS-IS shortcuts for a specified LSP, such as the system ID and matching To address of the tunnel, configured metric values, and the time period for which the LSP has been an IS-IS shortcut. This command also displays whether IGP Ignore LSP metric is enabled.
- **show isis** - Enhanced output indicates whether or not IS-IS shortcuts are configured, the number of shortcuts configured, how many are UP, and how many are advertised.
- **show isis debug** - Shows debugging information for IS-IS shortcuts. For more information, refer to the diagnostic reference guide.

   NOTE
   Only LSPs that are UP (administratively and operationally enabled in the MPLS domain) are kept in the database and displayed in the show command outputs. LSPs that are down are not kept in the database and are not displayed in the command outputs.

   NOTE
   There is no show command for OSPF shortcut.

show isis shortcut

Displays information about all IS-IS shortcuts configured on the device.

Syntax

show isis shortcut

show isis shortcut detail

show isis shortcut lsp *name* **detail**

Parameters

detail
Specifies detailed information.

lsp *name*
Specifies a link-state packet (LSP).

Modes

User EXEC mode.

Usage Guidelines

Only LSPs that are UP (administratively and operationally enabled in the MPLS domain) are kept in the database and displayed in the show command outputs. LSPs that are down are not kept in the database and are not displayed in the command outputs.

This command also operates in all modes.

Command Output

The **show isis shortcut** command displays the following information:

Output field	Description
Configured	The number of IS-IS shortcuts configured.
Up	The number of IS-IS shortcuts that are UP.
Announced	The number of IS-IS shortcuts that are advertised.
Name	The name of the IS-IS shortcut. When the name is longer than 11 characters, it wraps to the next line.
To	The LSP endpoint address.
Metric (SPF or Announce)	The metric used in the SPF calculation or the metric used in the advertisement of the IS adjacency TLV.
	The SPF metric can be one of the following:
	• The metric configured at the MPLS LSP configuration level.
	• The native IGP metric plus or minus (+ or –) the relative metric configured with the **shortcuts isis** command.
	• The native IGP metric
	• A dash (–) denotes that the tunnel is not used in SPF calculations.

Output field	Description
	The Announce metric can be one of the following: • 10 (the default announce metric) • The metric configured with the **announce-metric** keyword • A dash (-) denotes that the tunnel is not used in the IS adjacency TLV advertisement.
Announce	Indicates whether or not IS-IS shortcuts are advertised: • Yes - IS-IS shortcuts are advertised • No - IS-IS shortcuts are not advertised.
Tunnel Intf	The tunnel index of the LSP. This is assigned by MPLS whenever an LSP is created.

Examples

The following example shows the output of the **show isis shortcut** command.

```
device# show isis shortcut
Configured: 3, Up: 2, Announced: 1
Name         To            Metric          Announce   Tunnel
                           (SPF/Announce)             Intf
lsp tomu2    10.4.1.1      10/-            No         tnl1
lsp tomu3    10.3.1.1      -/-             Yes        tnl2
lsp toolong 10.20.1.1      10/10           Yes        tnl3
toreachmu3
```

The following example shows the **show isis shortcut detail** command.

```
device# show isis shortcut lsp tomu2 detail
lsp tomu2
 To 10.1.1.1, Used by SPF (10), Not Announced
 LSP metric: 10, Relative metric: -, Announce metric: -
 ISIS System Id for 10.4.1.1. is mu2.00-00
 Not announced due to configuration
Last notification from MPLS received 0hhm35s ago.
```

Displaying detailed information about IS-IS shortcuts

The **show isis shortcuts detail** command displays detailed information about IS-IS shortcuts, including:

- The system ID and matching 'To' address of the tunnel.
- Configured metric values.
- How long the LSP has been an IS-IS shortcut.

For additional information, see the CLI command page for the **show isis shortcuts detail** commands in *NetIron Command Line Interface (CLI) Reference Guide*.

Displaying IS-IS shortcut statistics

The **show isis** command output includes the following information about IS-IS shortcuts:

- Whether or not IS-IS shortcuts are enabled.
- The number of IS-IS shortcuts configured.
- How many IS-IS shortcuts are UP.
- How many IS-IS shortcuts are advertised.

The information is displayed at the bottom of the **show isis** display output. For example:

```
device# show isis
(truncated for brevity)...
ISIS Shortcuts: 20 configured, 10 are up, and 10 are announced
```

Or

```
device# show isis
(truncated for brevity)...
No isis shortcuts configured
```

ECMP forwarding for IP over MPLS

ECMP hardware forwarding is supported for IP over MPLS packets when an outgoing interface is configured as a physical port *and* a VE interface, or configured on an MPLS tunnel. When multiple routes use ECMP to reach a destination, hardware ECMP is automatically enabled. ECMP load sharing for IP over MPLS is supported for 2-8 tunnels, with a default of four tunnels.

XMR Series and MLX Series devices support ECMP hardware forwarding only in static CAM mode. ECMP hardware forwarding is not supported for dynamic mode. Hitless upgrade for ECMP hardware forwarding is not supported. ECMP hardware forwarding is not supported on CES 2000 Series and CER 2000 Series devices.

For ECMP hardware forwarding, all outgoing interface paths must configured in the same VRF, and must belong to an MPLS tunnel. A hash value is computed for a packet when it is received by XPP. XPP uses the hash value to select a PRAM that forwards the packet to the destination. An ECMP PRAM block consists of eight PRAMs. The hash value for each outgoing packet on a customer edge router interface is calculated based on source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, and TCP or UDP destination port.

The hash value for each incoming packet on the route target is calculated based on the source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, TCP or UDP destination port, and a VC label for an MPLS packet.

Handling IS-IS-overload-bit in MPLS

Glossary of acronyms

Acronym	Meaning
CLI	Command Line Interface
CSPF	Constrained Shortest Path First
IGP	Interior Gateway Protocol
IS-IS	Intermediate System to Intermediate System
LDP	Label Distribution Protocol
LSP	Label Switch Path
MPLS	Multi-Protocol Label Switching
OSPF	Open Shortest Path First
PLR	Point of Local Repair
QOS	Quality Of Service
RSVP	Resource ReSerVation Protocol
TE	Traffic-Engineering

Introduction

IS-IS overload bit is a special bit set in IS-IS-LSP to indicate that the advertising router is not yet ready to forward transit traffic.

The overload bit was originally used to indicate the resource shortage to the network. When overload bit is set on a specific router, it effectively signifies to other routers in the network to not use it as a transit hop in their SPF calculations.

Some of the overload bit use cases:

- To verify operation of new installed routers before allowing them to forward transit traffic.
- Preventing control plane routers like (Route Reflectors) from being used accidentally in the forwarding path.
- On routers' start-up to avoid traffic black-holes until routing protocols (like BGP) are fully converged.
- To isolate a specific *router before decommissioning on* a maintenance operation.

MPLS-TE reacts to the IS-IS overload bit when it is set on a router. IS-IS notifies this information to MPLS and it is stored in the TE database. This feature fixes some of the current behaviors for handling overload bit by MPLS TE and also gives additional options for the user to control the behavior precisely.

1. This feature changes the behavior to only rejecting paths if the overload bit is set for a router in the path which acts as transit. Therefore, egress routers can have overload bit set and still be in the LSPs CSPF path. Behavior change is so it is concurrent with the IS-IS overload bit definition.
2. This feature handles the overloaded router condition when a specific path is configured specifically for the LSP by rejecting the path when the overloaded router is one of the hops. Here, the hop could be strict or loose. This can overridden with a new CLI.
3. Part of RSVP-IGP sync phase-2, the feature acts on sessions from ingress only when overload bit is set on a transit router with respect to the LSP. Even when overloaded bit is on egress router with respect to LSP, from ingress LSP is not acted upon.
4. A new CLI is available to override the behavior to ignore the overloaded router by CSPF. The new CLI allows computation to pass for newly coming up LSPs.

Overriding the overload bit behavior

All the behavior with respect to the overload bit *for future sessions can be overridden with this* CLI at the policy level to ignore the overload bit at the transit router.

```
device(config-mpls-policy)# cspf-computation-mode ?
  use-bypass-liberal    use liberal mode for CSPF facility backup computation
  use-bypass-metric     use bypass LSP's path cost for selection between bypass LSP's
  use-igp-metric        use IGP metric of the link for CSPF computation
  use-te-metric         use TE metric of the link for CSPF computation
  ignore-overload-bit   ignore ISIS overload bit during cspf computation

device(config-mpls-policy)# cspf-computation-mode ignore-overload-bit <cr>
```

- With this enabled, even when overload bit is set on a transit a router, the CSPF at the ingress does not reject any path for new LSPs.
- If the ignore overload bit is set, the already existing transit sessions are not brought DOWN from ingress on enabling overload bit on transit router.

Behaviors when ignoring the overload bit

IS-IS reacts to the overload bit by removing the route reachable through an overloaded router from the routing table. This does not mean MPLS TED links are also removed. This only has an impact on the MPLS in the CSPF calculation. The CSPF always tries to resolve the "to" address at the start of the CSPF calculation. If the resolve fails, the CSPF stops right there and the error displays "No route to destination". This behavior effects how the ignore-overload-bit is handled. See the example shown below:
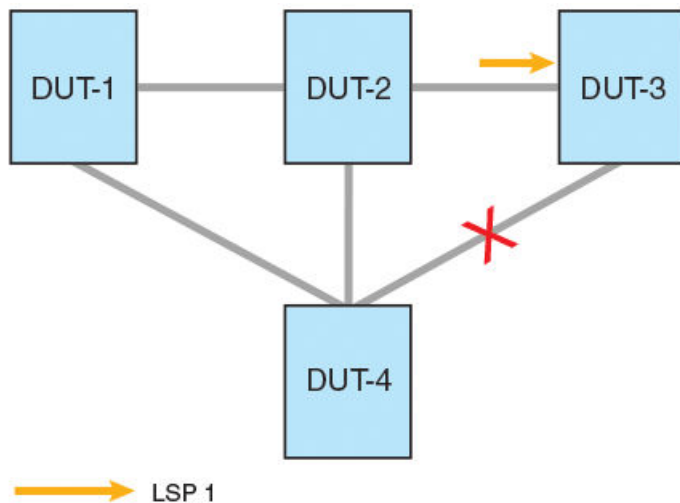
- Consider in the above example the overload bit is set on device2.
- Imagine a strict hop LSP trying to come UP from device1 through device2 though device3.
- The ignore overload bit is set on device1.

Since the overload bit is set on device2, device3s reachability by device1 is based on:

1. If there is link between device4 and device3, then device3 is reachable from device1 by way of device4 with respect to the IP routing point of view. Thus, in this case, the LSP will come up through device1->device2->device3 (strict hop configured) because ignore overload bit is set on ingress. Here the LSP comes through device2 even though device3 is not reachable by way of device2 and is reachable only through device4.

2. If there is no link from device4 to device3. In this case, LSP does not come UP even though ignore overload bit is set on ingress because there is no reachability for device3 through any path in the IP routing table.

The behavior is valid and user must carefully consider these cases before using the **ignore overload bit** at ingress. The reason is because the MPLS always decides the route on its own without the discretion of the IGP even though it gets information from the IGP. The decision to use a particular path is decided by the user from the MPLS point of view by not considering the path derived from the routing table.

Future sessions on the overloaded router

This behavior is currently partly handled. This does not need the RSVP-IGP sync feature to be enabled. Additional behavioral changes are incorporated with this feature to make it consistent *with the IS-IS overload bit definition* mentioned below.

When the overloaded bit is set on the ingress router

When a new session is trying to come UP from overloaded router acting as ingress, CSPF does not reject the path because it is the responsibility of the user to decide whether there is need for a new LSP from the overloaded router.

When the overloaded bit is set on the egress router

When the egress router of the new session is overloaded, CSPF does not reject the path because it is the responsibility of the user configuring the LSP to decide, as the traffic is for the egress router instead of it forwarding it to some other router in the network.

When the overloaded bit is set on a transit router

When a new session that is trying to come UP through an overloaded transit router, CSPF rejects the path because when the overload bit is set, the transit router is not ready to forward the traffic. This is true even when a specific path is configured for the LSP (containing strict or loose hops).

Customer configurations

Existing sessions on the overloaded router
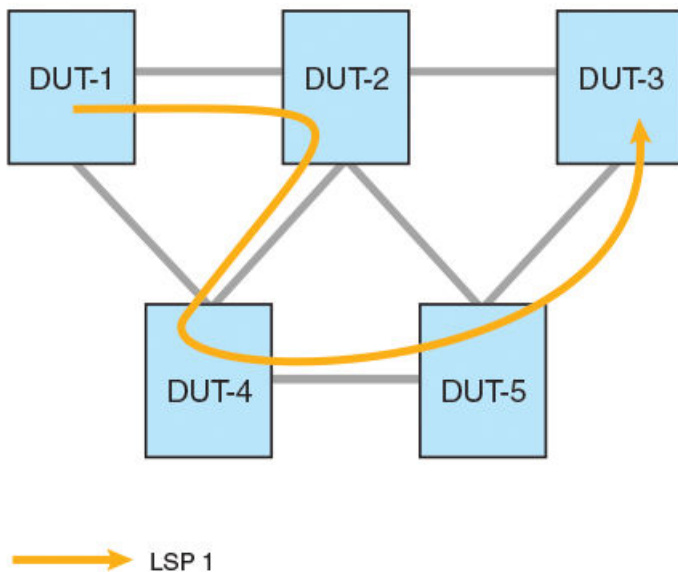
Existing sessions on the overloaded router

This behavior is only valid when RSVP-IGP sync feature is enabled on the router at the policy level. When an overload bit is set on any router in the network, the IGP floods this information across through TLVs. Thus, the TED database in the MPLS enabled routers across the network will have this information.

Ingress and egress sessions on an overloaded router

When an overload bit is set on ingress or egress routers, the decision to bring DOWN the LSP or not that is starting from ingress overloaded router or ending at egress overloaded router is taken at the ingress. These LSPs which are already UP from or to the ingress or egress overloaded router respectively are not brought DOWN. This is to keep up with the definition of the overload bit which indicates to the other routers that it is not ready to forward only the transit traffic. From the above figure, if overload bit is set on device1 or device3, LSP 1 is not brought DOWN from ingress.

Transit sessions on an overloaded router

When an overload bit is set on transit routers, the decision to bring the LSPs that are transiting through this overloaded router is taken at the ingress router. All the transit sessions are brought DOWN and retried in this case. In the case of adaptive LSPs, an MBB is performed. From the above figure, if an overload *bit is set on device2 or device4 or* device5, then LSP 1 is brought DOWN from ingress and is retried again *avoiding the router which is overloaded* .

QoS mapping between IP packets and MPLS

The 3-bit EXP field in the MPLS header can be used to define a *Class of Service (CoS)* value for packets the traverse an LSP. The CoS value specifies a priority for MPLS packets.

There are two ways that a CoS value can be applied to packets that traverse an MPLS network through an LSP:

- A CoS value is manually configured for the LSP. This is the default operation.
- No CoS value is set for an LSP, and the *Type of Service (ToS)* field in the IP header is used. In this situation, the device copies the first three bits in the ToS field of the packet to the CoS (EXP) field in the MPLS header. The ToS value maps to one of the four priority queues on the device.

LDP interface transport address

LDP interface transport address

The LDP interface transport address provides flexibility to the user to configure a different source IP address for the LDP session's TCP connection.

The LDP session establishment between two routers needs a TCP session to be created. Each LDP router must know the transport address of its LDP peer to establish the TCP connection.

As per the LDP protocol specification, the transport address can be explicitly notified using the transport address TLV in the LDP discovery hello messages, or omitted in which case the LDP peers uses the source IP address of the Hello messages as the transport address.

The default behavior of LDP in NetIron is to advertise its LSR-ID as the transport address in LDP hello messages. It uses it as the source address for its TCP connection. The LDP interface transport address provides a mechanism to configure a different IP address to be used for the transport address for the session.

In NetIron, the source IP address for the LDP link discovery messages is that of the interface.

Conditions

Pre-requisites

1. The interface transport address must not be enabled when there is multiple parallel adjacencies (interfaces) present between the LDP routers. When the user wishes to enable the interface transport address, then the user must remove the additional adjacencies. Disabling any additional adjacencies does not have any impact to the LDP ingress ECMP as the interface still is used in the ECMP irrespective of **ldp-enable** configuration being present.

2. When a user enables the interface transport address with multiple adjacencies to a peer, then it is possible that the interface transport address may not be used and the session is torn down due to a role conflict.

3. -Enabling the interface transport address on the interface (adjacency) causes any existing session to flap and come back up with the interface IP address as the transport address (only in cases where there is a single adjacency between the peers). This can be service impacting, and the user must be aware before executing the command.

Upgrade and downgrade

The downgrade is seamless; however, the user may notice the error during configuration replay as the **transport-address interface** configuration fails.

Upgrading to a version which supports this feature, the user has the flexibility to turning the interface transport address on at a later point in time.

Backward compatibility

When peering with an LDP neighbor running older version of NetIron software:

- The user sees no impact when the interface transport address is enabled on the peer with a single adjacency (except for the session bounce).
- With multiple adjacencies, an LDP session role conflict may be detected and tear down the session.

Performance

The number of TCP listen sockets are proportional to the number of interfaces on which this configuration is present. Without the interface transport address, there is only a single TCP listen socket.

Using the interface transport address does not increase the total number of TCP sessions as compared with the previous release.

Security

Use the interface transport address as a security mechanism in cases where the user does not want the TCP packets used for LDP session from being routed.

Setting the interface address

Use the **transport-address interface** command to use the primary IP address of the interface as the transport address instead of the default LSR-ID.

To configure the interface transport address the user must be in the **ldp-params** mode under **mpls-interface** mode within **router mpls configuration** mode. Executing this command causes an existing LDP session on the go down and come back UP (if this was the only adjacency)

```
device>enable
device# configure terminal
device(config)# router mpls
device(config-mpls)# mpls-interface ethernet xx
device(config-mpls-if-xxx-x/x)#ldp-params
device(config-mpls-if-xxx-x/x-ldp-params)#transport-address interface
```

LDP interface transport address_customer configurations

The topology below is referred in the following cases.

Customer sample configuration 1

1. Configure interface transport address on PE-1 for LDP session between PE-1 and P-1.
   - Verify the LDP session comes UP using interface IP address of PE-1.

- Verify the TCP session and also the source IP address of the Hello packets.
- Verify all services on MPLS are functional.

2. Disable the interface transport address feature on PE-1.

- Verify the LDP between PE-1 and P-1 comes back up with LSR-ID as the transport address.

Customer sample configuration 2

Configure interface transport address feature on P-1 for LDP session between P-1 and PE-1.

- Verify the LDP session comes UP using interface IP address of P1. LDP session between P-1 and P-3 and P-1 and P-2 are also UP using the LSR-ID as transport address for P-1.
- Verify all MPLS based services are functional.

Customer sample configuration 3

Repeat customer sample configuration 1 with VE/LAG interfaces.

Customer sample configuration 4

Repeat customer sample configurations 1 and 2 with Cisco and Juniper to test inter-operability.

Customer sample configuration 5

Use customer sample configurations 1 and 2 with different line cards (Generation 1.1, 2, and 3).

IPv6 over MPLS

IPv6 over MPLS overview

IPv6 over Multi-Protocol Label Switching (MPLS) enables propagation of IPv6 routes across an IPv4-based MPLS network by connecting the IPv6 provider edge (6PE) routers with the existing IPv4-based MPLS infrastructure.

IPv6 over MPLS is described in RFC 4798 and this technology is also known as IPv6 Provider Edge Router (6PE) over MPLS. The 6PE router can be an existing PE router or a new PE router dedicated to IPv6 traffic. The 6PE routers run dual stack IPv4 and IPv6. The 6PE router utilizes the existing MPLS cloud to forward IPv6 traffic from the customer edge (CE) routers, based on the labels assigned for each IPv6 packet received from the CE routers.

IPv6 over MPLS architecture

Running IPv6 traffic over Multiprotocol Layer Switching (MPLS) involves IPv6 provider edge (6PE) routers communicating with customer edge (CE) routers and the MPLS domain.

The figure below outlines the IPv6 over MPLS architecture. The CE1 to CE6 routers are either IPv4 or IPv6. The 6PE1 to 6PE4 routers run dual stack IPv4 and IPv6. The CE router provides connectivity with a customer's network and a 6PE router. The CE router advertises IPv6 reachability information from the customer's network using IPv6 static routes, Routing Information Protocol - next generation (RIPng), Intermediate System to Intermediate System routing protocol support for IPv6 (IS-ISv6), or Open Shortest Path First version 3 (OSPFv3) routing protocol. The 6PE router provides connectivity with the CE router and with the MPLS domain. The 6PE router communicates with other 6PE routers using Multi-protocol Border Gateway Protocol (MP-BGP). The outbound packets from a customer's network are forwarded from the CE router to the 6PE router, and the inbound packets are forwarded from the 6PE router to the CE router attached to the customer's network.

FIGURE 71 IPv6 over MPLS
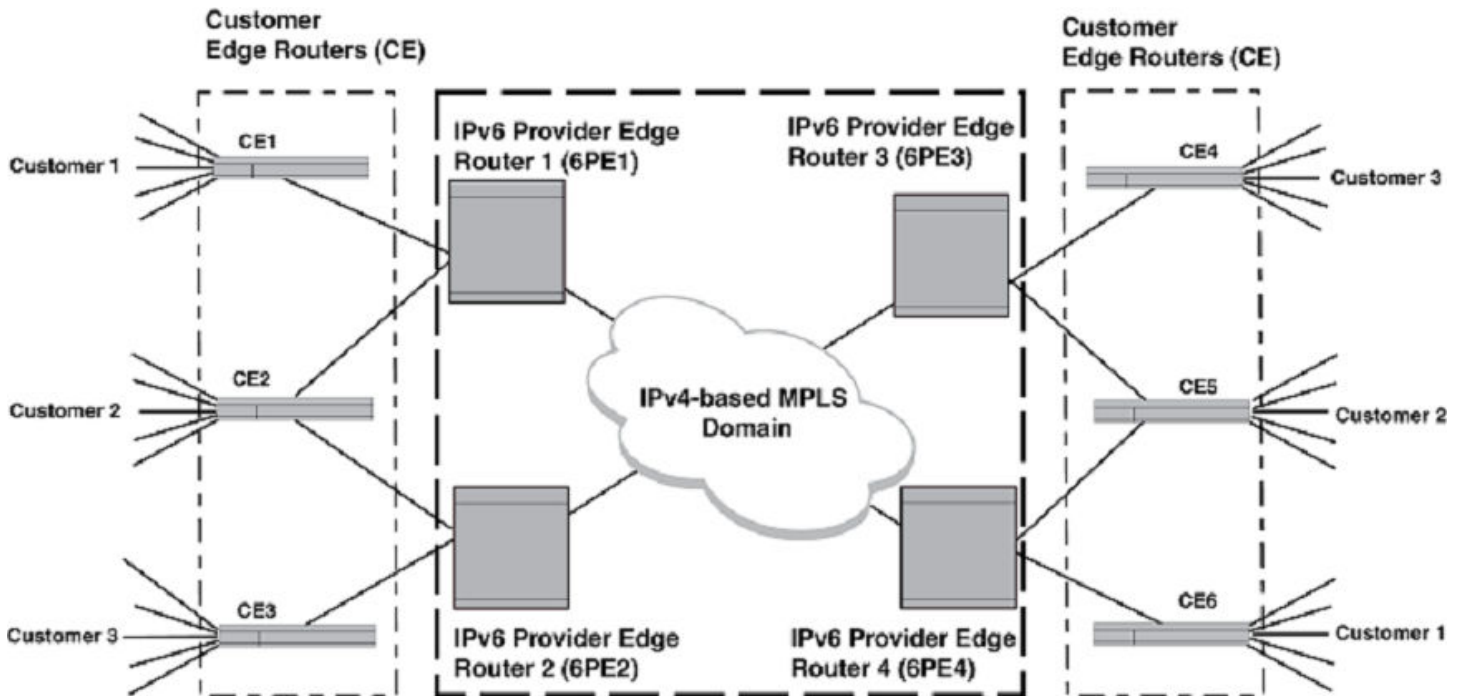
How IPv6 over MPLS works

IPv6 over Multiprotocol Layer Switching (MPLS) operates using an IPv6 provider edge (6PE) router that forwards IPv6 traffic between customer network remote sites utilizing the existing IPv4-signaled Label Switched Path (LSP).

Creating routes in an MPLS domain

In the figure below, IPv6 routes are created in an MPLS domain. A customer edge (CE) router maintains the connection to the customer network and is configured within the network to send or receive IPv6 packets. The IPv6 routes that are available through the CE router are then made available to the 6PE router using RIPng, OSPFv3, Exterior Border Gateway Protocol (EBGP), IS-ISv6, or a static route.

The 6PE router connected to the MPLS domain through one or more interfaces advertises the available IPv6 routes across the MPLS domain to other 6PE routers using MP-BGP. OSPF is used as the Interior Gateway Protocol (IGP) within the MPLS domain to provide connectivity. The 6PE router obtains an LSP required to switch traffic to the other 6PE routers using the Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP). The network is now populated with all the IPv6 routes required to forward IPv6 packets between the customer networks.

FIGURE 72 MPLS route discovery
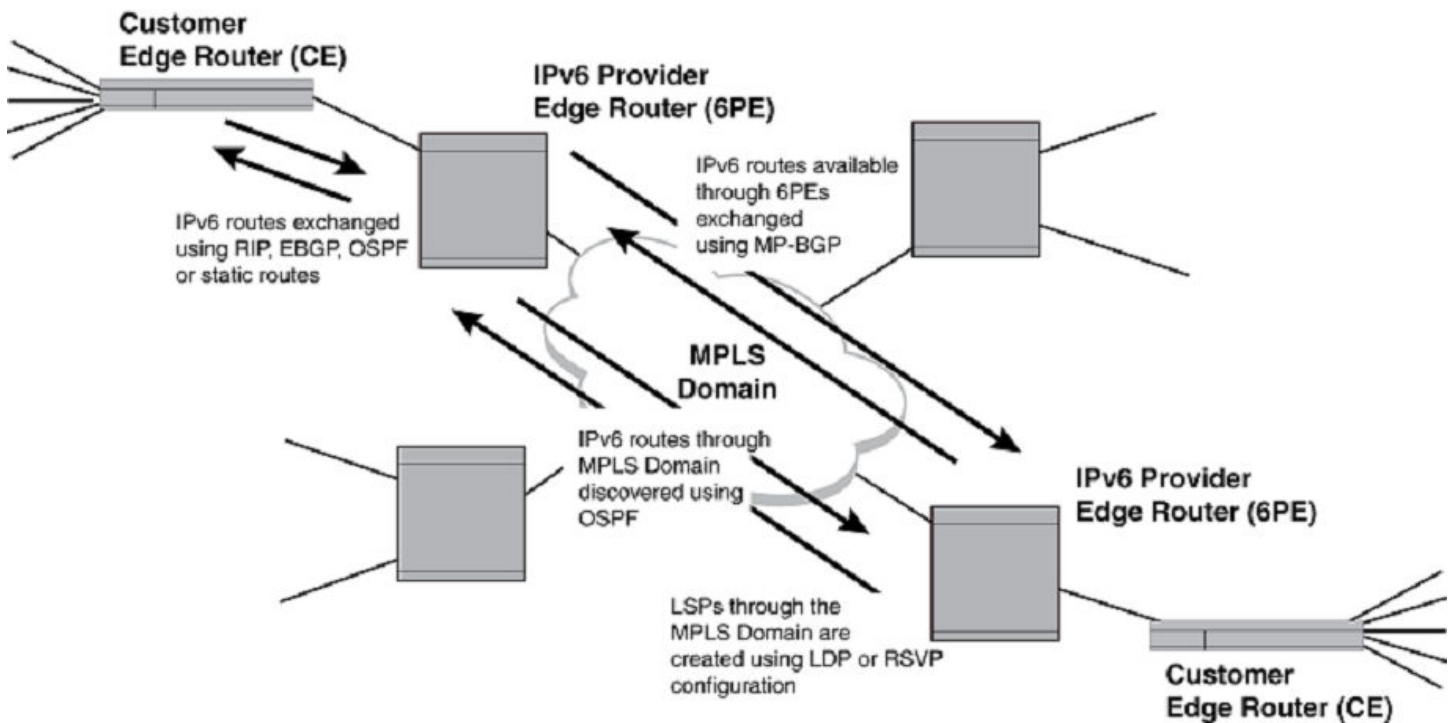
Routing an IPv6 packet through an MPLS domain

The following steps describe how an IPv6 packet is routed through an existing MPLS domain.

1. A 6PE router receives the IPv6 packets from the CE router.

2. A 6PE router assigns labels to all the received IPv6 packets.

3. A 6PE router exchanges the IPv6 packets along with the labels with the other 6PE routers.

4. A6PE router transports the IPv6 packets from the CE router using the existing IPv4 LSPs.

The figure below shows how an IPv6 packet is forwarded from CE1 to CE2 through the MPLS domain. When an ingress 6PE router (6PE-I) receives the IPv6 packet from CE1, the 6PE-I assigns an inner label containing IPv4-mapped IPv6 BGP next-hop information, and an outer label containing the IPv4 address corresponding to the IPv4-mapped IPv6 address to the received IPv6 packet.

The egress 6PE router (6PE-E) advertises the IPv6 reachability information to the 6PE-I by distributing the inner label using the MP-BGP. The 6PE-I obtains an LSP to switch the IPv6 packet to the 6PE-E using LDP or RSVP. The packet is then forwarded through the MPLS domain and is switched using the labels. At the penultimate device in the LSP, the outer label is removed and the packet is forwarded to the 6PE-E. The 6PE-E uses the inner label to identify the destination router (CE2) to which the packet must be forwarded. The 6PE-E removes the inner label and forwards the packet to the CE2.

FIGURE 73 Routing an IPv6 packet through an MPLS domain
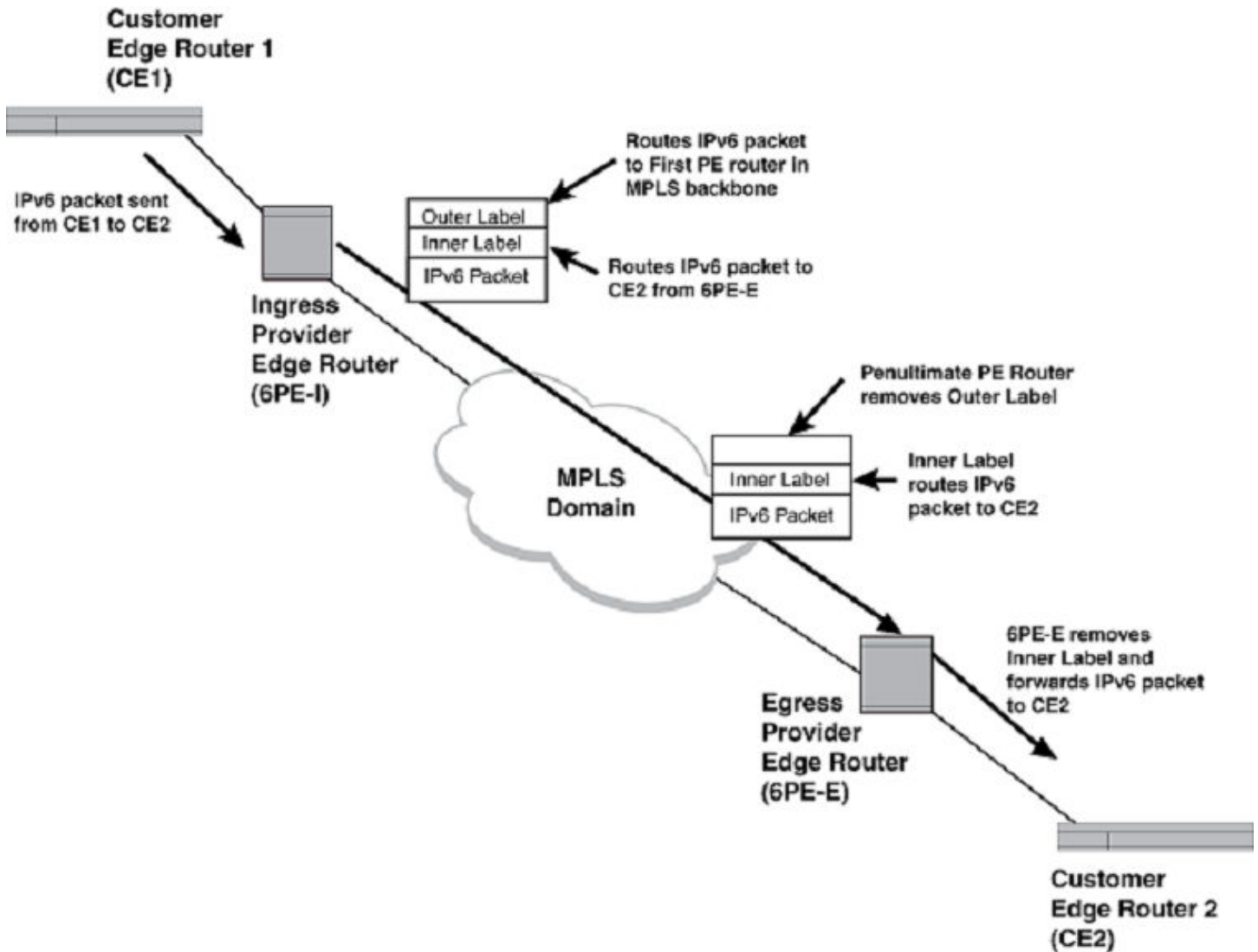
IPv6 over MPLS limitations

IPv6 over Multiprotocol Layer Switching (MPLS) has a few restrictions.

IPv6 over MPLS has the following limitations

- IPv6 over MPLS supports only IPv6 unicast forwarding
- IPv6 over MPLS is supported only in the default VRF
- IPv6 over MPLS does not support GREv6 over v4 tunnel

   NOTE
   Do not forward packets from one type of tunnel to another type of tunnel in the MLX packet processor (XPP). Packets may not be routed properly.

Configuring BGP to enable IPv6 over MPLS

IPv6 over MPLS configuration starts with configuring BGP peers and activating IPv6 unicast and label capability for the BGP peers.

Enabling IPv6 over MPLS

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **router bgp** command to enable BGP routing.

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

3. Enter the **local-as** command to configure the autonomous system number (ASN) in which your device resides.

   ```
   device(config-bgp)# local-as 1
   ```

4. Enter the **neighbor remote-as** command, and specify an IP address, to specify the ASN in which the remote neighbor resides.

   ```
   device(config-bgp)# neighbor 10.40.40.1 remote-as 1
   ```

5. Enter the unicast IPv6 address family configuration mode.

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

6. Enter the **neighbor remote-as** command, and specify an IPv6 address, to specify the ASN in which the remote neighbor resides.

   ```
   device(config-bgp-ipv6u)# neighbor 3ffe:db8:1111::1 remote-as 2
   ```

   This neighbor is in a different ASN at the other end of an MPLS tunnel.

7. Enter the **neighbor activate** command to enable IPv6 unicast capability for the IPv4 peers.

   ```
   device(config-bgp-ipv6u)# neighbor 10.40.40.1 activate
   ```

8. Enter the **neighbor send-label** command to enable IPv6 label capability for the IPv4 peers.

   ```
   device(config-bgp-ipv6u)# neighbor 10.40.40.1 send-label
   ```

9. Return to global configuration mode.

   ```
   device(config-bgp-ipv6u)# exit
   ```

The following example configures BGP to enable IPv6 over MPLS.

```
device# configure terminal
device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.40.40.1 remote-as 1
device(config-bgp)# address-family ipv6 unicast
device(config-bgp-ipv6u)# neighbor 3ffe:db8:1111::1 remote-as 2
device(config-bgp-ipv6u)# neighbor 10.40.40.1 activate
device(config-bgp-ipv6u)# neighbor 10.40.40.1 send-label
device(config-bgp-ipv6u)# exit
```

Use the "Configuring MPLS to enable IPv6 over MPLS" task on the same PE6 router.

Configuring MPLS to enable IPv6 over MPLS

IPv6 over MPLS configuration requires enabling MPLS and creating a tunnel over which the IPv6 traffic can travel.

Perform the Configuring BGP to enable IPv6 over MPLS task to activate IPv6 unicast and label capability for the BGP peers.

An LSP tunnel must be created across which the IPv6 MPLS traffic will be carried. After the tunnel is created, IPv6 over MPLs is enabled.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter router MPLS configuration mode.

   ```
   device(config)# router mpls
   ```

3. Enter LSP configuration mode to configure the LSP tunnel.

   ```
   device(config-mpls)# lsp tunnel 1
   ```

4. Configure the egress router.

   ```
   device(config-mpls-lsp-tunnel1)# to 10.40.40.1
   ```

   In this example, the egress router has a router ID of 10.40.40.1.

5. Configure the ingress router.

   ```
   device(config-mpls-lsp-tunnel1)# from 10.30.30.1
   ```

   In this example, the ingress router has a router ID of 10.30.30.1.

6. Enable the IPv6 over MPLS configuration.

   ```
   device(config-mpls-lsp-tunnel1)# enable
   ```

7. Return to Privileged EXEC mode.

   ```
   device(config-mpls-lsp-tunnel1)# exit
   ```

The following example configures an LSP tunnel and activates IPv6 over MPLS.

```
device# configure terminal
device(config)# router mpls
device(config-mpls)# lsp tunnel 1
device(config-mpls-lsp-tunnel1)# to 10.40.40.1
device(config-mpls-lsp-tunnel1)# from 10.30.30.1
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit
```

Configuration example for IPv6 over MPLS

In the figure, the network is configured to use EBGP to forward routes between the networks attached to the CE routers and the 6PE routers. MP-BGP is used to forward routes between the 6PE routers, and an LSP tunnel is configured across the MPLS domain. The configurations are shown for only the CE1, CE2, 6PE1, and 6PE2 routers to enable propagation of IPv6 packets from CE1 to CE2 through the MPLS domain.

FIGURE 74 Deploying IPv6 over MPLS
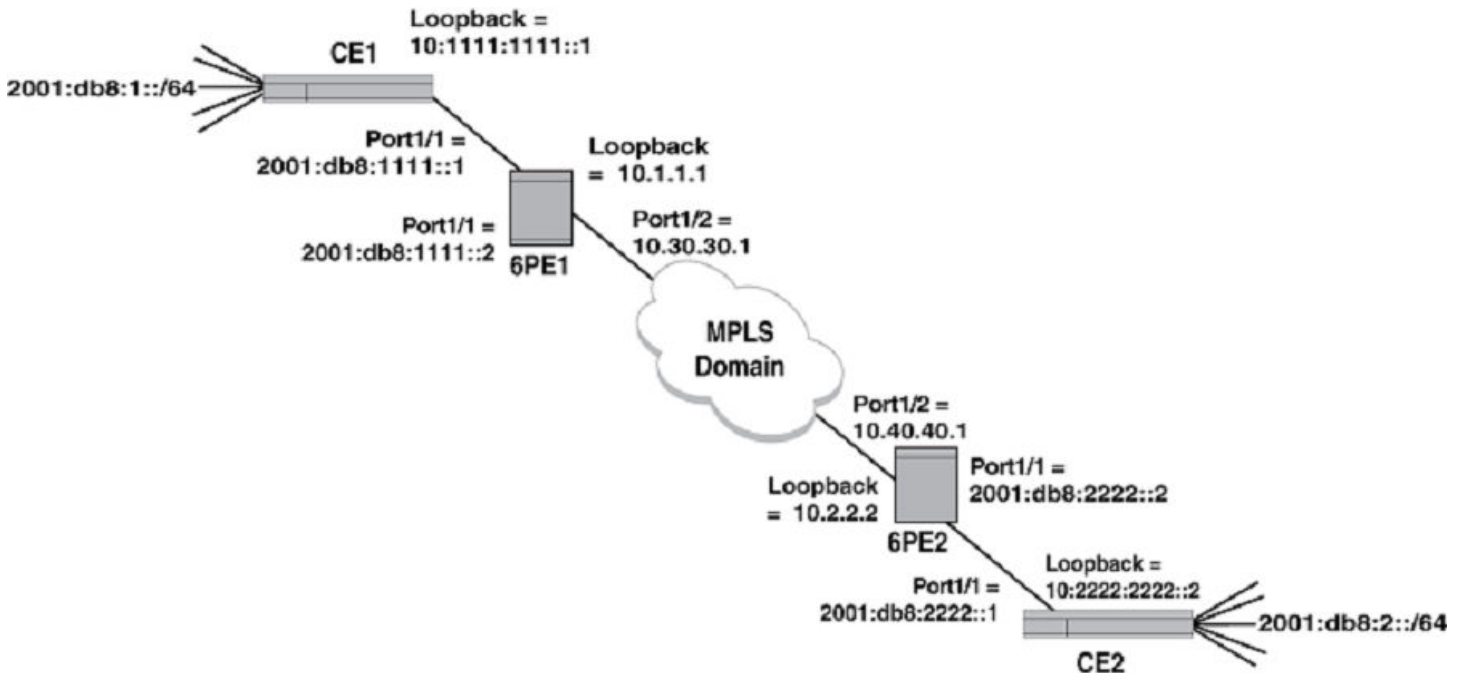
CE1 configuration

The following example describes the operational requirements for the CE1 router. External BGP is configured between the CE1 and 6PE1 routers, and the connected route is redistributed through this connection.

```
CE1(config)# interface loopback 1
CE1(config-lbif-1)# ipv6 address 10:1111:1111::1/128
CE1(config-lbif-1)# exit
CE1(config)# interface ethernet 1/1
CE1(config-if-e1000-1/1)# ipv6 address 3ffe:db8:1111::1/64
CE1(config-if-e1000-1/1)# exit
CE1(config)# router bgp
CE1(config-bgp)# local-as 2
CE1(config-bgp)# address-family ipv6 unicast
CE1(config-bgp-ipv6u)# neighbor 3ffe:db8:1111::2 remote-as 1
CE1(config-bgp-ipv6u)# redistribute connected
CE1(config-bgp-ipv6u)# exit
```

6PE1 configuration

The following example describes the configuration for the 6PE1 router. The 6PE and the MPLS tunnel are configured between the 6PE1 and 6PE2 routers.

```
6PE1(config)# interface ethernet 1/1
6PE1(config-if-e1000-1/1)# ipv6 address 3ffe:db8:1111::2/64
6PE1(config-if-e1000-1/1)# exit
6PE1(config)# interface ethernet 1/2
6PE1(config-if-e10000-1/2)# ip address 10.30.30.1/24
6PE1(config-if-e10000-1/2)# exit
6PE1(config)# router bgp
6PE1(config-bgp)# local-as 1
6PE1(config-bgp)# neighbor 10.40.40.1 remote-as 1
6PE1(config-bgp)# address-family ipv6 unicast
```

```
6PE1(config-bgp-ipv6u)# neighbor 3ffe:db8:1111::1 remote-as 2
6PE1(config-bgp-ipv6u)# neighbor 10.40.40.1 activate
6PE1(config-bgp-ipv6u)# neighbor 10.40.40.1 send-label
6PE1(config-bgp-ipv6u)# exit
6PE1(config)# router mpls
6PE1(config-mpls)# mpls-interface ethernet 1/2
6PE1(config-mpls)# lsp tunnel1
6PE1(config-mpls-lsp-tunnel1)# to 10.40.40.1
6PE1(config-mpls-lsp-tunnel1)# from 10.30.30.1
6PE1(config-mpls-lsp-tunnel1)# enable
6PE1(config-mpls-lsp-tunnel1)# exit
```

6PE2 configuration

The following example describes the configuration for the 6PE2 router. The 6PE and the MPLS tunnel are configured between the 6PE1 and 6PE2 routers.

```
6PE2(config)# interface ethernet 1/1
6PE2(config-if-e1000-1/1)# ipv6 address 3ffe:db8:2222::2/64
6PE2(config-if-e1000-1/1)# exit
6PE2(config)#interface ethernet 1/2
6PE2(config-if-e1000-1/2)# ip address 10.40.40.1/24
6PE2(config-if-e1000-1/2)# exit
6PE2(config)# router bgp
6PE2(config-bgp)# local-as 1
6PE2(config-bgp)# neighbor 10.30.30.1 remote-as 1
6PE2(config-bgp)# address-family ipv6 unicast
6PE2(config-bgp-ipv6u)# neighbor 3ffe:db8:2222::1 remote-as 3
6PE2(config-bgp-ipv6u)# neighbor 10.30.30.1 activate
6PE2(config-bgp-ipv6u)# neighbor 10.30.30.1 send-label
6PE2(config-bgp-ipv6u)# exit
6PE2(config)# router mpls
6PE2(config-mpls)# mpls-interface ethernet 1/2
6PE2(config-mpls)# lsp tunnel1
6PE2(config-mpls-lsp-tunnel1)# to 10.30.30.1
6PE2(config-mpls-lsp-tunnel1)# from 10.40.40.1
6PE2(config-mpls-lsp-tunnel1)# enable
6PE2(config-mpls-lsp-tunnel1)# exit
```

CE2 configuration

The following example describes the operational requirements for the CE2 router. External BGP is configured between the CE2 and 6PE2 routers, and the connected route is redistributed through this connection.

```
CE2(config)# interface loopback 1
CE2(config-lbif-1)# ipv6 address 10:2222:2222::2/128
CE2(config-lbif-1)# exit
CE2(config)# interface ethernet 1/1
CE2(config-if-e1000-1/1)# ipv6 address 3ffe:db8:2222::1/64
CE2(config-if-e1000-1/1)# exit
CE2(config)# router bgp
CE2(config)# local-as 3
CE2(config-bgp)# address-family ipv6 unicast
CE2(config-bgp-ipv6u)# neighbor 3ffe:db8:2222::2 remote-as 1
CE2(config-bgp-ipv6u)# redistribute connected
CE2(config-bgp-ipv6u)# exit
```

Displaying IPv6 over MPLS information

Various show commands can be used to view information about IPv6 over MPLS.

IPv6 over MPLS is also known as 6PE technology, referring to the MPLS provider edge routers which forward the IPv6 traffic.

You can display the following information about the IPv6 over MPLS configuration on the device:

- IPv6 over MPLS neighbors information
- Error information for IPv6 over MPLS neighbors
- Route summary information for IPv6 over MPLS neighbors
- IPv6 over MPLS summary information
- IPv6 over MPLS packet count information

The **show ip bgp 6pe neighbors** command displays the IPv6 over MPLS neighbor information.

```
device# show ip bgp 6pe neighbors 10.30.30.1

Total number of BGP Neighbors: 1
1 IP Address: 10.30.30.1, AS: 1 (IBGP), RouterID: 10.30.30.1, VRF: default-vrf
State: ESTABLISHED, Time: 0h34m48s, KeepAliveTime: 60, HoldTime: 180
KeepAliveTimer Expire in 47 seconds, HoldTimer Expire in 124 seconds
Minimal Route Advertisement Interval: 0 seconds
RefreshCapability: Received
Messages: Open Update KeepAlive Notification Refresh-Req
Sent : 6 6 58 2 0
Received: 6 6 56 3 0
Last Update Time: NLRI Withdraw NLRI Withdraw
Tx: 0h34m48s --- Rx: 0h34m48s ---
Last Connection Reset Reason:User Reset Peer Session
Notification Sent: Cease/Administrative Reset
Notification Received: Cease/Administrative Reset
Neighbor NLRI Negotiation:
Peer Negotiated IPV4 unicast capability
Peer Negotiated IPV6 unicast capability
Peer configured for IPV4 unicast Routes
Peer configured for IPV6 unicast Routes
Neighbor ipv6 MPLS Label Capability Negotiation:
Peer Negotiated ipv6 MPLS Label capability
Peer configured for ipv6 MPLS Label capability
Neighbor AS4 Capability Negotiation:
Outbound Policy Group:
ID: 2, Use Count: 1
BFD:Disabled
TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
Maximum segment size: 1460
TTL check: 0, value: 0, rcvd: 62
Byte Sent: 981, Received: 962
Local host: 40.40.40.1, Local Port: 179
Remote host: 30.30.30.1, Remote Port: 8167
ISentSeq: 3437281442 SendNext: 3437282424 TotUnAck: 0
TotSent: 982 ReTrans: 0 UnAckSeq: 3437282424
IRcvSeq: 3443025033 RcvNext: 3443025996 SendWnd: 64981
TotalRcv: 963 DupliRcv: 0 RcvWnd: 65000
SendQue: 0 RcvQue: 0 CngstWnd: 3102
```

The following example from the **show ip bgp 6pe neighbors last-packet-with-error** command displays error information for the 6PE neighbors.

```
device# show ip bgp 6pe neighbors last-packet-with-error

Total number of BGP Neighbors: 1
No received packet with error logged for any neighbor
```

The following example from the **show ip bgp 6pe neighbors routes-summary** command displays the routes summary for the 6PE device neighbors.

```
device# show ip bgp 6pe neighbors routes-summary

  Total number of BGP Neighbors: 1
1 IP Address: 30.30.30.1
Routes Accepted/Installed:2, Filtered/Kept:0, Filtered:0
```

```
    Routes Selected as BEST Routes:2
      BEST Routes not Installed in IP Forwarding Table:0
    Unreachable Routes (no IGP Route for NEXTHOP):0
    History Routes:0
  NLRIs Received in Update Message:2, Withdraws:0 (0), Replacements:0
    NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Invalid Confed aspath:0, maxas-limit aspath:0
      Duplicated Originator_ID:0, Cluster_ID:0
  Routes Advertised:2, To be Sent:0, To be Withdrawn:0
  NLRIs Sent in Update Message:2, Withdraws:0, Replacements:0
  Peer Out of Memory Count for:
      Receiving Update Messages:0, Accepting Routes(NLRI):0
      Attributes:0, Outbound Routes(RIB-out):0 Outbound Routes Holder:0
```

The following example from the **show ip bgp 6pe summary** command displays summary information for IPv6 over MPLS on a 6PE device.

```
device> show ip bgp 6pe summary

BGP4 Summary
  Router ID: 10.40.40.1 Local AS Number: 1
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 1, UP: 1
  Number of Routes Installed: 4, Uses 344 bytes
  Number of Routes Advertising to All Neighbors: 2 (2 entries), Uses 96 bytes
  Number of Attribute Entries Installed: 4, Uses 360 bytes
  Neighbor Address  AS#   State  Time     Rt:Accepted Filtered Sent ToSend
  10.30.30.1        1     ESTAB  0h35m18s 2           0        2    0
```

The following example from the **show mpls statistics 6pe** command displays the count of IPv6 over MPLS packets per packet processor on a 6PE device.

```
device> show mpls statistics 6pe

In-Port(s)      Endpt Out-Pkt     Tnl Out-Pkt
e1/1 - e1/2     184116072         1697803327
e1/3 - e1/4     389547885         6036111
e2/1 - e2/24    1088610           0
e2/25 - e2/48   0                 248406
e3/1            2045067126        5288598554
e3/2            0                 0
```

Clearing IPv6 over MPLS neighbor information

IPv6 over MPLS uses BGP neighbors to carry label information and the session counters can be cleared using a CLI command.

IPv6 over MPLS is configured on IPv6 provider edge (6PE) routers and some configuration scenarios require that the BGP neighbor information is cleared. To determine the effect of clearing the IPv6 over MPLs neighbor information, an appropriate **show** command is entered before and after the **clear** command.

1. Enter the **end** or **exit** command to return to privileged EXEC mode.

2. Enter the **show ip bgp 6pe neighbors** command.

    ```
    device# show ip bgp 6pe neighbors


    Statistics:
        Advertisements: Rx: 0, Tx: 60
        Neighbor Advertisements: Tx: 30
    ```

3. To clear all BGP neighbor information, enter the **clear ip bgp 6pe neighbors all** command.

```
device# clear ip bgp 6pe neighbors all
```

4. Enter the **show ip bgp 6pe neighbors** command.

```
device# show ip bgp 6pe neighbors


Statistics:
    Advertisements: Rx: 0, Tx: 60
    Neighbor Advertisements: Tx: 30
```

In this show output after the **clear ip bgp 6pe neighbors all** command has been entered, you can see that the statistical counters have been reset. Although some of the counters are showing numbers because BGP traffic is still flowing, the numbers are much lower (6 transmissions instead of 60 transmissions) than in the initial **show ip bgp 6pe neighbors** command output.

IPv6 over MPLS behavioral differences on Extreme devices

The IPv6 over MPLS support provided on the MLX Series and XMR Series devices differs from that provided on the CES 2000 Series and CER 2000 Series devices.

Command output statistics

For the output of the **show mpls statistics 6pe** command, the MLX Series and XMR Series devices can count both the Endpt Out-pkt and Tnl Out-Pkt statistics, whereas the CES 2000 Series and CER 2000 Seriesdevices can count only the Endpt Out-pkt statistics due to hardware limitations.

TTL propagation for IPv6 over MPLS packets

The Time-to-Live (TTL) propagation for IPv6 over MPLS packets is controlled by the **propagate-ttl** and **label-propagate-ttl** commands on the routers. By default, the **propagate-ttl** command is enabled and the **label-propagate-ttl** command is disabled for the MLX Series and XMR Series devices. There are differences in functionality for the CES 2000 Series and CER 2000 Series devices.

TABLE 45 Differences in the IPv6 over MPLS TTL propagation behavior on the Extreme devices

Behavior of MLX Series and XMR Series devices	Behavior of CES 2000 Series and CER 2000 Series devices
The TTL propagation for IPv6 over MPLS packets is controlled by the **propagate-ttl** command on the ingress and egress IPv6 provider edge (6PE) routers, and by the **label-propagate-ttl** command on the transit and the Penultimate Hop Popping (PHP) routers.	The TTL propagation for 6PE packets is controlled only by the **propagate-ttl** command on the ingress 6PE, egress 6PE, transit, and PHP routers. The **label-propagate-ttl** command is not applicable for the CES 2000 Series and CER 2000 Series devices. Due to design limitations, when the **propagate-ttl** command is enabled on the PHP router, the TTL value is not decremented and therefore the recorded TTL value at the endpoint is one more than the expected value.

CoS behavior for IPv6 over MPLS packets

The table describes the Class of Service (CoS) treatment for IPv6 over MPLS packets at the ingress 6PE, egress 6PE, transit, and PHP routers, and the differences in the CoS behavior between the MLX Series and XMR Series devices, and the CES 2000 Series and CER 2000 Series devices.

TABLE 46 Differences in the IPv6 over MPLS TTL propagation CoS behavior on the Extreme devices

Behavior of MLX Series and XMR Series devices	Behavior of CES 2000 Series and CER 2000 Series devices
At the ingress router, the tunnel and 6PE label EXP bits are set to internal priority. If the tunnel has a configured CoS value, the configured value will override the internal priority.	At the ingress router, the tunnel and 6PE label EXP bits are set to internal priority. If the tunnel has a configured CoS value, the configured value will override the internal priority.
At the transit router, if the port encode EXP bit is set, then both the MPLS tunnel EXP and 6PE label EXP bits will be overwritten from the encoding table.	At the transit router, if the port encode EXP bit is set, then only the MPLS tunnel EXP bit will be overwritten from the encoding table.
At the PHP router, the tunnel label is popped and the 6PE label keeps its EXP value unchanged.	At the PHP router, the tunnel label is popped and the 6PE label keeps its EXP value unchanged.
At the egress router, the 6PE label is popped and the IPv6 packet keeps its DSCP value unchanged. If the port encode DSCP bit is set, then the IPv6 DSCP value will be overwritten from the encoding table.	At the egress router, the 6PE label is popped and the DSCP value in the IPv6 header is updated based on the DSCP encode policy. For these devices, the DSCP encode policy cannot be switched off.

IPv6 over MPLS VPN overview

IPv6 over MPLS VPN, also known as IPv6 VPN Provider Edge Router (6VPE), enables connecting two private IPv6 networks over an IPv4-based MPLS public network.

The implementation of IPv6 over MPLS VPN follows RFC 4798. The 6VPE router can be an existing Provider Edge (PE) router or a new PE router dedicated to IPv6 traffic. The Provider Edge (PE) router supports dual stack IPv4 and IPv6. Similar to L3VPN, the 6VPE router utilizes the existing MPLS backbone to forward IPv6 traffic from PE to PE using the MPLS labels.

The following figure is the network topology for 6VPE forwarding.

FIGURE 75 IPv6 over MPLS VPN topology
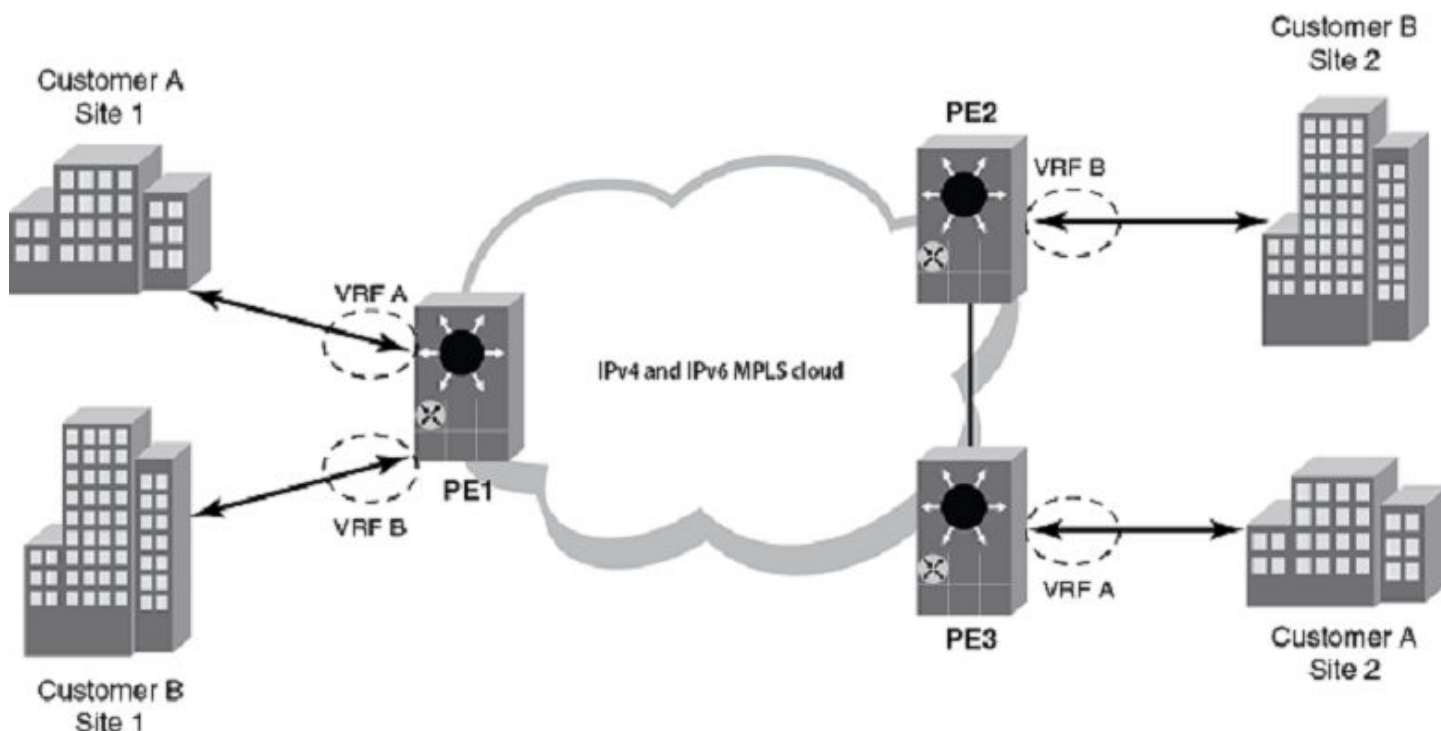
The IPv6 over MPLS VPN network consists of the following key elements in the topology:

- Customer A sites—Advertise IPv6 routes
- Customer B sites—Advertise IPv6 routes
- PE1—The source or the ingress label switch router (LSR)
- PE2 and PE3—The destination or the egress LSRs

The topology shows different customer sites that are connected through the IPv4 MPLS network and operating with IPv6 address families. The PE routers run dual stack IPv4 and IPv6. Customer A's site 1 and site 2 are connected through the MPLS cloud and a different VRF is maintained at the PE for each customer. Customer A's site 1 is connected to PE1 with VRF A and customer B's site 1 is connected to PE1 with VRF B.

> **NOTE**
> IPv6 over MPLS VPN supports only IPv6 unicast forwarding and does not support IPv6 multicast forwarding.

The table provides information about the different routing protocols that are used for advertising routes between the customer sites (CE) and the PE routers in the MPLS network.

TABLE 47 IPv6 over MPLS VPN route advertising

Connectivity	Protocols
PE to CE routing	BGP+, OSPFv3, or RIPng
PE to PE routing (exchanging IPv6 routes with VRF label)	MP-BGP
Exchanging of tunnel labels in MPLS cloud	LDP, RSVP

The following steps describe the IPv6 over MPLS VPN process in an existing MPLS domain:

1. Each customer site advertises IPv6 routes to attached PE routers. The PE routers receive the routes on their specific VRFs.

2. The PE routers assign VRF labels to all the received IPv6 packets.

3. The PE routers advertise the received routes to remote PE routers with IPv6 VRF label. The IPv6 VRF label is unique for each VRF within the PE.

4. The IPv6 packets are forwarded in the MPLS cloud based on the corresponding VRF routing table with IPv6 VRF labels.

5. The 6VPE router forwards the IPv6 packets from the customer router using the existing IPv4 LSPs.

Configuring IPv6 over MPLS VPN

To enable the forwarding of IPv6 traffic across an IPv4-based MPLS infrastructure you can configure IPv6 over MPLS VPN.

The IPv4 MPLS network must be configured.

An initial VRF is configured, an interface is assigned to the VRF, and an IPv6 address is configured on the interface. BGP routing is enabled and an IPv4 neighbor is configured under VPNv6 unicast address family. An MPLS interface is configured and the LSP is enabled. This configuration for IPv6 over MPLS VPN allows the 6VPE router to use the existing MPLS backbone to forward IPv6 traffic from one Provider Edge (PE) route to another PE using the MPLS labels.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a VRF instance named A.

```
device(config)# vrf A
```

3. Assign a Route Distinguisher (RD).

```
device(config-vrf-A)# rd 1:1
```

4. Configure the address-family.

```
device(config-vrf-A)# address-family ipv6
```

    IPv6 address-family is configured here.

5. Exit to global configuration mode.

```
device(config-vrf-A-ipv6)# exit
```

    The next step of configuring an interface starts in global configuration mode.

6. Configure an interface.

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

7. Assign an interface to the VRF.

```
device(config-if-e10000-1/1)# vrf forwarding A
```

8. Configure an IPv6 address.

```
device(config-if-e10000-1/1)# ipv6 address 3ffe:db8::2/64
```

9. Exit to global configuration mode.

    ```
    device(config-if-e10000-1/1)# exit
    ```

    The next step of configuring a BGP instance starts in global configuration mode.

10. Configure BGP routing.

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

11. Enter VPNv6 address family mode.

    ```
    device(config-bgp)# address-family vpnv6 unicast
    ```

12. Configure an IPv4 peer under VPNv6 address family.

    ```
    device(config-bgp-vpnv6u)# neighbor 10.0.0.1 remote-as 100
    ```

13. Configure outbound route filtering send capability to neighbor 10.0.0.1.

    ```
    device(config-bgp-vpnv6u)# neighbor 10.0.0.1 capability orf prefixlist send
    ```

14. Exit to global configuration mode.

    ```
    device(config-bgp-vpnv6u)# exit
    ```

    The next step of configuring MPLS and the LSP starts in global configuration mode.

15. Configure MPLS.

    ```
    device(config)# router mpls
    ```

16. Configure the MPLS interface.

    ```
    device(config-mpls)# mpls-interface ethernet 1/5
    ```

17. Configure the LSP.

    ```
    device(config-mpls)# lsp 6vpetunn
    ```

18. Configure the LSP.

    ```
    device(config-mpls-lsp-6vpetunn)# to 10.0.0.1
    ```

19. Enable the LSP.

    ```
    device(config-mpls-lsp-6vpetunn)# enable
    ```

In the following example, all the components to enable IPv6 over MPLS VPN are configured.

```
device# configure terminal
device(config)# vrf A
device(config-vrf-A)# rd 1:1
device(config-vrf-A)# address-family ipv6
device(config-vrf-A-ipv6)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding A
device(config-if-e10000-1/1)# ipv6 address 3ffe:db8::2/64
device(config-if-e10000-1/1)# exit
device(config)# router bgp
device(config-bgp)# address-family vpnv6 unicast
device(config-bgp-vpnv6u)# neighbor 10.0.0.1 remote-as 100
device(config-bgp-vpnv6u)# neighbor 10.0.0.1 capability orf prefixlist send
device(config-bgp-vpnv6u)# exit
device(config)# router mpls
device(config-mpls)# mpls-interface ethernet 1/5
device(config-mpls)# lsp 6vpetunn
device(config-mpls-lsp-6vpetunn)# to 10.0.0.1
device(config-mpls-lsp-6vpetunn)# enable
```

IPv6 over MPLS VPN use case scenarios

IPv6 over MPLS VPN can be implemented in various use cases. Some example network diagrams demonstrate the use cases.

IPv6 over MPLS VPN, also known as IPv6 VPN Provider Edge Router (6VPE), enables connecting two private IPv6 networks over an IPv4-based MPLS public network.

Use Case Scenario: 6VPE forwarding with P router

The figure shows shows 6VPE packet flow when PE routers are connected with intermediate P routers in an MPLS network.

FIGURE 76 6VPE forwarding with P router
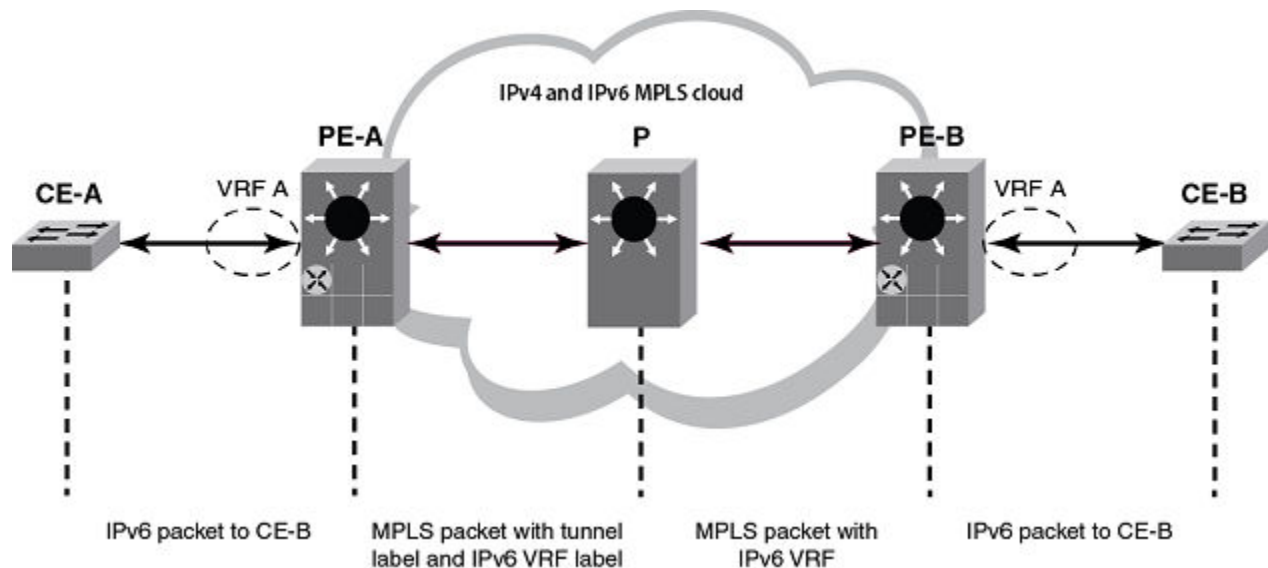
Use Case Scenario: 6VPE forwarding when PEs are directly connected

The figure shows 6VPE packet flow when PE routers are directly connected to each other in an MPLS network.

FIGURE 77 6VPE forwarding when PEs are directly connected
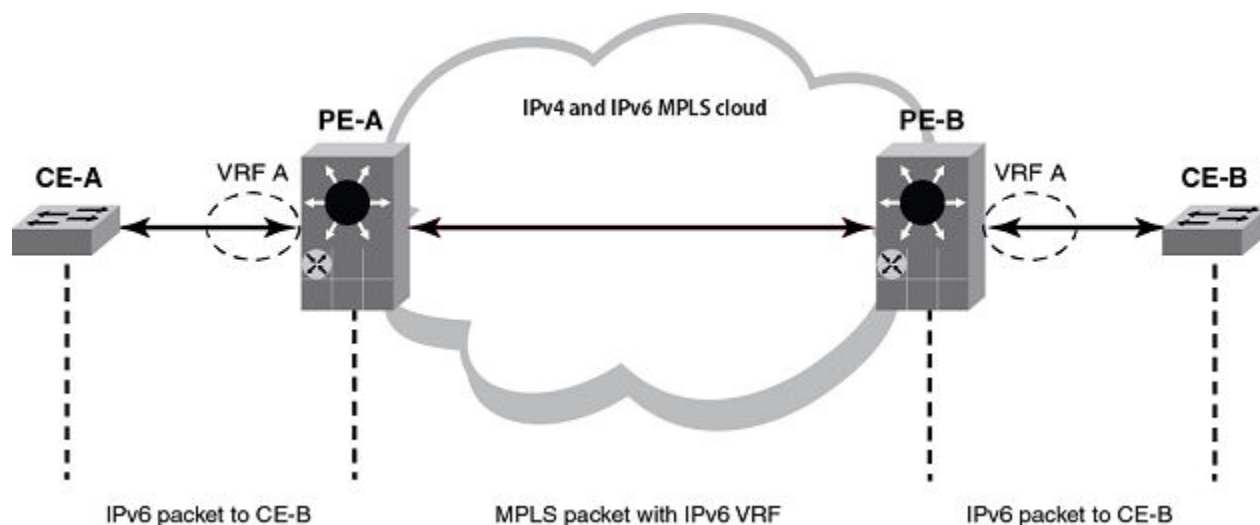
Deploying IPv6 over MPLS VPN configuration example

To deploy IPv6 over MPLS VPN all the devices used to forward IPv6 and MPLS traffic must be configured with IP and IPv6 addresses, routing protocols, VRF configuration, and MPLS must be enabled.

The figure displays the network topology used in this configuration example.

FIGURE 78 IPv6 over MPLs VPN example topology
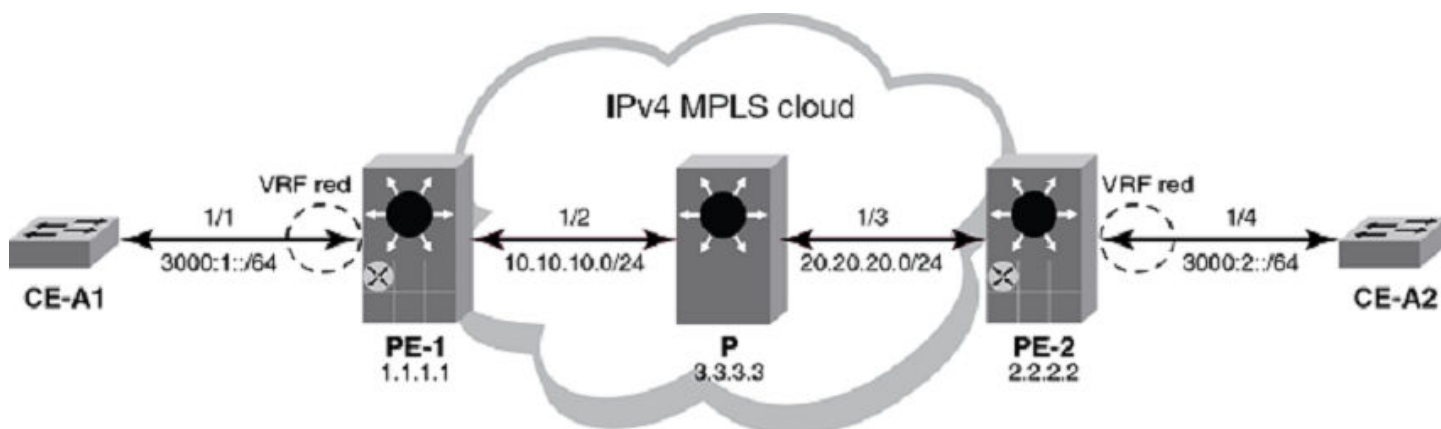
CE-A1 configuration

The following example shows both OSPFv3 and interface configuration for the customer edge device CE-A1 in the figure.

```
CE-A1(config)# ip router-id 10.110.110.1
CE-A1(config)# ipv6 router ospf
CE-A1(config-ospf6-router)# area 1
CE-A1(config-ospf6-router)# exit
CE-A1(config)# interface ethernet 1/1
CE-A1(config-if-e10000-1/1)# enable
CE-A1(config-if-e10000-1/1)# ipv6 address 3000:1::1/64
```

```
CE-A1(config-if-e10000-1/1)# ipv6 ospf area 1
CE-A1(config-if-e10000-1/1)# exit
```

PE1 configuration

The following example shows the VRF, OSPFv2, OSPFv3, BGP, Loopback interface, IPv4 and IPv6 addressing, and MPLS configuration for the provider edge device PE-1 in the figure.

```
PE-1(config)# vrf red
PE-1(config-vrf-red)# rd 1:1
PE-1(config-vrf-red)# ip router-id 10.120.120.2
PE-1(config-vrf-red)# address-family ipv6
PE-1(config-vrf-red-ipv6)# exit
PE-1(config)#
PE-1(config)# ipv6 router ospf vrf red
PE-1(config-ospf6-router-vrf-red)# area 1
PE-1(config-ospf6-router-vrf-red)# exit
PE-1(config)# interface ethernet 1/1
PE-1(config-if-e10000-1/1)# enable
PE-1(config-if-e10000-1/1)# vrf forwarding red
PE-1(config-if-e10000-1/1)# ipv6 address 3000:1::2/64
PE-1(config-if-e10000-1/1)# ipv6 ospf area 1
PE-1(config-if-e10000-1/1)# exit
PE-1(config)# router ospf
PE-1(config-ospf-router)# area 0
PE-1(config-ospf-router)# exit
PE-1(config)# interface loopback 1
PE-1(config-lbif-1)# ip address 1.1.1.1/32
PE-1(config-lbif-1)# ip ospf area 0
PE-1(config-lbif-1)# exit
PE-1(config)# interface ethernet 1/2
PE-1(config-if-e10000-1/2)# enable
PE-1(config-if-e10000-1/2)# ip address 10.10.10.1/24
PE-1(config-if-e10000-1/2)# ip ospf area 0
PE-1(config-if-e10000-1/2)# exit
PE-1(config)# router bgp
PE-1(config-bgp)# address-family vpnv6 unicast
PE-1(config-bgp-vpnv6u)# neighbor 20.0.0.1 remote-as 100
PE-1(config-bgp-vpnv6u)# neighbor 20.0.0.1 capability orf prefixlist send
PE-1(config-bgp-vpnv6u)# exit
PE-1(config)# router mpls
PE-1(config-mpls)# mpls-interface ethernet 1/2
PE-1(config-mpls)# lsp 6vpetunn
PE-1(config-mpls-lsp-6vpetunn)# to 2.2.2.2
PE-1(config-mpls-lsp-6vpetunn)# enable
PE-1(config-mpls-lsp-6vpetunn)# exit
```

P configuration

The following example shows OSPFv2, IPv4 addressing, Loopback interface, and MPLS configuration for the provider core device P in the figure.

```
P(config)# router ospf
P(config-ospf-router)# area 0
P(config-ospf-router)# exit
P(config)# interface ethernet 1/2
P(config-if-e10000-1/2)# enable
P(config-if-e10000-1/2)# ip address 10.10.10.1/24
P(config-if-e10000-1/2)# ip ospf area 0
P(config-if-e10000-1/2)# exit
P(config)# interface ethernet 1/3
P(config-if-e10000-1/3)# enable
P(config-if-e10000-1/3)# ip address 20.20.20.2/24
P(config-if-e10000-1/3)# ip ospf area 0
P(config-if-e10000-1/3)# exit
P(config)# interface loopback 1
P(config-lbif-1)# ip address 3.3.3.3/32
```

```
P(config-lbif-1)# ip ospf area 0
P(config-lbif-1)# exit
P(config)# router mpls
P(config-mpls)# mpls-interface ethernet 1/2
P(config-mpls)# mpls-interface ethernet 1/3
P(config-mpls)# exit
```

PE-2 configuration

The following example shows the VRF, OSPFv2, OSPFv3, Loopback interface, IPv4 and IPv6 addressing, and MPLS configuration for the provider edge device PE-2 in the figure.

```
PE-2(config)# vrf red
PE-2(config-vrf-red)# rd 1:1
PE-2(config-vrf-red)# ip router-id 130.130.130.3
PE-2(config-vrf-red)# address-family ipv6
PE-2(config-vrf-red-ipv6)# exit
PE-2(config)# ipv6 router ospf vrf red
PE-2(config-ospf6-router-vrf-red)# area 1
PE-2(config-ospf6-router-vrf-red)# exit
PE-2(config)# interface ethernet 1/4
PE-2(config-if-e10000-1/4)# enable
PE-2(config-if-e10000-1/4)# vrf forwarding red
PE-2(config-if-e10000-1/4)# ipv6 address 3000:2::2/64
PE-2(config-if-e10000-1/4)# ipv6 ospf area 1
PE-2(config-if-e10000-1/4)# exit
PE-2(config)# router ospf
PE-2(config-ospf-router)# area 0
PE-2(config-ospf-router)# exit
PE-2(config)# interface loopback 1
PE-2(config-lbif-1)# ip address 2.2.2.2/32
PE-2(config-lbif-1)# ip ospf area 0
PE-2(config-lbif-1)# exit
PE-2(config)# interface ethernet 1/3
PE-2(config-if-e10000-1/3)# enable
PE-2(config-if-e10000-1/3)# ip address 20.20.20.1/24
PE-2(config-if-e10000-1/3)# ip ospf area 0
PE-2(config-if-e10000-1/3)# exit
PE-1(config)# router bgp
PE-1(config-bgp)# address-family vpnv6 unicast
PE-1(config-bgp-vpnv6u)# neighbor 10.0.0.1 remote-as 100
PE-1(config-bgp-vpnv6u)# neighbor 10.0.0.1 capability orf prefixlist send
PE-1(config-bgp-vpnv6u)# exit
PE-2(config)# router mpls
PE-2(config-mpls)# mpls-interface ethernet 1/3
PE-2(config-mpls)# lsp 6vpetunn
PE-2(config-mpls-lsp-6vpetunn)# to 1.1.1.1
PE-2(config-mpls-lsp-6vpetunn)# enable
PE-2(config-mpls-lsp-6vpetunn)# exit
```

CE-A2 configuration

The following example shows both OSPFv3 and interface configuration for the customer edge device CE-A2 in the figure.

```
CE-A2(config)# ip router-id 10.140.140.4
CE-A2(config)# ipv6 router ospf
CE-A2(config-ospf6-router)# area 1
CE-A2(config-ospf6-router)# exit
CE-A2(config)# interface ethernet 1/4
CE-A2(config-if-e10000-1/4)# enable
CE-A2(config-if-e10000-1/4)# ipv6 address 3000:2::1/64
CE-A2(config-if-e10000-1/4)# ipv6 ospf area 1
CE-A2(config-if-e10000-1/4)# exit
```

Displaying IPv6 over MPLS VPN information

Various show commands can be used to view information about IPv6 over MPLS VPN.

IPv6 over MPLS VPN is also known as 6VPE technology, referring to the MPLS provider edge routers which forward the IPv6 traffic. The actual statistics come from a command associated with the IPv6 over MPLS, known as 6PE technology, and output fields have been added to display the Virtual Routing and Forwarding (VRF) information.

You can display the following information about the IPv6 over MPLS configuration on the device:

- IPv6 over MPLS VPN statistics for all VRFs configured in the system
- IPv6 over MPLS VPN statistics for a specific VRF

Displaying the IPv6 over MPLS VPN packet count

The following example from the **show mpls statistics 6pe vrf** command displays the count of 6VPE packets for all VRFs configured in the system.

```
Device# show mpls statistics 6pe vrf

VRF Name    In-Port(s)     Endpt Out-Pkt    Tnl Out-Pkt
Red         e1/1 - e1/2    184116072        1697803327
            e1/3 - e1/4    389547885        6036111
            e2/1 - e2/24   1088610          0
            e2/25 - e2/48  0                248406
Green       e1/1 - e1/2    116072           1697803
            e1/3 - e1/4    0                60111
            e2/1 - e2/24   1088610          0
            e2/25 - e2/48  0                248406
```

Displaying the IPv6 over MPLS VPN packet count for a specific VRF

The following example from the **show mpls statistics 6pe vrf** command with a specific *vrf-name* displays the count of 6VPE packets for a specific VRF.

```
Device# show mpls statistics 6pe vrf Red

VRF Name    In-Port(s)     Endpt Out-Pkt    Tnl Out-Pkt
Red         e1/1 - e1/2    184116072        1697803327
            e1/3 - e1/4    389547885        6036111
            e2/1 - e2/24   1088610          0
            e2/25 - e2/48  0                248406
```

Configuring BGP or MPLS VPNs

BGP or MPLS VPN Overview

This chapter describes how to configure BGP or MPLS VPNs on devices. BGP or MPLS VPNs as defined by RFC 2547 can be used by internet service providers to provide remote wide-area connectivity services using an MPLS domain for data traffic and IBGP to distribute routing information. Each customer network can be completely segregated from every other customer network while sharing the same infrastructure.

What is a BGP or MPLS VPN

MPLS provides scalable and efficient switching over an indeterminate group of devices along a predetermined Labeled Switch Path (LSP). Using MPLS, LSPs can be set statically or determined dynamically by the ISPs to provide traffic engineering features. BGP or MPLS VPNs build on this infrastructure to provide virtual-circuit connectionless service between remote sites. Using a common MPLS-domain, multiple Virtual Private Networks (VPNs) can be configured across a service-provider MPLS core network. Each VPN provides a secure data path that allows IP packetized traffic to share the infrastructure while being effectively segregated from other VPNs that are using the same MPLS domain.

In the diagram below, four separate customers (1–4) each have remote sites. Each customer is connected to a network at a remote site through the MPLS domain while being completely segregated and secure from traffic between other sites. For instance, CE 1 and CE 8 belong to Customer 1. CE 1 is connected to the BGP or MPLS VPN network through PE 1 and CE 8 through PE 4. Using the service provider's BGP or MPLS VPN service, traffic can be forwarded between CE1 and CE8 at the same time that Customers 2 through 4 use VPNs that operate over the same network infrastructure. Different customers can even use the same IP addresses without conflicting with other customers networks or creating any routing problems.

FIGURE 79 BGP or MPLS VPN network
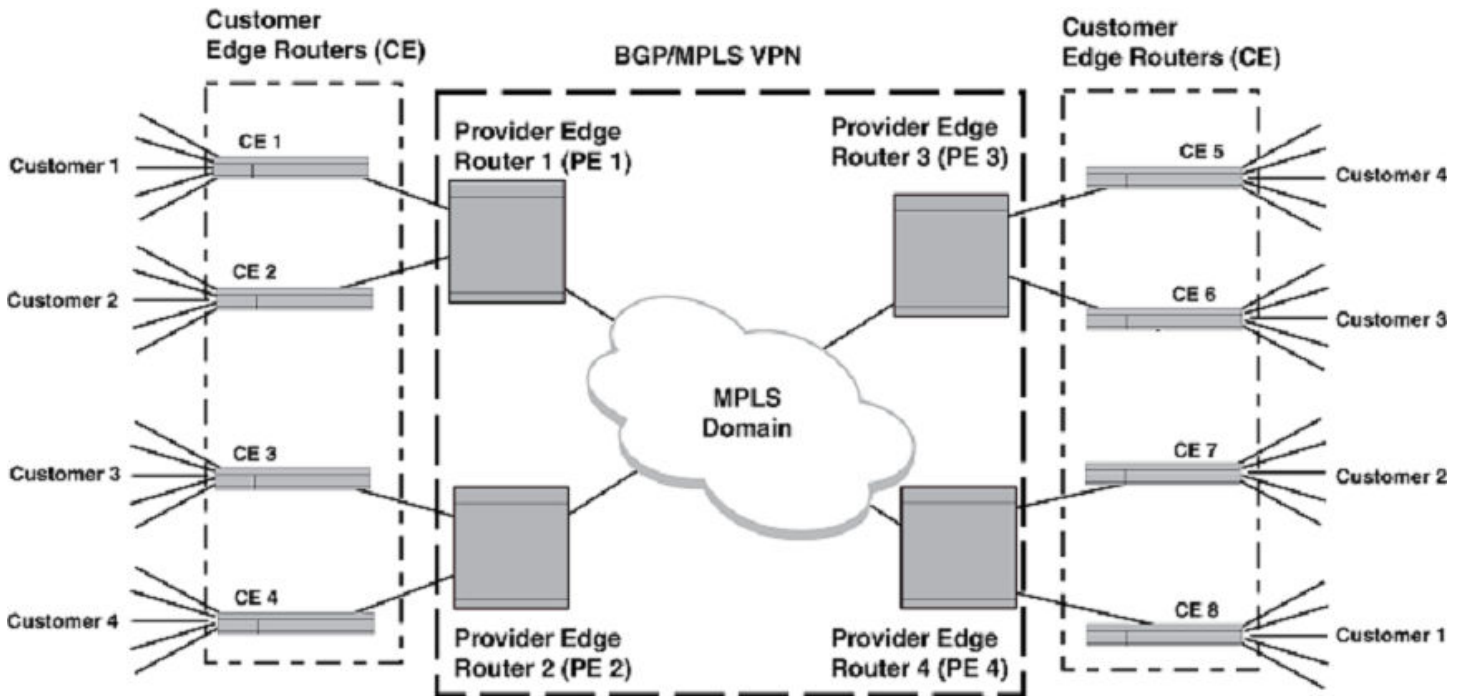
IETF RFC and Internet Draft support

The implementation of BGP or MPLS VPNs supports the following IETF RFCs and Internet Drafts:

BGP or MPLS VPNs

- RFC 4364: BGP or MPLS IP VPNs
- RFC 4577: OSPF as the PE or CE Protocol in BGP or MPLS IP VPNs
- RFC 4576: Using LSA Options Bit to Prevent Looping in BGP or MPLS IP VPNs (DN Bit)

BGP

- RFC 1771—A Border Gateway Protocol 4 (BGP-4)
- RFC 1997—BGP Communities Attribute
- RFC 2283—Multiprotocol Extensions for BGP-4
- RFC 2842—Capabilities Advertisement with BGP-4
- RFC 2858—Multiprotocol Extensions for BGP-4
- RFC 3107—Carrying Label Information in BGP-4

Draft standards

draft-ietf-idr-route-filter-11

draft-ietf-idr-bgp-ext-communities-07

MIB spport

RFC 4382 - MPLS or BGP Layer 3 Virtual Private Network (VPN) Management Information Base (with full support introduced in version 03.2.00 of the Multi-Service IronWare software).

BGP or MPLS VPN components and what they do

The following components, as shown in the diagram below, comprise a BGP or MPLS VPN.

- Customer Edge device (CE)—The CE provides connectivity with a customer's network and a Provider Edge device (PE). It can advertise routes available from the customer's network using RIP, OSPF or EBGP. Alternately, the CE can create a static default route to a PE. Outbound packets from a customer's network are forwarded from the CE to the PE, and inbound packets are forwarded from the PE to the CE attached to the customer's network.

- Provider Edge device (PE)—In a BGP or MPLS VPN, the central component is the PE. The PE provides connectivity with the CE and with the MPLS domain. On one side of the PE, routing information is exchanged with the CE using either static routes, RIP, OSPF, or EBGP. On the other side, IBGP is used with BGP multiprotocol extensions to communicate with all of the other PEs that are connected to networks in the same VPN and available to the customer's network. When a CE sends packets to a PE to forward across an MPLS domain, that PE functions as an MPLS ingress Label Edge router (LER) and the PE on the other end of the domain functions as an MPLS egress LER.

- Virtual Routing and Forwarding table (VRF)—Virtual Routing and Forwarding table (VRF) - The PE maintains a Virtual Routing and Forwarding table (VRF) for each customer that is attached to it through a CE. The VRF contains routes between the PE and the CE and *Label Switched Paths (LSPs)* across the MPLS domain for each PE that is a member of the customer's VPN. VRFs are defined on interfaces of the PEs.

- Provider MPLS domain—The Provider MPLS domain is composed of Provider (P) devices. An MPLS domain can traverse more than one service provider's MPLS network. The P devices do not store any VPN information; they just switch traffic from the ingress PE device along the LSP to the egress PE device.

FIGURE 80 BGP or MPLS VPN components
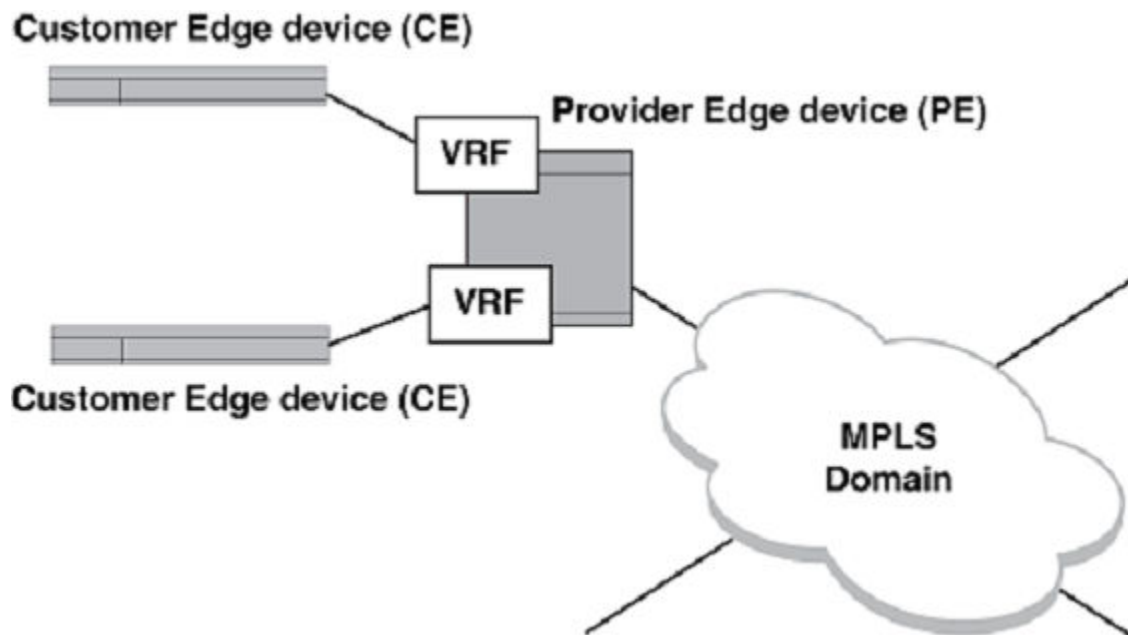
BGP or MPLS VPN operation

The purpose of a BGP or MPLS VPN is to forward packets between remote sites of a customer's network through a service provider's MPLS infrastructure. The section titled BGP or MPLS VPN components and what they do on page 435 describes the network components required to perform that task. The following sections describe how those components work together to create this service:

-
-

Creating routes in a BGP or MPLS VPN

The diagram below illustrates the various components involved in creating routes in a BGP or MPLS VPN.

A CE device maintains the connection to the customer's network and is configured within that network to share access to its available network prefixes and to receive packets from other VPN-connected networks. That CE is connected to a PE through an interface that is configured for a specified VRF for connection to the BGP or MPLS VPN. This connection places the CE in the BGP or MPLS VPN. Routes that are available through the CE are then made available to the PE using RIP, OSPF, EBGP or a static route. These routes are then stored in the VRF where they are associated with the VPN. The route from the CE to the PE is kept in the CEs routing table.

The PE device is connected to the MPLS domain through one or more interfaces. The PE must advertise the routes that it has available in it's VRF tables across the MPLS domain to its PE peers. Available routes in the VRF are prepended with a Route Distinguisher (RD) and advertised across the MPLS domain using IBGP. The PEs can either be configured for IBGP as either full mesh or with a route reflector to allow greater scalability. Routes that are advertised from other PEs in the VPN are received at the PE and collected in the VRF table. This procedure establishes which other PEs are in the VPN and what networks are available through them.

OSPF is used as the Interior Gateway Protocol (IGP) within the service provider's MPLS domain to provide connectivity. OSPF also populates the traffic engineering (TE) used by RSVP-TE.

Labeled Switch Paths (LSPs) are then created using Label Distribution Protocol (LDP), Resource Reservation Protocol (RSVP) configurations in the MPLS domain. Using this protocol, the PE obtains an LSP required to switch traffic to the other PEs. The network is now populated with all of the routes required to forward packets between the customer's networks.

FIGURE 81 BGP or MPLS VPN route discovery
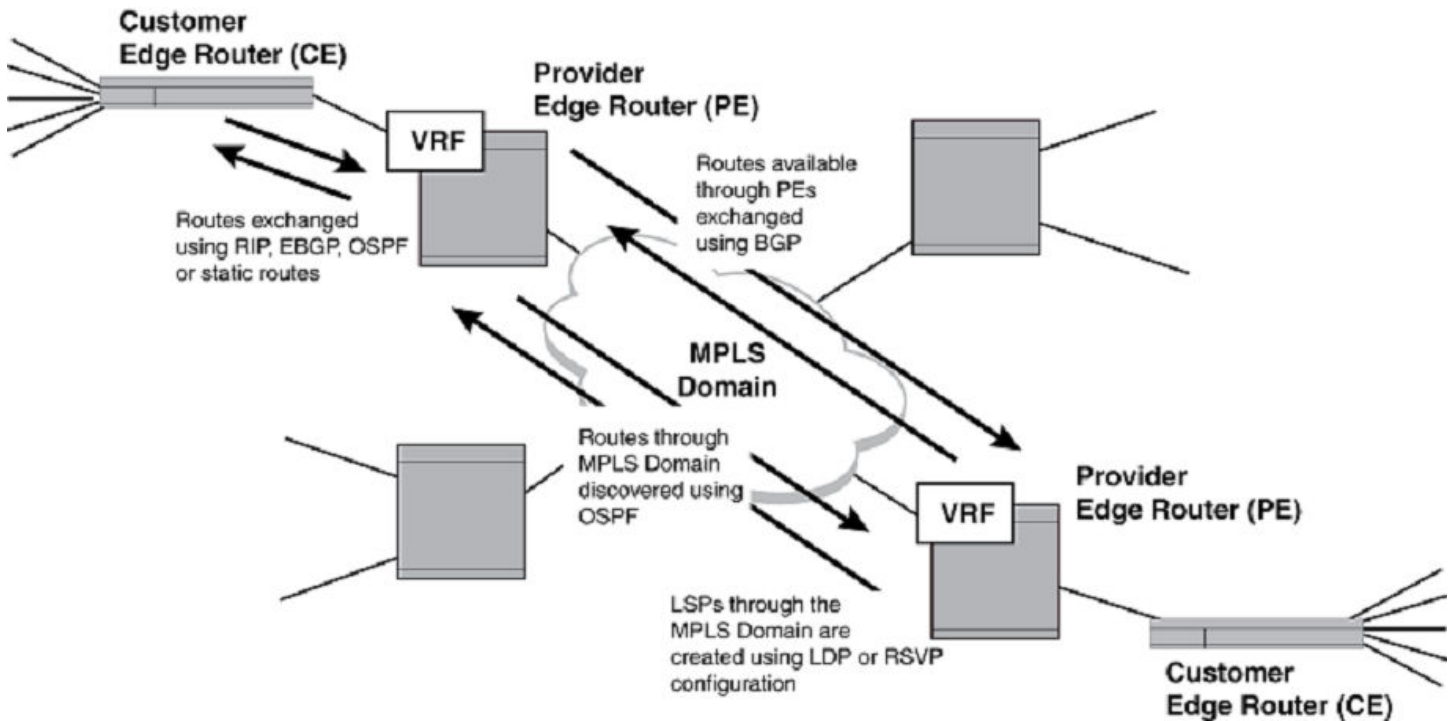
Routing a packet through a BGP or MPLS VPN

When a packet is forwarded from a CE to a PE, a bottom label is attached to the packet by the PE that is associated with the final destination. This label is obtained from the egress PE as part of the route discovery conducted by IBGP. Then, the top label which is obtained by the LSP connecting to the egress PE is added to the packet. The packet is then forwarded through the MPLS domain and is switched using the top label. At the penultimate device in the LSP, the top label is removed and the packet is forwarded to the egress PE. The egress PE uses the inner label to identify the CE to which the packet must be forwarded. The egress PE removes the inner label and forwards the packet to the correct CE.

The diagram below describes how a packet is forwarded through a BGP or MPLS VPN.

FIGURE 82 Routing a packet through a BGP or MPLS VPN
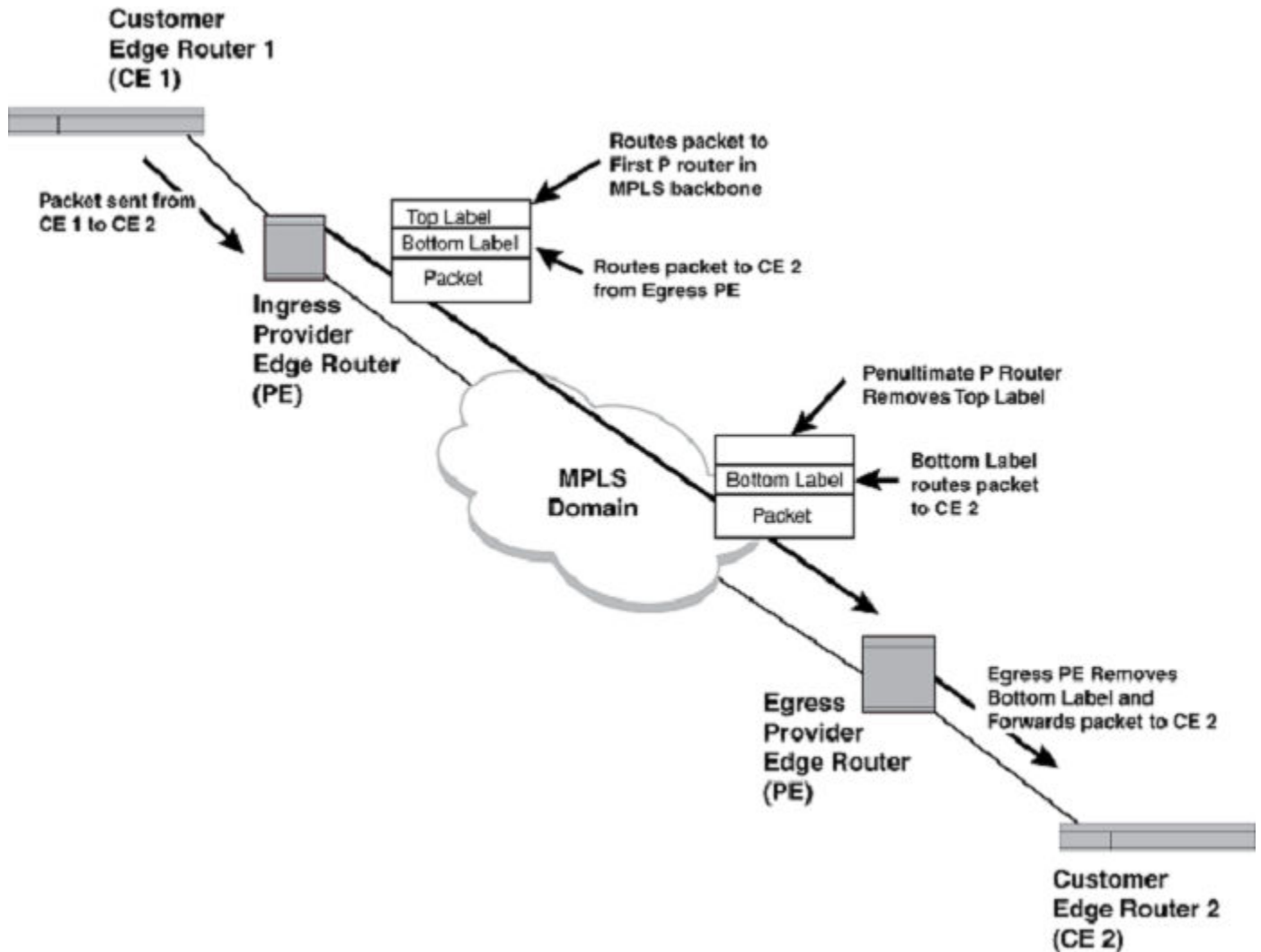
Configuring BGP or MPLS VPNs on a PE

Configuring BGP or MPLS VPNs on a PE involves the following configuration at a high level:

- Configuring VRF
- Associating the VRF to the interface
- Configuring BGP under VPNV4 unicast
- Configuring MPLS using LDP or RSVP

Perform the following steps to complete a basis BGP VPN configuration on a PE.

1. Enter the configuration mode.

   ```
   device (config)#
   ```

2. Configure route maps using the **route-map** command.

3. Define an extended community to use with a route map using the **ip extcommunity** command.

```
device(config)# ip extcommunity-list 20 deny rt 100:6
```

4. Complete the required VRF configuration steps for L3 VPN as following.

    a. Define a VRF routing instance.

    ```
    device(cionfig-vrf-VPN1)#
    ```

    b. Assign a route distinguisher to the VRF using the rd command.

    ```
    device(configute-vrf-VPN1)#rd 3:5
    ```

    c. Define an IPv4 or IPv6 address family for the VRF using the **address-family** command.

    ```
    device(config-vrf-VPN1)#address-family ipv4 max-route 130
    device(config-vrf-VPN1)#address-family ipv6 max-route 70
    ```

    d. Define an automatic route filtering using the **route-target** command.

    ```
    device(configure-vrf-VPN1)#route-target 3:5
    ```

    e. Apply the route maps to the VRF using the **import** command, to filter VPNv4 routes between the PEs in a BGP or MPLS VPN.

5. Complete the required interface configuration steps for L3 VPN as following.

    a. Configure a dynamic LAG using the **lag** command.

    ```
    device(config)#lag RED dynamic id 4
    ```

    b. Assign the VRF routing instance to the LAG interface from the ethernet interface configuration mode.

    ```
    device(config)# interface ethernet 1/1
    device(config-if-e10000-1/1)# vrf forwarding VPN1
    ```

    c. Enable VRF routing instance on virtual or physical interfaces on the PE using the vrf forwarding command.

    ```
    device(config-vif-1)#vrf forwarding VPN1
    ```

6. Complete the required BGP configuration steps for L3 VPN as follows.

    a. Switch to BGP configuration mode.

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

    b. Configure BGP VRF load sharing using the **ip load-sharing** and **maximum-paths** command, which works in relationship with each other.

    ```
    device(config-bgp)# maximum-paths 4
    ```

    c. Configure cooperative route filtering on the PE under the VPNV4u address family using the **neighbor capability** commands.

    ```
    device(config-bgp)#address-family vpnv4 unicast
    device(config-bgp-vpnv4u)#
    device(config-bgp-vpnv4u)# neighbor 10.3.3.1 capability orf extended-community send-vrf-filter
    device(config-bgp-vpnv4u)# neighbor 10.3.3.2 capability orf extended-community receive
    ```

    d. Create an extended community list for the routes that you want to filter, using the **rr-group** command.

    ```
    device(config-bgp-vpnv4u)# rr-group 1
    ```

e. Create VPNv4 route reflector using the **neighbor route-reflector-client** command.

```
device(config-bgp-vpnv4u)# neighbor 10.11.11.2 route-reflector-client
```

f. Configure autonomous system number override using the **neighbor as-override** command.

g. Execute the **neighbor allowas-in** command for a PE to allow routes with it's AS number.

```
device(config-bgp-ipv4u-vrf)# neighbor 10.33.36.2 allowas-in 3
```

7. Complete the required MPLS configuration steps for L3 VPN as follows.

a. Switch to VRF configuration mode.

b. Configure a Label Switched Path (LSP) per VRF using the **next-hop** loopback command to tunnel through the MPLS domain to the destination PE.

```
device(config-vrf-blue)# bgp next-hop loopback 2
```

c. Switch to configuration mode and define an OSPF instance in VRF using the **router ospf vrf** .

```
device(config)# router ospf vrf VPN1
device(config-ospf-router-VPN1)#
```

d. Create an OSPF area in the OSPF VRF instance using the **area** command.

```
device(config-ospf-router)# area 1
```

e. Assign a domain identifier to the OSPF VRF instance using the **domain-id** command.

```
device(config-ospf-router)# domain-id 10.0.0.100
```

f. Assign a domain tag to the OSPF VRF instance using he **domain-tag** command.

```
device(config-ospf-router)# domain-tag 1200
```

g. Configure a virtual intra-area OSPF link (sham link) between two PEs using the **area sham-link** command.

```
device(config-ospf-router-vrf-VPN1)# area 1 sham-link 10.2.2.1 10.2.2.2 cost 10
```

h. Switch to MPLS configuration mode.

```
device(config)#router mpls
device(config-mpls)#
```

i. Configure IP TTL to MPLS TTL propagation in an IPVPN from the MPLS policy configuration mode using the **vrf-propagate-ttl** and **label-propagate-ttl** commands.

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# vrf-propagate-ttl
device(config-mpls-policy)# label-propagate-ttl
```

Defining a VRF routing instance

A single PE can contain one or more VRFs. Each of these VRFs must be defined separately on a PE. A PE distributes routes and route packets to other members of the same VRF but not to other VRFs. The VRF name can be any string that the user wants to define it as.

To define the VRF routing instance VPN1 on a PE, enter the following command.

```
device(config)# vrf VPN1
device(config-vrf-vpn1)# exit-vrf
device(config)#
```

Syntax: [no] **vrf** *vrf_name*

Configures a VRF table on the device with the name *vrf_name* and puts the device in config-vrf mode.

The *vrf_name* parameter specifies a name for the VRF being created.

Syntax: [no] **exit-vrf**

The **exit-vrf** command moves the user out of the VRF configuration mode for the VRF the user is configuring.

Assigning a Route Distinguisher to a VRF

Each instance of a VRF must have a unique Route Distinguisher (RD) assigned to it. The RD is pre-pended on any address being routed or advertised. The RD can be defined as either ASN-relative or IP address-relative. Because the RD is unique to an instance of a VRF, it allows the same IP address to be used in different VPNs without creating any conflict.

To assign a Route Distinguisher (RD) for a VRF based on the AS number 3 and the arbitrary identification number 6, enter the following command.

```
device(config-vrf)# rd 3:6
```

For additional information on the **rd** command, see *NetIron Command Line Interface (CLI) Reference Guide*.

Defining IPv4 or IPv6 address families of a VRF

Each address family configuration level allows the user to access commands that apply to that particular address family only.

To define IPv4 or IPv6 address families of a VRF, enter the following command.

```
device(config)# vrf VPN1
device(config-vrf-vpn1)# address-family ipv4
device(config-vrf-vpn1-ipv4)# exit-address-family
device(config-vrf-vpn1)# exit-vrf
device(config)#
```

Syntax: [no] **address-family** *ipv4/ipv6*

Syntax: **exit-address-family**

The **exit-address-family** command moves the user out of the IPv4 or IPv6 address family of a VRF the user is configuring.

Defining automatic route filtering

Each VRF is configured with import and export route targets. The export route target sets an extended community attribute number that is appended to all routes that are exported from the VRF. The import route target value sets a filter that determines the routes that are accepted into the VRF. Any route with a value in its import route-target contained in its extended attributes field matching the value in the VRFs import route target is accepted. Otherwise, the route is rejected. This process is referred to as automatic route filtering.

To define an import route target of 3:6 and an export route target of 3:8 for a VPN, enter the following commands.

```
device(config-vrf)# route-target import 3:6
device(config-vrf)# route-target export 3:8
```

Syntax: [no] **route-target** [**import** | **export** | **both**] *route-target*

This command associates a route target specified by the route-target variable with a specified VRF for control on routes.

The **import** parameter specifies that routes with route-target extended community attributes matching the specified route-target variable can be imported into the VRF where this command is configured.

The **export** parameter specifies the route-target extended community attributes that are attached to routes export from the specified VRF.

The **both** parameter specifies that both the import and export values apply to the specified route-target variable for the VRF where this command is configured. This is the default state. It applies when no specific value for this parameter is set.

The *route-target* variable specifies a target VRF extended community. Like a route distinguisher, it is either AS-relative or IP address-relative.

Assigning a VRF routing instance to an interface

Once a VRF routing instance is defined, it must be assigned to one or more virtual or physical interfaces on a PE.

To assign the VRF named VPN1 to Ethernet interface 1/1, enter the following commands.

```
        device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
```

Syntax: [no] vrf forwarding *vrf-name*

The *vrf-name* variable is the name of the VPN that the interface is being assigned to.

Assigning a VRF routing instance to a LAG interface

A VRF routing instance can be assigned to a dynamic LAG interface. To assign a VRF routing instance to a LAG the following rules must be observed:

- The dynamic LAG must be configured before assigning any of its ports to a non-default VRF routing instance.
- Before deployment of the dynamic LAG all members of the LAG must be in the default VRF routing instance.
- After the LAG is deployed, the primary port can be assigned to a non-default VRF routing instance.
- Once the dynamic LAG is deployed, all ports are in the LACP_BLOCK state until the LACP protocol can negotiate with the other end. Once the negotiation with the other end is completed, all the LAP ports are set to the FORWARD state.
- When the Dynamic LAG is undeployed, the primary port stays in the VRF that it was assigned to but all secondary ports moves back to the default VRF.

The following configuration creates a dynamic LAG named "red" and assigns port 1/1 as the primary port and port 1/2 as a secondary port. The LAG is deployed and the primary port (1/1) is assigned to the VRF routing instance named "VPN1". All ports in the LAG named "red" are then assigned to the VRF routing instance named "VPN1".

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

When the dynamic LAG named "red" is undeployed as shown in the following, port 1/1 remains in the VRF routing instance named "VPN1" but port 1/2 is returned to the default VRF.

```
        device(config)# lag red dynamic
device(config-lag-red)# no deploy
```

Setting up cooperative route filtering

Automatic route filtering in VRFs is provided through the **route-target** import command. By placing this command in the VRF configuration, routes can be filtered from being imported into a given VRF. Routes with extended community route targets matching the VRF's import route-targets are permitted into a VRF. Otherwise, the routes are rejected.

The cooperative route filtering feature requires that the user sets a send command on the device that is sending the ORF, and a receive command on the device that is installing the ORF. To configure the sending device, use the following command in the VPNv4 address family.

```
        device(config-bgp-vpnv4u)# neighbor 10.3.3.1 capability orf extended-community send-vrf-filter
```

Syntax: [no] neighbor *neighbor_IPaddress* **capability orf extended-community send-vrf-filter**

To configure the peering device use the following command in the VPNv4 address family.

```
        device(config-bgp-vpnv4u)# neighbor 10.3.3.2 capability orf extended-community receive
```

Syntax: [no] neighbor *neighbor_IPaddress* **capability orf extended-community receive**

Importing and exporting route maps

Route-maps configured using the **route-map** command can be applied to a VRF to provide filtering of VPNv4 routes between PEs in a BGP or MPLS VPN. When a route-map is applied to a VRF, only VPNv4 routes are filtered. Other routes such as static routes, connected routes, OSPF VRF routes, or BGP CE side routes are not affected. Because the route map is applied to the VRF, it filters traffic to all connected PEs. This is in contrast to applying a route-map using the BGP neighbor. In that case, the route map applies to routes imported from or exported to the neighbor that is specified.

Route maps applied to a VRF can coexist with route maps that are applied to a BGP neighbor. The user can filter routes from being imported into a VRF using the import and export route commands. This allows the user to accept or deny the routes for one VRF without affecting the routes that are imported or exported from other VRFs. To do this, the user must define a route-map import or export command.

To configure a VRF to apply the import route map ImportOne, use the following command at the VPNv4 prompt.

```
        device(config)# vrf vrfone
device(config-vrf-vrfone)# import map ImportOne
device(config-vrf-vrfone)# exit-vrf
device(config)#
```

Syntax: [no] import map *map-name*

The *map-name* variable is the name of the route map that the user wants to apply to the VRF.

To configure a VRF to apply the export route map ExportOne, use the following command at the VPNv4 prompt.

```
        device(config)# vrf vrfone
device(config-vrf-vrfone)# export map ExportOne
device(config-vrf-vrfone)# exit-vrf
device(config)#
```

Syntax: [no] export map *map-name*

The *map-name* variable is the name of the route map that the user wants to apply to the VRF.

Defining an extended community for use with a route map

Routes can be filtered in or out of a PE by the use of an IP extended community to identify them. In this situation, a route is identified by its extended community variable. It is entered as a route target in an IP extended community list and then matched in a route-map command. This route map is then applied from the PE that is defining the route to be filtered to the PE where the route filter is to be implemented by using a **neighbor route-map** command. When a VRF exists on the neighbor that exports the route-target being blocked, all routes from that VRF are blocked from being sent to the PE where the filter is defined.

To define the IP extended community list 20 to define route target RT 100:6 to be denied, enter the following command.

```
device(config)# ip extcommunity-list 20 deny rt 100:6
```

Syntax: [no] ip extcommunity-list *num* [permit] [deny] [rt *routeID*] [soo *routeID*]

The *num* variable is the extended community list number.

The **permit** or **deny** parameters indicate the action that the device takes when the match is true.

The **rt** *route ID* variable specifies the route target that is applied to filtering. The *route ID* has the format of either ASN:nn or IP-address:nn. When four-byte ASNs have been enabled or when four-byte IP addresses are used, the user-purposed *nn* value can be a maximum of two bytes instead of our bytes.

The **soo** *route ID* variable specifies the site of origin. The *route ID* has the format of either ASN:nn or IP-address:nn. When four-byte ASNs have been enabled or when four-byte IP addresses are used, the user-purposed *nn* value can be a maximum of two bytes instead of our bytes.

Creating a VPNv4 route reflector

PE devices in a BGP or MPLS VPN share routes between each other using IBGP. This can be accomplished using a full mesh configuration or a route reflector can be used to simplify a networks topology and improve scalability. While the general concepts are the same for using Route Reflectors in a normal IBGP network as in an BGP or MPLS VPN, there are some differences. In addition, there are special conditions that apply when a route reflector is configured for normal IPv4 BGP traffic (IPv4) and for BGP or MPLS VPN traffic (VPNv4). The differences and special considerations are described in the following:

Special considerations when configuring a route reflector for both IPv4 and VPNv4:

- A VPNv4 route does not need to be installed in any VRF before being reflected.
- Route reflector configurations for IPv4 and VPNv4 are separated in different address family configurations.
- For a VPNv4 route installed to a VRF, the reflected VPNv4 route still carries the original RD and PA.
- When there is a route reflector configuration change, a warning message is displayed that requests the user to clear the neighbor session.

Specific commands for VPNv4 - There are VPNv4 specific commands that must be configured to configure a route reflector for a BGP or MPLS VPN under address family VPNv4. A route reflector can be configured on a PE for IPv4 and VPNv4 or for either exclusively. When the user is configuring a route reflector for a BGP or MPLS VPN, the user must configure it specifically using the VPNv4 specific commands.

To create a VPNv4 route reflector with a client at the IP address 10.11.11.2, enter the following commands at the VPNv4 level of BGP Config level.

```
device(config-bgp-vpnv4u)# neighbor 10.11.11.2 route-reflector-client
```

Syntax: [no] neighbor *IPaddress* route-reflector-client

The *IPaddress* variable is the IP address of the PE device that the user wants to define the route reflector client.

A route reflector can be setup with local import filtering to filter out VPNv4 routes matched by an extended community list. This requires that the user creates an extended community list for the routes the user wants to filter and set the following command.

```
        device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# rr-group 1
```

Syntax: [no] **rr-group** *group-num*

The *group-num* variable refers to an extended community list number from 1 to 99 that specifies the routes that the user wants to filter.

Configuring BGP VRF load sharing

The default for each VRF is to maintain only the lowest-cost route in its routing table for each VPN that it is connected to. When a lower-cost route is discovered, it replaces the route that is currently in the table. When another route of equal cost is discovered, it is rejected. The Extreme device, however, is able to perform load sharing over multiple routes to the same destination. In order to make this feature operational, the user must increase the number of path entries allowed in a VRFs routing table.

Configuring BGP VRF load sharing requires two different CLI commands that work in relationship with each other. These are the global **ip load-sharing** command and the BGP VRF specific **maximum-path** command. The value set for ip load-sharing provides a maximum number that the maximum-path value for a specific route can be set to. The **maximum-path** command has a maximum value of 32.

> **NOTE**
> The maximum number of paths supported by the BR-MLX-10Gx24-DM module is 16.

When the IP load-sharing value is set to 4 or greater, the maximum-path value for a specific BGP VRF can be set to a value of from 1 to 4. The default is 1. When the IP load-sharing value is set to less than 4, the maximum-path value for a specific BGP VRF can only be set to the global IP load-sharing value or less.

To set the **maximum-path** value to 4, enter the following commands at the VPNv4 level of the BGP configuration level.

```
device(config-bgp)# maximum-paths 4
```

Syntax: [no] **maximum-paths** *number*

The *number* variable is the maximum number of routes that can be maintained for a VRF. The *number* value range is 2 to 32 and the default is 1.

> **NOTE**
> The *number* value cannot exceed the value set for the device by the **ip load-sharing** command.

ECMP forwarding for IP VPN

ECMP hardware forwarding is now supported for IP VPN packets when an outgoing interface is configured as a physical port and a VE interface. When multiple routes are using ECMP to route to a destination, then hardware ECMP is automatically enabled. ECMP load sharing for IP over MPLS is supported for 2 to 32 tunnels. For more information on configuring ECMP load sharing for IP VPN, refer to the Configuring BGP VRF load sharing task.

ECMP hardware forwarding is supported only in static CAM mode. ECMP hardware forwarding is not supported for dynamic mode. Hitless upgrade for ECMP hardware forwarding is not supported.

When configuring ECMP hardware forwarding, all outgoing paths must configured in the same VRF. A hash value is computed for a packet when it is received by XPP. XPP uses the hash value to select a PRAM that is used to forward the packet to its destination. An ECMP PRAM block consists of 8 PRAMs. The hash value for each outgoing packet on a Customer Edge device interface is calculated

based on the source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, and TCP or UDP destination port.

The hash value for each incoming packet on the route target is calculated based on the source MAC address, destination MAC address, VLAN ID, source IP address, destination IP address, IP protocol field, TCP or UDP source port, TCP or UDP destination port, and a VC label for an MPLS packet.

Configuring autonomous system number override

There are some situations where a customer wants to connect to a service provider's BGP or MPLS VPN network using the same AS number at more than one site. This can create a problem because it is the default BGP procedure to reject routes from the same AS. One solution to this problem is to configure a PE router to override the AS_PATH attribute of its BGP neighbor. This is accomplished by configuring the **neighbor as-override** command on the PE. When this is enabled, the PE device determines when the AS_PATH attribute in a route intended for a neighbor CE contains the same AS number as the CE. When this is determined, the PE device substitutes its own AS number for the CEs in the AS_PATH attribute. The CE is then able to receive the route. The following additional conditions apply when this feature is in effect:

- In a situation where the AS_PATH attribute contains more than one occurrence of the CEs AS number in the initial sequence, the PE device replaces all those occurrences with its own AS number.
- The PE device adds its own AS number to the AS_PATH attribute just as it would normally.

The following command configures the PE device to replace its attached CEs AS number with its own AS number. BGP neighbor at IP address 10.33.36.2 the configuration of PE 2 required to enable Autonomous System number override for the BGP neighbor CE 2.

To configure a PE device to replace its attached CEs AS number with its own AS number, enter the following commands at the VRF level of the BGP Config level.

```
device(config-bgp-vpnv4u)# neighbor 10.33.36.2 as-override
```

Syntax: [no] neighbor *ip-address* **as-override**

The *ip-address* variable is the IP address of the CE whose AS number is being replaced with the PEs AS number.

Configuring a PE to allow routes with its AS number

BGP rejects routes that contain its own AS number within its AS_PATH attribute to prevent routing loops. In an MPLS or VPN hub and spoke topology this can stop legitimate routes from being accepted. The **allowas-in** command fixes this problem by allowing the user to set a parameter that disables the AS_PATH check function for routes learned from a specified location.

To configure a PE to disable the AS_PATH check function for routes sent to it by its BGP neighbor (a CE device with the IP address 10.33.36.2) for a maximum limit of three occurrences of the route, enter the following command at the BGP VRF configuration level.

```
device(config-bgp-ipv4u-vrf)# neighbor 10.33.36.2 allowas-in 3
```

Syntax: [no] neighbor *IPaddress* **allowas-in** *asn_limit*

The *IPaddress* is the IP address of the neighbor CE device from which the PE device can accept routes that have the same AS number.

The asn_limit value prevents loops by limiting the number of occurrences that the PEs AS number can be accepted in routes that are received from the specified device.

Setting up LSPs per VRF

IBGP is used between PEs to determine routes that are available between VRFs. These routes are linked to a Label Switched Path (LSP) that has been defined separately either as a static path or using LDP or RSVP. The LSP is used to tunnel through the MPLS domain to the destination PE. Under most circumstances, the default route between two PEs is chosen by IBGP between the VRFs with the PEs loopback address as the next hop. When there is a single loopback on the PE, the same LSP tunnel is the only path used between any VRF defined on a PE and VRFs on other specified PEs.

More than one LSP can be configured between PEs however, where each LSP is associated with a different Loopback address on the PE. In this case, any loopback address on a PE can be assigned as the nexthop address for a specific or multiple VRFs. This allows the user to assign some VRFs on a PE to one LSP and other VRFs to a different LSP. Through this method, traffic from different VRFs can be assigned to LSPs that provide different qualities of service. This feature can also be employed to provide for load-balancing across the MPLS domain.

To configure a PE device to use different LSPs, a BGP next hop must be configured for a VRF as the following example illustrates.

```
        device(config)# vrf blue
device(config-vrf-blue)# bgp next-hop loopback 2
device(config-vrf-blue)# exit-vrf
device(config)#
```

Syntax: [no] bgp next-hop *loopback-interface*

The *loopback-interface* variable is the number of the loopback interface that the user is assigning to the VRF as a BGP next hop. The loopback address becomes the defined VRF's nexthop for its VPNv4 routes that are sourced by this device only when:

- The loopback interface exists and has an IP address set.
- The loopback interface has an IP a subnet mask of /32
- The loopback interface is in the default VRF.

When these conditions are not met, the default nexthop is used.

For a detailed example of this feature refer to

Configuring OSPF sham links

OSPF can be used to propagate links between a Customer Edge device (CE) and a Provider Edge device (PE). Normal operation of this type of network assumes that the only connections between CEs pass through the provider network. However, when other links or routes between the CEs exist within the same area, problems can arise due to the OSPF preference for Intra-area links over Inter-area links.

Problems can be avoided by creating a virtual intra-area OSPF link between two PEs. This virtual link is called a sham link. When the OSPF instances exist in the same area, a sham link causes OSPF to treat the route through the service provider network as an intra-area link instead of an inter-area link.

> **NOTE**
> When no backdoor link exists, no purpose exists for creating a sham link.

A cost is assigned to the sham link to help the OSPF network determine when to route over the sham link and when to route over the backdoor link. Because this virtual link (sham-link) appears as an intra-area link, the OSPF areas in which each of the PEs reside must be the same.

To configure an OSPF sham link, use the command for creating a sham link on both the local device and the remote PE device. Before attempting to create a sham link, note the following important information:

- For sham links to work, OSPF cannot be configured on the loopback interface in the applicable area.
- The redistribution of BGP to OSPF must be configured.

- A BGP VPN4 route to the loopback address must exist in both of the pertinent VRFs' routing tables.
- After the BGP VPN4 route exists in the VRF IP route table, the hello (and other) packet exchanges can go through for sham links even when the backdoor CE link does not exist.

The first example that follows illustrates the command for creating an OSPF sham link between PE devices. The command shows the command entry on one device with a source IP address of 10.2.2.1 and destination address of 10.2.2.2. The second example shows the complete configuration sequence (from both PE devices) and uses the **show ip route vrf** command that is used for viewing the sham link.

Use this command in the OSPF VRF configuration level.

```
device(config-ospf-router)# area 1 sham-link 10.2.2.1 10.2.2.2 cost 10
```

Syntax: [no] area *area_id* sham-link *source_address /destination_address* **cost** *cost_value*

Possible values :

The *area_id* variable is the ID number of the OSPF area assigned to the sham ink being defined in this command.

The *source_address* variable is the IP address of the source PE device.

The *destination_address* variable is the IP address of the destination PE device.

The *cost_value* variable sets the OSPF cost for sending packets over the sham link. This parameter can be a numeric value in the range 1 - 65535.

Sham link configuration on PE1

The following illustrates the configuration for PE1:

```
router ospf vrf CustomerA
area 1
area 1 sham-link 172.31.255.1 172.31.255.2 cost 1
redistribution bgp

interface loopback 2
vrf forwarding CustomerA
ip address 172.31.255.1/32
!
device# show ip route vrf CustomerA 172.31.255.2

Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
Destination Gateway Port Cost Type Uptime
1 172.31.255.2/32 172.30.255.48 lsp PE1-PE2 200/0 B 10m3s
```

Sham link configuration on PE2

The following illustrates the configuration for PE2:

```
router ospf vrf CustomerA
area 1
area 1 sham-link 172.31.255.2 172.31.255.1 cost 1
redistribution bgp

interface loopback 2
vrf forwarding CustomerA
ip address 172.31.255.2/32
!
device# show ip route vrf CustomerA 172.31.255.1

Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
```

```
Destination Gateway Port Cost Type
1 172.31.255.1/32 172.30.255.32 lsp PE2-PE1 200/0 B
```

Configuring OSPF on a PE device to redistribute BGP-VPNv4 routes

To allow OSPF route exchange between a specified VRF on a PE device and its associated CE device, OSPF must be configured to redistribute BGP routes from the local AS as described in the following steps:

> **NOTE**
> This feature is not supported on CES 2000 Series or CER 2000 Series devices.

Defining an OSPF instance in a VRF

To define an OSPF instance in VRF VPN1, enter the following command at the OSPF Config level.

```
device(config)# router ospf vrf VPN1
```

Syntax: [no] router ospf vrf *vrf_name*

The *vrf_name* value specifies the name of the VRF that the user is creating an instance of OSPF in.

Creating an OSPF area in an OSPF VRF instance

To create OSPF area 1 in OSPF VRF instance VPN1, enter the following command in the OSPF VRF Instance Config level.

```
device(config-ospf-router)# area 1
```

Syntax: [no] area *area_id*

The *area-id* value is the number of the OSPF area instance being created.

Creating a domain identifier in an OSPF VRF instance

To create OSPF domain identifier 10.0.0.100 in OSPF VRF instance VPN1, enter the following command in the OSPF VRF Instance Config level.

```
device(config-ospf-router)# domain-id 10.0.0.100
```

Syntax: [no] domain-id *domain_identifier*

The *domain_identifier* value specifies an four-byte quantity.

Assigning a domain tag in an OSPF VRF instance

To assign OSPF domain tag 1200 in OSPF VRF instance VPN1, enter the following command in the OSPF VRF Instance Config level.

```
device(config-ospf-router)# domain-tag 1200
```

Syntax: [no] domain-tag *domain_tag*

The *domain_tag* parameter specifies an arbitrary four-byte quantity. It is added in tag fields of Type-5 and Type-7 LSAs generated by a PE device for redistributed BGP-VPNv4 routes.

When not specified, the domain-tag value is calculated from the autonomous system number of the MPLS domain.

Adding a static ARP entry for a VRF

> NOTE
> This features is not supported on CES 2000 Series or CER 2000 Series devices.

To configure a static ARP entry to a VRF enter the following command at the global configuration level.

```
device(config)# arp vrf green 192.168.201.2 2001:DB8.52cf.e840 ethernet 6/3
```

To clear the ARP entries for a specified VRF, enter the following command.

```
device# clear arp vrf blue
```

For additional information regarding the **arp vrf** command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

For additional information regarding the **clear arp vrf** command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Configuring IP TTL to MPLS TTL propagation in an IPVPN

The **vrf-propagate-ttl** and **label-propagate-ttl** commands configure the device to propagate TTL values in an IPVPN between the IP TTL value and the MPLS TTL value, as described in the tables below.

TABLE 48 MPLS TTL propagation behavior with IPVPNs on XMR Series and MLX Series devices.

With vrf-propagate-ttl and label-propagate-ttl configured	Without vrf-propagate-ttl and label-propagate-ttl configured (default)
• At the ingress device, the IP TTL value -1 is copied to both the tunnel label and the VC label. • At the transit device, the tunnel label is decremented by 1. • At the PHP device, the tunnel label TTL is set to the VC label and the tunnel label is popped. • At the egress device, the IP TTL value is set to min (VC label TTL, IP TTL) and the VC label is popped. The IP TTL value is then decremented by 1 when it is being forwarded out of the device.	• At the ingress device, both the tunnel TTL value and the VC label TTL value are set to 255. • At the transit device, the tunnel label is decremented by 1. • At the PHP device, the tunnel label TTL is popped without changing the VC label's TTL. • At the egress device, the VC label is popped without copying the TTL value to the IP packet. The IP TTL value is then decremented by 1 when it is being forwarded out of the device.

To configure a MLX Series, or XMR Series device to propagate the IP TTL values to and from the MPLS TTL values in an IPVPN, enter the **vrf-propagate-ttl** and **label-propagate-ttl** commands, as shown in the following example.

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# vrf-propagate-ttl
device(config-mpls-policy)# label-propagate-ttl
```

Syntax: [no] vrf-propagate-ttl

Using the [no] option returns the condition to the default off state when the **vrf-propagate-ttl** command has been previously configured.

Syntax: [no] label-propagate-ttl

Using the **no** option returns the condition to the default off state when the **label-propagate-ttl** command has been previously configured.

TABLE 49 MPLS TTL propagation behavior with IPVPNs on the CES 2000 Series and the CER 2000 Series devices

> **NOTE**
> The **no label-propagate-ttl** and **vrf-propagate-ttl** commands are not supported on the CES 2000 Series and the CER 2000 Series devices. The propagation of TTL from IP VPN to MPLS and from MPLS to IP VPN is controlled by **propagate-ttl** command.

With propagate-ttl configured (default)	With no propagate-ttl configured
• At the ingress device, the IP TTL value – 1 is copied to both the tunnel label and VC label. • At the transit device, tunnel label TTL is decremented by 1. • At the PHP device, the Tunnel label TTL is not decremented and Tunnel label TTL is set to the VC label TTL and the tunnel label is popped • At the egress device, the VC label TTL is set to IP TTL and the VC label is popped. The IP TTL value is then decremented by 1 when it is being forwarded out of the device.	• At the ingress device, the IP TTL value – 1 is copied to VC label and the Tunnel label TTL is set to 255 • At the transit device, tunnel label TTL is decremented by 1. • At the PHP device, the Tunnel label is popped without changing the VC label TTL. • At the egress device, the VC label popped without copying the TTL value to IP packet. The IP TTL value is then decremented by 1 when it is being forwarded out of the device.

> **NOTE**
> When configuring **no propagate-ttl** on the CES 2000 Series and the CER 2000 Series devices, at PHP, after the outermost Label is popped, the IP header TTL is decremented by 1, therefore the MPLS domain appear as 3 hops instead of 2 hops.

CES 2000 Series or CER 2000 Series devices by default propagates IP TTL values to and from MPLS TTL values in an IPVPN.

To disable TTL propagation, enter the **no propagate-ttl** command as shown in the following example.

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# no propagate-ttl
```

Syntax: [no] propagate-ttl

When the **no propagate-ttl** is configured, enter the **propagate-ttl** command to return to the default behavior.

Configuring a static route within the VRF context

> **NOTE**
> This feature is not supported on CES 2000 Series or CER 2000 Series devices.

To configure a static route entry in a VRF, enter the following command.

```
        device(config)# vrf blue
device(config-vrf-blue)# ip route 10.0.0.0 255.0.0.0 10.1.1.1
```

Syntax: [no] ip route *dest-ip-addr/mask-bits* | *next-hop-ip-addr* [*metric*]

The *dest-ip-addr* is the route's destination. The *dest-mask* is the network mask for the route's destination IP address. Alternatively, the user can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, the user can enter 10.0.0.0 10.255.255.0 as 10.0.0.0/.24. To configure a default route, enter 0.0.0.0 for *dest-ip-addr* and 0.0.0.0 for *dest-mask* (or 0 for the *mask-bits* when the user specifies the address in CIDR format). Specify the IP address of the default gateway using the *next-hop-ipaddr* parameter.

The *next-hop-ip-addr* is the IP address of the next-hop device (gateway) for the route.

The *metric* parameter specifies the cost of the route and can be a number from 1 - 16. The default is 1.

> **NOTE**
> Note that the **ip route** command is executed in VRF configuration mode.

Configuring a backup Virtual Router for VRF using VRRPE

NOTE

This features is not supported on CES 2000 Series or CER 2000 Series devices.

The Backup Virtual Router for VRF using VRRP-E feature allows the user to configure the Virtual Router Redundancy Protocol Extended (VRRP-E) to provide a redundant connection to a VRF instance in a BGP or MPLS VPN. This is accomplished by assigning the VRRP-E interface to a port that is also assigned to the VRF.

Configuration of VRRPE support for a VRF must be accomplished in the order described below.

1. Enable an interface

2. Enable VRF forwarding on the interface

3. Configure an IP address on the interface

4. Enable VRRP-E on the interface and set the VRID

5. Configure the IP address for the Virtual Router

6. Activate the virtual interface

WARNING

The user must configure a VRF on an interface before configuring a Virtual Router (VRRP-E) on it. When the user enables the Virtual Router before the user enables the VRF, the Virtual Router configuration is deleted.

The following example configures a backup virtual device using VRRPE for VRF "blue" on an Ethernet interface.

```
      device(config)# interface ethernet 3/1
device(config-if-3/1)# vrf forwarding blue
device(config-if-3/1)# ip address 10.2.3.1/8
device(config-if-3/1)# ip vrrp-extended vrid 1
device(config-if-3/1-vrid-1)# backup
device(config-if-3/1-vrid-1)# ip-address 10.2.3.10
device(config-if-3/1-vrid-1)# activate
```

Ping and Traceroute for layer-3 VPNs

The Ping and Traceroute utilities have been enhanced to help with management of Layer-3 VPNs:

Ping VRF

NOTE

This feature is not supported on CES 2000 Series or CER 2000 Series devices.

A VRF option has been added to the **ping** command. To use this option, enter the following command.

```
      device# ping vrf blue
```

Syntax: ping vrf { *vrf-name* | *ip-address* }

The *vrf-name* is the name of the VRF that the user wants to send a ping packet to.

The *ip-address* is the IP address containing the VRF to which the user wants to send a ping packet.

Traceroute VRF

NOTE
This feature is not supported on CES 2000 Series or CER 2000 Series devices.

A VRF option has been added to the **traceroute** command. To use this option, enter the following command.

```
device# traceroute vrf 10.10.10.10
```

Syntax: **traceroute vrf** { *vrf-name* | *ip-address* }

The *vrf-name* is the name of the VRF that the user wants to conduct a traceroute to.

The *ip-address* is the IP address containing the VRF to which the user wants to conduct a traceroute.

Generating traps for VRFs

The user can enable and disable SNMP traps for VRFs. VRF traps are enabled by default.

To enable VRF traps after they have been disabled, enter the following command.

```
device(config)# snmp-server enable traps vrf
```

Syntax: **[no] snmp-server enable traps vrf**

Use the **[no]** form of the command to disable VRF traps.

Displaying BGP or MPLS VPNv4 information

The user can display the following information about a BGP or MPLS VPN configuration on the device:

- Displaying VPNv4 route information on page 454
- Displaying VPNv4 route information for a specified IP address on page 456
- Displaying VPNv4 attribute entries information on page 456
- Displaying VPNv4 dampened paths information on page 457
- Displaying VPNv4 filtered routes information on page 458
- Displaying VPNv4 Flap statistics information on page 458
- Displaying VPNv4 route distinguisher information on page 458
- Displaying VPNv4 neighbor information on page 459
- Displaying attribute entries for a specified VPNv4 neighbor on page 465
- Displaying Flap statistics for a specified VPNv4 neighbor by IP address on page 467
- Displaying received ORFs information for a specified VPNv4 neighbor on page 467
- Displaying a specified neighbor VPNv4 routes on page 467
- Displaying routes summary for a specified VPNv4 neighbor on page 470
- Displaying summary route information on page 472
- Displaying the VPNv4 route table on page 472
- Displaying BGP VPNv4 MPLS tag information on page 477

Displaying VPNv4 route information

The user can display route information about VPNv4 routes by entering the following command at any level of the CLI.

```
         device# show ip bgp vpnv4
Total number of BGP VPNv4 Routes: 285
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network         Next Hop          Metric LocPrf Weight      Path
Route Distinguisher: 1:1
*i  10.80.1.1/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.2/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.3/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.4/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.5/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.6/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.7/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.8/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.9/32      10.2.2.2         100    0      206 311 i
*i  10.80.1.10/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.11/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.12/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.13/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.14/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.15/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.16/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.17/32     10.2.2.2         100    0      206 311 i
*i  10.80.1.18/32     10.2.2.2         100    0      206 311 i
--More--, next page: Space, next line: Return key, quit: Control-c
```

This display shows the following information.

TABLE 50 BGP4 summary information

This field...	Displays...
Total number of BGP VPNv4 Routes:	The number of BGP VPNv4 routes.
Status or Status Codes	The route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE** When the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b - NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes. • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - INTERNAL. The route was learned through BGP4. • L - LOCAL. The route originated on this device.

TABLE 50 BGP4 summary information (continued)

This field...	Displays...
	• M - MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** When the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **NOTE** This field appears only when the user enters the **route** option.
Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
Route Distinguisher	A unique ID that is prepended on any address being routed or advertised from a VRF. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" and a unique arbitrary number.
Network	IP address or mask of the destination network of the route.
Next Hop	The next-hop device for reaching the network from this device.
Metric	The value of the route's MED attribute. When the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares route on the basis of local preference, the route with the higher local preference is chosen. The preference can have a value from 0-4294967295
Weight	The value that this route associates with routes from a specific neighbor. For example, when the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The routes AS path.

To clear the VPNv4 routing table, the user must enter the following commands.

```
        device# clear ip bgp vpnv4 neighbor all soft out
    device# clear ip bgp vpnv4 neighbor all soft in
```

Syntax: clear ip bgp vpnv4 { dampening | flap-statistics | neighbor }

The **dampening** parameter clears route flap dampening information.

The **flap-statistics** parameter clears route flap statistics.

The **neighbor** parameter clears BGP neighbors.

Displaying VPNv4 route information for a specified IP address

To display only the routes to a specified network, enter a command such as the following at any level of the CLI.

```
          device# show ip bgp vpnv4 10.2.2.0/24
    Route Distinguisher: 2:1
    Number of BGP Routes matching display condition : 1
    Status codes: s suppressed, d damped, h history, * valid, > best, i internal
    Origin codes: i - IGP, e - EGP, ? - incomplete
       Network                        Next Hop           Metric      LocPrf      Weight      Path
    *i  10.2.2.0/24                     10.4.4.4         1           100      0        ?
```

Syntax: **show ip bgp vpnv4** *ip-address/mask*

The *ip-address/mask* parameter specifies a particular route. When the user also uses the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, when the user specifies **10.157.0.0 longer** , then all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

The number of BGP routes matching display conditions field in this display is described in the table below.

TABLE 51 Route flap dampening statistics

This field...	Displays...
Number of BGP Routes matching display conditions	The number of routes to the network specified as a parameter in the **show ip bgp vpnv4** *ip-addr* command.

Displaying VPNv4 attribute entries information

The route-attribute entries table lists the sets of BGP VPNv4 attributes stored in the device memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the device typically has fewer route attribute entries than routes. To display the route-attribute entries table at any level of the CLI.

```
          device# show ip bgp vpn attribute-entries
          Total number of BGP Attribute Entries: 55
    1      Next Hop  :0.0.0.0           Metric   :0             Origin:IGP
           Originator:0.0.0.0           Cluster List:None
           Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
           Local Pref:100              Communities:Internet
           Extended Community: RT 600:1
           AS Path   :310
           Address: 0x24644060  Hash:45 (0x0100036e) Reference Counts: 0:0:30
    2      Next Hop  :0.0.0.0           Metric   :0             Origin:IGP
           Originator:0.0.0.0           Cluster List:None
           Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
           Local Pref:100              Communities:Internet
           Extended Community: RT 600:1
           AS Path   :311
           Address: 0x24645f48  Hash:47 (0x01000370) Reference Counts: 0:0:30
    3      Next Hop  :2.2.2.2           Metric   :0             Origin:IGP
           Originator:0.0.0.0           Cluster List:None
           Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
           Local Pref:100              Communities:Internet
           Extended Community: RT 100:1 RT 200:1
           AS Path   :206 311
           Address: 0x24645538  Hash:276 (0x0100087a) Reference Counts: 30:0:0
```

This display shows the following information.

TABLE 52 BGP VPNv4 attribute entries

This field...	Displays...
Total number of BGP Attribute Entries	The number of routes contained in the BGP4 route table for this device.

TABLE 52 BGP VPNv4 attribute entries (continued)

This field...	Displays...
Next Hop	The IP address of the next hop device for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • **EGP** - The routes with this set of attributes came to BGP through EGP. • **IGP** - The routes with this set of attributes came to BGP through IGP. • **INCOMPLETE** - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. **Router-ID** shows the device that originated this aggregator
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss. • TRUE - Indicates information loss has occurred • FALSE - Indicates no information loss has occurred **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
Extended Community	The extended community attributes.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This is an internal value used for debugging purposes only.
Hash	This is an internal value used for debugging purposes only.
Reference Counts	This is an internal value used for debugging purposes only.

Displaying VPNv4 dampened paths information

To view BGP VPNv4 paths suppressed due to dampening, enter the following command.

```
device# show ip bgp vpnv4 dampened-paths
```

Displaying VPNv4 filtered routes information

To view BGP VPNv4 filtered paths information, enter the following command.

```
device# show ip bgp vpnv4 filtered-routes
```

Displaying VPNv4 Flap statistics information

To display route flap statistics for all routes, enter the following command at any level of the CLI.

```
device# show ip bgp vpnv4 flap-statistics ?
```

Syntax: **show ip bgp vpnv4 flap-statistics** [**regular-expression** *regular-expression* | *address /mask* [**longer-prefixes**] | **neighbor** *ip-addr* | **filter-list** *num...*]

The *address / mask* parameter specifies a particular route. When the user also uses the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, when the user specifies **10.157.0.0 longer** , then all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

The **as-path-filter***num* parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filter or filters are displayed.

The **neighbor** *ip-addr* parameter displays route flap dampening statistics only for routes learned from the specified neighbor. The user also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor** *ip-addr* **flap-statistics** .

The **regular-expression** *regular-expression* parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters.

This display shows the following information.

TABLE 53 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

The user also can display all the dampened routes by entering the following command: **show ip bgp dampened-paths** .

Displaying VPNv4 route distinguisher information

In order to view the BGP VPNv4 information for routes that contain a specific route distinguisher, enter the following command.

```
device# show ip bgp vpnv4 rd 5:1 detail
Total number of BGP Routes: 34
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
      E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
1       Prefix: 10.6.1.0/24,  Status: I,  Age: 16h9m21s
         NEXT_HOP: 10.4.4.4,  Learned from Peer: 10.4.4.4 (1)
         Out-Label: 500000
          LOCAL_PREF: 100,  MED: 2,  ORIGIN: incomplete,  Weight: 0
            AS_PATH:
           Extended Community: RT 300:1 RT 100:2 RT 100:3
2       Prefix: 10.40.1.1/32,  Status: I,  Age: 16h9m21s
        NEXT_HOP: 10.4.4.4,  Learned from Peer: 10.4.4.4 (1)
         Out-Label: 500000
          LOCAL_PREF: 100,  MED: 2,  ORIGIN: incomplete,  Weight: 0
            AS_PATH:
            Extended Community: RT 300:1 RT 100:2 RT 100:3
```

```
3         Prefix: 10.40.1.2/32,  Status: I,  Age: 16h9m21s
           NEXT_HOP: 10.4.4.4,  Learned from Peer: 10.4.4.4 (1)
          Out-Label: 500000
            LOCAL_PREF: 100,  MED: 2,  ORIGIN: incomplete,  Weight: 0
             AS_PATH:
               Extended Community: RT 300:1 RT 100:2 RT 100:3
```

TABLE 54 BGP VPNv4 route distinguisher entries

This field...	Displays...
Total number of BGP Routes	The number of routes contained in the BGP4 route table that contain the specified RD.
Prefix	The network address and prefix.
Age	The last time an update occurred.
Learned from Peer	The IP address of the neighbor that sent this route.
Out-Label	MPLS label associated with this device.
MED	The route's metric. When the route does not have a metric, this field is blank.
AS Path	The route's AS path.
Extended Community	Extended community attributes associated with this device.

Displaying VPNv4 neighbor information

To view BGP4 configuration information and statistics for VPNv4 neighbors, enter the following command.

```
      device# show ip bgp vpnv4 neighbors
    Total number of BGP Neighbors: 2
1   IP Address: 10.2.2.2, AS: 1 (IBGP), RouterID: 10.2.2.2, VRF: default
    State: ESTABLISHED, Time: 14h47m39s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 21 seconds, HoldTimer Expire in 141 seconds
       UpdateSource: Loopback 1
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent   : 1       40      887       0          0
       Received: 1       35      887       0          0
    Last Update Time: NLRI        Withdraw        NLRI        Withdraw
             Tx: ---         ---         Rx: ---         ---
    Last Connection Reset Reason:Unknown
    Notification Sent:      Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer Negotiated VPNv4 unicast  capability
      Peer configured for IPV4 unicast  Routes
      Peer configured for VPNv4 unicast  Routes
    TCP Connection state: ESTABLISHED
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:  29202, Received: 28108
       Local host: 3.3.3.3, Local  Port: 179
       Remote host: 2.2.2.2, Remote Port: 8079
       ISentSeq:    7683960  SendNext:    7713163  TotUnAck:        0
       TotSent:       29203  ReTrans:           0  UnAckSeq:   7713163
       IRcvSeq:   256457831  RcvNext:   256485940  SendWnd:     65000
       TotalRcv:      28109  DupliRcv:          0  RcvWnd:      65000
       SendQue:           0  RcvQue:            0  CngstWnd:     1479
```

This example shows how to display information for VPNv4 neighbors. None of the other display options are used; thus, all of the information is displayed for all neighbors. The number in the far left column indicates the neighbor for which information is displayed. When the user lists information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the Transmission Control Block (TCB) for the TCP session between the device and a neighbor. These fields are described in detail in section *3.2 of RFC 793, "Transmission Control Protocol Functional Specification"* .

Syntax: show ip bgp vpnv4 neighbors [*ip-addr* [**advertised-routes** [**detail** [*ip-addr* [**/** *mask-bits*]]]] | [**attribute-entries** [**detail**]] | [**flap-statistics**] | [**last-packet-with-error**] | [**received extended-community**] | [**received prefix-filter**] | [**routes** [**best**] | [**detail** [**best**] | [**not-installed-best**] | [**unreachable**]] | [**rib-out-routes** [*ip-addr/mask-bits* | *ip-addr /net-mask* | **detail**]] | [**routes-summary**]]

The *vrf-name* parameter specifies the VRF whose neighbor the user wants to display information about.

The *ip-addr* option lets the user narrow the scope of the command to a specific neighbor. The display is the same as that for the command without this option except that it is limited to only the neighbor specified.

The **advertised-routes** option displays only the routes that the device has advertised to the neighbor during the current BGP4 neighbor session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet's contents are displayed in decoded (human-readable) format.

The **received extended-community** option displays the received extended community Outbound Route Filters (ORFs) received from this neighbor.

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **routes** option lists the routes received in UPDATE messages from the neighbor. The user can specify the following additional options:

- **best** – Displays the routes received from the neighbor that the device selected as the best routes to their destinations.
- **not-installed-best** – Displays the routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes.
- **unreachable** – Displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** – Displays detailed information for the specified routes. The user can refine the information request by also specifying one of the options above (**best** , **not-installed-best** , or **unreachable**).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. The user can display all the routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this device from the neighbor
- Number of routes this device filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor

This display shows the following information.

TABLE 55 BGP4 neighbor information

This field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP - The neighbor is in another AS. • EBGP_Confed - The neighbor is a member of another sub-AS in the same confederation. • IBGP - The neighbor is in the same AS.
RouterID	The neighbor's ID.
Description	The description the user gave the neighbor when the user configured it on the device.
State	The state of the session with the neighbor. The states are from the perspective of this device of the session, not the perspective of the neighbor. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each device: • IDLE - The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND - The neighbor has been administratively shut down. • A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT - BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE - BGP4 is waiting for a TCP connection from the neighbor. **NOTE** When the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT - BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM - BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. When the device receives a KEEPALIVE message from the neighbor, the state changes to Established. When the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED - BGP4 is ready to exchange UPDATE messages with the neighbor. • When there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. **NOTE** When the user displays information for the neighbor using the **show ip bgp neighbor** *ip-addr* command, the TCP receiver queue value is greater than 0.
Time	The amount of time this session has been in its current state.
KeepAliveTime	The KeepAliveTime, which specifies how often this device sends keep alive messages to the neighbor.

TABLE 55 BGP4 neighbor information (continued)

This field...	Displays...
HoldTime	The hold time, which specifies how many seconds the device waits for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead.
PeerGroup	The name of the peer group the neighbor is in, when applicable.
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.
MaximumPrefixLimit	Lists the maximum number of prefixes the device accepts from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, when configured.
Filter-list	Lists the filter list parameters, when configured.
Prefix-list	Lists the prefix list parameters, when configured.
Route-map	Lists the route map parameters, when configured.
Messages Sent	The number of messages this device has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this device has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following:
	• Reasons described in the BGP specifications:
	  – Message Header Error
	  – Connection Not Synchronized
	  – Bad Message Length
	  – Bad Message Type
	  – OPEN Message Error
	  – Unsupported Version Number
	  – Bad Peer AS Number
	  – Bad BGP Identifier
	  – Unsupported Optional Parameter
	  – Authentication Failure
	  – Unacceptable Hold Time
	  – Unsupported Capability
	  – UPDATE Message Error
	  – Malformed Attribute List
	  – Unrecognized Well-known Attribute

TABLE 55 BGP4 neighbor information (continued)

This field...	Displays...
	– Missing Well-known Attribute – Attribute Flags Error – Attribute Length Error – Invalid ORIGIN Attribute – Invalid NEXT_HOP Attribute – Optional Attribute Error – Invalid Network Field – Malformed AS_PATH – Hold Timer Expired – Finite State Machine Error – Rcv Notification
Last Connection Reset Reason (cont.)	• Reasons specific to the implementation: – Reset All Peer Sessions – User Reset Peer Session – Port State Down – Peer Removed – Peer Shutdown – Peer AS Number Change – Peer AS Confederation Change – TCP Connection KeepAlive Timeout – TCP Connection Closed by Remote – TCP Data Stream Error Detected
Notification Sent	When the device receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages. • Message Header Error: – Connection Not Synchronized – Bad Message Length – Bad Message Type – Unspecified • Open Message Error: – Unsupported Version – Bad Peer As – Bad BGP Identifier – Unsupported Optional Parameter – Authentication Failure – Unacceptable Hold Time – Unspecified • Update Message Error: – Malformed Attribute List – Unrecognized Attribute – Missing Attribute – Attribute Flag Error – Attribute Length Error – Invalid Origin Attribute – Invalid NextHop Attribute – Optional Attribute Error – Invalid Network Field – Malformed AS Path – Unspecified • Hold Timer Expired • Finite State Machine Error • Cease

TABLE 55 BGP4 neighbor information (continued)

This field...	Displays...
	• Unspecified
Notification Received	See above.
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states:
	• LISTEN - Waiting for a connection request.
	• SYN-SENT - Waiting for a matching connection request after having sent a connection request.
	• SYN-RECEIVED - Waiting for a confirming connection request acknowledgment after having both received and sent a connection request.
	• ESTABLISHED - Data can be sent and received over the connection. This is the normal operational state of the connection.
	• FIN-WAIT-1 - Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.
	• FIN-WAIT-2 - Waiting for a connection termination request from the remote TCP.
	• CLOSE-WAIT - Waiting for a connection termination request from the local user.
	• CLOSING - Waiting for a connection termination request acknowledgment from the remote TCP.
	• LAST-ACK - Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).
	• TIME-WAIT - Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.
	• CLOSED - There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the device.
Local port	The TCP port the device is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the device.
SentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the device that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the device retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.

TABLE 55 BGP4 neighbor information (continued)

This field...	Displays...
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying advertised routes for a specified VPNv4 neighbor

To display the routes the device has advertised to a specific VPNv4 neighbor, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vpnv4 neighbors 10.2.2.2 advertised-routes
        There are 231 routes advertised to neighbor 10.2.2.2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
        Prefix          Next Hop        Metric      LocPrf    Weight Status
1       10.100.100.30/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
2       10.100.100.29/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
3       10.100.100.28/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
4       10.100.100.27/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
5       10.100.100.26/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
6       10.100.100.25/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
7       10.100.100.24/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
8       10.100.100.23/32 0.0.0.0                    100       0      BE
          AS_PATH: 310
```

Syntax: show ip bgp vpnv4 neighbor *ip-addr* **advertised-routes** [*ip-addr/prefix*]

Displaying attribute entries for a specified VPNv4 neighbor

The neighbor attribute entries table lists the sets of BGP4 attributes stored in device memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the device typically has fewer route attribute entries than routes. To display the route-attribute entries table for a specified VPNv4 neighbor, enter the following command.

```
        device# show ip bgp vpnv4 neighbors 10.2.2.2 attribute-entries
        Total number of BGP Attribute Entries: 35
1       Next Hop  :0.0.0.0           Metric   :0            Origin:IGP
        Originator:0.0.0.0           Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0      Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 600:1
        AS Path   :310
        Address: 0x247194b0  Hash:45 (0x0100036e) Reference Counts: 0:0:30
2       Next Hop  :0.0.0.0           Metric   :0            Origin:IGP
        Originator:0.0.0.0           Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0      Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 600:1
        AS Path   :311
        Address: 0x2471a480  Hash:47 (0x01000370) Reference Counts: 0:0:30
```

Syntax: show ip bgp vpnv4 neighbors *IPaddress* **attribute-entries**

The *IPaddress* variable is the IP address of the neighbor whose attribute entries the user wants to display.

This display shows the following information.

TABLE 56 BGP4 route-attribute entries information

This field...	Displays...
Total number of BGP Attribute Entries	The number attribute entries in the BGP4 route table for this device.
Next Hop	The IP address of the next hop device for routes with this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP through EGP. • IGP - The routes with this set of attributes came to BGP through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • **Router-ID** shows the device that originated this aggregator.
Router ID	
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE - Indicates information loss has occurred • FALSE - Indicates no information loss has occurred **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
Extended Community	The extended community attributes of the device.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This field is for internal Extreme debugging purposes only.
Hash	This field is for internal Extreme debugging purposes only.
Reference Counts	This field is for internal Extreme debugging purposes only.

Displaying Flap statistics for a specified VPNv4 neighbor by IP address

To display flap-statistics for routes learned from the specified VRF neighbor, enter the following command at any level of the CLI.

```
        device(config)# show ip bgp vpnv4 neighbors 10.2.2.2 flap-statistics
    Total number of flapping routes: 0
```

Syntax: show ip bgp vpnv4 *vrf-name* **neighbor** *ip-addr* **flap-statistics**

The *vrf-name* parameter specifies the VPNv4 neighbor the user wants to display flap-statistics for.

The *address / mask* parameter specifies a particular route. When the user also uses the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, when the user specifies **10.157.0.0 longer** , then all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

This display shows the following information.

TABLE 57 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

Displaying received ORFs information for a specified VPNv4 neighbor

To view BGP4 configuration information and statistics for a specified VPNv4 neighbor, enter the following command.

```
        device# show ip bgp vpn neighbors 10.2.2.2 received extended-community Extended-community ORF
    capability was not negotiated
     No Prefix filter ORF received from neighbor 10.2.2.2!
```

Displaying a specified neighbor VPNv4 routes

To view the route table for a specified neighbor, enter the following command.

```
        device# show ip bgp vpnv4 neighbors 10.10.2.3 routes
        There are 30 accepted routes from neighbor 10.10.2.3
    Searching for matching routes, use ^C to quit...
    Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
           E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
           Prefix            Next Hop       Metric    LocPrf    Weight Status
    1      10.100.100.1/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    2      10.100.100.2/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    3      10.100.100.3/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    4      10.100.100.4/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    5      10.100.100.5/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    6      10.100.100.6/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    7      10.100.100.7/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    8      10.100.100.8/32   10.10.2.3                100       0      BE
             AS_PATH: 310
    9      10.100.100.9/32   10.10.2.3                100       0      BE
             AS_PATH: 310
```

Syntax: show ip bgp vpnv4 neighbors *ip-addr* **routes**

For information about the fields in this display, see the following table.

TABLE 58 BGP4 VPNv4 neighbors information

This field...	Displays...
Total number of BGP VPNv4 Routes:	The number of BGP VPNv4 routes.
Status or Status Codes	The route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE** When the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b - NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes. • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - INTERNAL. The route was learned through BGP4. • L - LOCAL. The route originated on this device. • M - MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** When the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **NOTE** This field appears only when the user enters the **route** option.
Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
Route Distinguisher	A unique ID that is prepended on any address being routed or advertised from a VRF. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" and a unique arbitrary number.
Network	IP address or mask of the destination network of the route.
Next Hop	The next-hop device for reaching the network from this device.
Metric	The value of the route's MED attribute. When the route does not have a metric, this field is blank.

TABLE 58 BGP4 VPNv4 neighbors information (continued)

This field...	Displays...
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares route on the basis of local preference, the route with the higher local preference is chosen. The preference can have a value from 0-4294967295
Weight	The value that this route associates with routes from a specific neighbor. For example, when the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The AS path of the route.

Displaying the best routes

To display the routes received from a specific neighbor that are the "best" routes to their destinations, enter a command such as the following at any level of the CLI.

```
device# show ip bgp vpnv4 neighbor 192.168.4.211 routes best
```

Syntax: show ip bgp vpnv4 neighbor *ip-addr* **routes best**

Displaying the best routes that were nonetheless not installed in the IP route table

To display the BGP4 routes received from a specific neighbor that are the "best" routes to their destinations but are not installed in the device's IP route table, enter a command such as the following at any level of the CLI.

```
device# show ip bgp vpnv4 neighbor 192.168.4.211 routes not-installed-best
```

Each of the displayed routes is a valid path to its destination, but the device received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The device always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: show ip bgp vpnv4 neighbor *ip-addr* **routes not-installed-best**

Displaying the routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
device# show ip bgp vpnv4 neighbor 192.168.4.211 routes unreachable
```

Syntax: show ip bgp vpnv4 neighbor *ip-addr* **routes unreachable**

Displaying the Adj-RIB-Out for a VRF neighbor

To display the device's current BGP4 Routing Information Base (Adj-RIB-Out) for a specific VRF neighbor and a specific destination network, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vpnv4 neighbor 10.10.2.3 rib-out-routes
        There are 154 RIB_out routes for neighbor 10.10.2.3
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
        Prefix            Next Hop        Metric      LocPrf      Weight Status
1       10.100.101.30/32  10.10.3.3                   100         0      BE
          AS_PATH: 311
2       10.100.101.29/32  10.10.3.3                   100         0      BE
          AS_PATH: 311
3       10.100.101.28/32  10.10.3.3                   100         0      BE
```

```
                AS_PATH: 311
4       10.100.101.27/32    10.10.3.3                      100         0      BE
                AS_PATH: 311
5       10.100.101.26/32    10.10.3.3                      100         0      BE
                AS_PATH: 311
6       10.100.101.25/32    10.10.3.3                      100         0      BE
                AS_PATH: 311
7       10.100.101.24/32    10.10.3.3                      100         0      BE
                AS_PATH: 311
8       10.100.101.23/32    10.10.3.3                      100         0      BE
                AS_PATH: 311
9       10.100.101.22/32    10.10.3.3                      100         0      BE
                AS_PATH: 311
10      10.100.101.21/32    10.10.3.3                      100         0      BE
                AS_PATH: 311
```

The Adj-RIB-Out contains the routes that the device either has most recently sent to the VRF neighbor or is about to send to the neighbor.

Syntax: show ip bgp vpnv4 neighbor *ip-addr* **rib-out-routes** [*ip-addr/prefix*]

Displaying routes summary for a specified VPNv4 neighbor

To view the route table for a specified VPNv4 neighbor, enter the following command.

```
        device# show ip bgp vpnv4 neighbor 10.10.2.3 routes-summary
1   IP Address: 10.10.2.3
Routes Accepted/Installed:30,  Filtered/Kept:0,  Filtered:0
    Routes Selected as BEST Routes:30
      BEST Routes not Installed in IP Forwarding Table:0
    Unreachable Routes (no IGP Route for NEXTHOP):0
    History Routes:0
NLRIs Received in Update Message:30,  Withdraws:0 (0),  Replacements:0
    NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0
Routes Advertised:154,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:154,  Withdraws:0,  Replacements:0
Peer Out of Memory Count for:
    Receiving Update Messages:0, Accepting Routes(NLRI):0
    Attributes:0, Outbound Routes(RIB-out):0 Outbound Routes Holder:0
```

This display shows the following information.

TABLE 59 BGP4 route summary information for a VPNv4 neighbor

This field...	Displays...
Routes Accepted or Installed	How many routes the has received from the neighbor during the current BGP4 session: • Filtered - Indicates how many of the received routes the device filtered and did not accept. • Filtered or kept - Indicates how many of the received routes the device did not accept or install because they were denied by filters.
Routes Selected as BEST Routes	The number of routes that the device selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).

TABLE 59 BGP4 route summary information for a VPNv4 neighbor (continued)

This field...	Displays...
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: • Withdraws - The number of withdrawn routes the device has received. • Replacements - The number of replacement routes the device has received.
NLRIs Discarded due to	Indicates the number of times the device discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit - The configured maximum prefix amount had been reached. • AS Loop - An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • Invalid Nexthop - The next hop value was not acceptable. • Duplicated Originator_ID - The originator ID was the same as the local device ID. • Cluster_ID - The cluster list contained the local cluster ID, or contained the local device ID (see above) when the cluster ID is not configured.
Routes Advertised	The number of routes the device has advertised to this neighbor: • To be Sent - The number of routes the device has queued to send to this neighbor. • To be Withdrawn - The number of NLRIs for withdrawing routes the device has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the has sent to this neighbor in UPDATE messages: • Withdraws - The number of routes the device has sent to the neighbor to withdraw. • Replacements - The number of routes the device has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the device has run out of BGP4 memory for the neighbor during the current BGP4 session: • Receiving Update Messages - The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes (NLRI) - The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes - The number of times there was no memory for BGP4 attribute entries. • Outbound Routes (RIB-out) - The number of times there was no memory to place a "best" route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying summary route information

To display summary statistics for all the VPNv4 routes in the device's BGP route table, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vpnv4 routes summary
Total number of BGP routes (NLRIs) Installed    : 184
Distinct BGP destination networks               : 184
Filtered bgp routes for soft reconfig           : 0
Routes originated by this router                : 4
Routes selected as BEST routes                  : 184
BEST routes not installed in IP forwarding table : 0
Unreachable routes (no IGP route for NEXTHOP)   : 0
IBGP routes selected as best routes             : 90
EBGP routes selected as best routes             : 90
```

Syntax: show ip bgp vpnv4 routes summary

This display shows the following information.

TABLE 60 BGP VPNv4 summary route information

This field...	Displays...
Total number of BGP VPNv4 routes (NLRIs) Installed	The number of BGP VPNv4 routes the device has installed in the BGP route table.
Distinct BGP VPNv4 destination networks	The number of destination networks the installed routes represent. The BGP route table can have multiple routes to the same network.
Filtered BGP VPNv4 routes for soft reconfig	The number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained.
Routes originated by this device	The number of VPNv4 routes in the BGP route table that this device originated.
Routes selected as BEST routes	The number of VPNv4 routes in the BGP route table that this device has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	The number of BGP VPNv4 routes that are the best BGP VPNv4 routes to their destinations but were not installed in the IP route table because the device received better routes from other sources.
Unreachable routes (no IGP route for NEXTHOP)	The number of routes in the BGP route table whose destinations are unreachable because the next hop is unreachable.
IBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are IBGP routes.
EBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are EBGP routes.

Displaying the VPNv4 route table

When the user wants to view all the VPNv4 routes in a network, the user can display the BGP VPNv4 table using the following method.

To view the BGP VPNv4 route table, enter the following command.

```
        device# show ip bgp vpnv4 routes
Total number of BGP Routes: 288
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix           Next Hop           Metric    LocPrf    Weight Status
Route Distinguisher: 4:1
1      10.6.1.0/24      10.2.2.2           3         100       0      I
         AS_PATH:
2      10.8.1.0/24      10.2.2.2           2         100       0      I
         AS_PATH:
3      10.40.1.1/32     10.2.2.2           4         100       0      I
```

```
        AS_PATH:
  4     10.40.1.2/32        10.2.2.2        4        100        0        I
        AS_PATH:
  5     10.40.1.3/32        10.2.2.2        4        100        0        I
        AS_PATH:
```

Syntax: show ip bgp vpnv4 routes [*ip-addr*] | *num* | [age *secs*] | [as-path-access-list *num*] | [as-path-filter *num,num,...*] | [best] |
[cidr-only] | [community *num* | no-export | no-advertise | internet | local-as] | [community-access-list *num*] | community-filter *num*
| community-reg-expression *regular-expression* | detail | local | neighbor *ip-addr* [next-hop *ip-addr*] | [no-best] | [not-installed-
best] | [prefix-list *string*] | [regular-expression *regular-expression*] | [route-map *map-name*] | [summary] | [unreachable]

The *ip-addr* option displays routes for a specific network.

The *num* option specifies the table entry with which the user wants the display to start. For example, when the user wants to list entries beginning with table entry 100, specify 100.

The age*secs* parameter displays only the routes that have been received or updated more recently than the number of seconds the user specifies.

The as-path-access-list *num* parameter filters the display using the specified AS-path ACL.

The best parameter displays the routes received from the neighbor that the device selected as the best routes to their destinations.

The cidr-only option lists only the routes whose network masks do not match their class network length.

The community option lets the user display routes for a specific community. The user can specify local-as , no-export , no-advertise , internet , or a private community number. The user can specify the community number as either two five-digit integer values of up to 1-65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The community-access-list *num* parameter filters the display using the specified community ACL.

The community-filter option lets the user display routes that match a specific community filter.

The community regular-expression *regular-expression* option filters the display based on a specified community regular expression.

The local option....

The neighbor *ip-addr* option displays the number of accepted routes from the specified BGP neighbor.

The detail option lets the user display more details about the routes. The user can refine the request by also specifying one of the other display options after the detail keyword.

The next-hop *ip-addr* option displays the routes for a given next-hop IP address.

The no-best option displays the routes for which none of the routes to a given prefix were selected as the best route.

The not-installed-best option displays the routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes.

The prefix-list*string* parameter filters the display using the specified IP prefix list.

The regular-expression *regular-expression* option filters the display based on a regular expression.

The route-map*map-name* parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The summary option displays summary information for the routes.

The unreachable option displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.

For information about the fields in this display, see the table below.

TABLE 61 BGP4 VPNv4 information

This field...	Displays...
Total number of BGP VPNv4 Routes:	The number of BGP VPNv4 routes.

Status or Status Codes	The route's status, which can be one or more of the following:
	
	• A - AGGREGATE. The route is an aggregate route for multiple networks.
	
	• B - BEST. BGP4 has determined that this is the optimal route to the destination.
	
	  NOTE
	  When the "b" is shown in lowercase, the software was not able to install the route in the IP route table.
	
	• b - NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes.
	
	• C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation.
	
	• D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable.
	
	• H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now.
	
	• I - INTERNAL. The route was learned through BGP4.
	
	• L - LOCAL. The route originated on this device.
	
	• M - MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B".
	
	  NOTE
	  When the "m" is shown in lowercase, the software was not able to install the route in the IP route table.
	
	• S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
	
	  NOTE
	  This field appears only when the user enters the **route** option.

Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
Route Distinguisher	A unique ID that is prepended on any address being routed or advertised from a VRF. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" and a unique arbitrary number.
Network	IP address or mask of the destination network of the route.
Next Hop	The next-hop device for reaching the network from this device.
Metric	The value of the route's MED attribute. When the route does not have a metric, this field is blank.

TABLE 61 BGP4 VPNv4 information (continued)

This field...	Displays...
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares route on the basis of local preference, the route with the higher local preference is chosen. The preference can have a value from 0-4294967295
Weight	The value that this route associates with routes from a specific neighbor. For example, when the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The AS path of the route.

Displaying the best VPNv4 routes

To display all the VPNv4 routes in the BGP VPNv4 route table for the Extreme device that are the best routes to their destinations, enter a command such as the following at any level of the CLI.

```
          device(config-bgp-router)# show ip bgp vpnv4 routes best
Total number of BGP Routes: 28
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix            Next Hop         Metric     LocPrf     Weight Status
Route Distinguisher: 4:1
1      3.0.0.0/8          192.168.4.106                100        0      BE
          AS_PATH: 65001 4355 701 80
2      4.0.0.0/8          192.168.4.106                100        0      BE
          AS_PATH: 65001 4355 1
3      4.60.212.0/22     192.168.4.106                100        0      BE
          AS_PATH: 65001 4355 701 1 189
4      6.0.0.0/8          192.168.4.106                100        0      BE
          AS_PATH: 65001 4355 3356 7170 1455
5      9.2.0.0/16         192.168.4.106                100        0      BE
          AS_PATH: 65001 4355 701
```

Syntax: show ip bgp vpnv4 routes best

For information about the fields in this display, see the Displaying the VPNv4 route table task.

Displaying best VPNv4 routes that are not in the IP route table

When the Extreme device has multiple routes to a destination, the device selects the route with the lowest administrative distance as the best route, and installs that route in the IP route table.

To display the BGP4 routes that are the "best" routes to their destinations but are not installed in the device's IP route table, enter a command such as the following at any level of the CLI.

```
          device(config-bgp-router)# show ip bgp vpnv4 routes not-installed-best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix            Next Hop         Metric     LocPrf                 Weight Status
1      10.0.0.0/8                                    192.168.4.106          100      0      BE
          AS_PATH: 65001 4355 701 80
```

Each of the displayed routes is a valid path to its destination, but the device received another path from a different source that has a lower administrative distance. The device always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: show ip bgp vpnv4 routes not-installed-best

For information about the fields in this display, see the Displaying the VPNv4 route table task.

> **NOTE**
> To display the routes that the Extreme device has selected as the best routes and installed in the IP route table, display the IP route table using the **show ip route** command.

Displaying VPNv4 routes with unreachable destinations

To display BGP VPNv4 routes whose destinations are unreachable using any of the paths in the BGP route table, enter a command such as the following at any level of the CLI.

```
        device(config-bgp-router)# show ip bgp vpnv4 routes unreachable
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix          Next Hop       Metric     LocPrf     Weight Status
1      10.0.0.0/8                                192.168.4.106           100          0     BE
       AS_PATH: 65001 4355 701 80
```

Syntax: **show ip bgp vpnv4 routes unreachable**

For information about the fields in this display, see the Displaying the VPNv4 route table task.

Displaying information for a specific VPNv4 route

To display BGP VPNv4 route information by specifying an IP address within the network, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vpnv4 routes 10.8.1.0/24
Route Distinguisher: 4:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix          Next Hop       Metric     LocPrf     Weight Status
1      10.8.1.0/24     10.2.2.2                             2          100        0     I
         AS_PATH:
Route Distinguisher: 5:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix          Next Hop         Metric     LocPrf     Weight Status
1      10.8.1.0/24                      10.4.4.4     3          100        0     I
         AS_PATH:
```

Syntax: **show ip bgp vpnv4 routes** *ip-address*/*prefix* [**longer-prefixes** | *ip-addr*]

Displaying VPNv4 route details

Here is an example of the information displayed when the user uses the **detail** option. In this example, the information for one route is shown.

```
        device# show ip bgp vpnv4 routes detail
Total number of BGP Routes: 288
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
Route Distinguisher: 4:1
1      Prefix: 10.6.1.0/24,  Status: I,  Age: 15h36m10s
       NEXT_HOP: 10.2.2.2,  Learned from Peer: 10.2.2.2 (1)
       Out-Label: 500000
         LOCAL_PREF: 100,  MED: 3,  ORIGIN: incomplete,  Weight: 0
```

```
            AS_PATH:
            Extended Community: RT 300:1 OSPF DOMAIN ID:0.0.0.0 OSPF RT 0:1:0 OSPF ROUTER ID:0.0.0.0
```

TABLE 62 BGP VPNv4 route information

This field...	Displays...
Prefix	The network address and prefix.
Age	The last time an update occurred.
Learned from Peer	The IP address of the neighbor that sent this route.
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
MED	The route's metric. When the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP through EGP. • IGP - The routes with this set of attributes came to BGP through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Atomic	Whether network information in this route has been aggregated and this aggregation has resulted in information loss. **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The router that originated this aggregator.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the Extreme device learned the route.
Admin Distance	The administrative distance of the route.
Adj_RIB_out	The number of neighbors to which the route has been or is advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.
Extended Community	The device's extended community attributes.

Displaying BGP VPNv4 MPLS tag information

To display the MPLS **in-label** and **out-label** tags in the VPNv4 routes, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vpnv4 tags
        Network           Next Hop           In-Label Out-Label
```

```
Route Distinguisher: 1:1
    10.80.1.1/32       10.2.2.2        -       500003
    10.80.1.2/32       10.2.2.2        -       500003
    10.80.1.3/32       10.2.2.2        -       500003
    10.80.1.4/32       10.2.2.2        -       500003
    10.80.1.5/32       10.2.2.2        -       500003
    10.80.1.6/32       10.2.2.2        -       500003
    10.80.1.7/32       10.2.2.2        -       500003
    10.80.1.8/32       10.2.2.2        -       500003
```

The '**In-Label**' and '**Out-Label**' fields in this display are described in the following table.

TABLE 63 Route flap dampening statistics

This field...	Displays...
In-Label	Local assigned MPLS label value.
Out-Label	Learned MPLS label value

Displaying BGP or MPLS VRF information

The user can display the following information about a BGP or MPLS VRF configuration on the device:

Displaying VRF route information

The user can display information about BGP routes that are contained within a specified VRF route table by entering a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf red
Total number of BGP Routes: 285
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network         Next Hop        RD      Metric LocPrf Weight        Path
*i  10.80.1.1/32                            10.2.2.2  1:1    100    0      206 311 i
*i  10.80.1.2/32        10.2.2.2  1:1    100    0      206 311 i
```

```
*i  10.80.1.3/32       10.2.2.2  1:1    100    0      206 311 i
*i  10.80.1.4/32       10.2.2.2  1:1    100    0      206 311 i
*i  10.80.1.5/32       10.2.2.2  1:1    100    0      206 311 i
--More--, next page: Space, next line: Return key, quit: Control-c
```

To view BGP VPNv4 paths suppressed due to dampening for a specified VRF, enter the following command.

```
device# show ip bgp vrf green dampened-paths
```

To view BGP VPNv4 filtered paths information for a specified VRF, enter the following command.

```
device# show ip bgp vrf green filtered-routes
```

This display shows the following information.

TABLE 64 BGP4 VRF information

This field...	Displays...
Total number of BGP Routes:	The number of BGP routes.
Status or Status Codes	The route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE** When the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b - NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes. • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - INTERNAL. The route was learned through BGP4. • L - LOCAL. The route originated on this Extreme device. • M - MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** When the "m" is shown in lowercase, the software was not able to install the route in the IP route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **NOTE** This field appears only when the user enters the **route** option.

TABLE 64 BGP4 VRF information (continued)

This field...	Displays...
Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
RD	The Route Distinguisher. A unique ID that is prepended on any address being routed or advertised from a VRF. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" and a unique arbitrary number.For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" and a unique arbitrary number.
Network	IP address or mask of the destination network of the route.
Next Hop	The next-hop router for reaching the network from this Extreme device.
Metric	The value of the route's MED attribute. When the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares route on the basis of local preference, the route with the higher local preference is chosen. The preference can have a value from 0-4294967295
Weight	The value that this route associates with routes from a specific neighbor. For example, when the Extreme device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The routes AS path.

Syntax: **show ip bgp vrf** *vrf-name* [*ip-address*/*prefix*] [**attribute-entries** | **dampened-paths** | **filtered-routes**]

The **attribute-entries** parameter displays the set of BGP attributes stored in memory.

The **dampened-paths** parameter displays BGP VPNv4 routes suppressed due to dampening for a specified VRF.

The**filtered-routes** parameter displays BGP VPNv4 filtered paths information for a specified VRF.

Clearing the route table for a specific BGP VRF

To clear the route table for a specific BGP VRF, enter the following command.

```
device# clear ip bgp vrf green
```

Syntax: **clear ip bgp vrf** *vrf-name* [**dampening** | **flap-statistics** | **local** | **neighbor** | **routes** | **traffic**]

The **dampening** parameter clears route flap dampening statistics.

The **flap-statistics** parameter clears route flap statistics.

The **local** parameter clears local information.

The **neighbor** parameter clears the BGP neighbor.

The **routes** parameter clears the BGP routes.

The **traffic** parameter clears BGP traffic counters.

Displaying VRF route information for a specified IP address

To display only the routes to a specified network, enter a command such as the following at any level of the CLI.

```
device# show ip bgp vrf green 10.2.2.0/24

Route Distinguisher: 2:1
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop              Metric LocPrf     Weight     Path
*i  10.2.2.0/24                           10.4.4.4      1      100    0        ?
       Route is advertised to 2 peers:
        10.4.4.4(1)            10.2.2.2(1)
```

Syntax: show ip bgp vrf *vrf-name ip-address*/*mask* [**longer-prefixes**]

The *ip-address*/*mask* parameters specify a particular route. When the user also uses the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, when the user specifies **10.157.0.0 longer** , then all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

The **Number of BGP Routes** matching display conditions field in this display is described in the table below. For information about all other fields in this display, refer to Displaying VRF route information on page 478.

TABLE 65 VRF route information

This field...	Displays...
Number of BGP Routes matching display conditions	The number of routes to the network specified as a parameter in the **show ip bgp vpnv4** command.

Displaying attribute entries information for a specified VRF

The route-attribute entries table lists the sets of BGP attributes stored in the Extreme device's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the device typically has fewer route attribute entries than routes. To display the route-attribute entries table at any level of the CLI.

```
device# show ip bgp vrf green attribute-entries
  Total number of BGP Attribute Entries: 26
1    Next Hop  :192.168.201.2     Metric   :1            Origin:INCOMP
     Originator:0.0.0.0           Cluster List:None
     Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
     Local Pref:100               Communities:Internet
     AS Path   :
     Address: 0x247017ec  Hash:279 (0x03000000) Reference Counts: 1:0:0
2    Next Hop  :192.168.201.2     Metric   :2            Origin:INCOMP
     Originator:0.0.0.0           Cluster List:None
     Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
     Local Pref:100               Communities:Internet
     AS Path   :
     Address: 0x247016d8  Hash:280 (0x03000000) Reference Counts: 1:0:0
3    Next Hop  :192.168.201.2     Metric   :3            Origin:INCOMP
     Originator:0.0.0.0           Cluster List:None
     Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
     Local Pref:100               Communities:Internet
     AS Path   :
     Address: 0x24701900  Hash:281 (0x03000000) Reference Count
```

This display shows the following information.

TABLE 66 BGP VPNv4 attribute entries

This field...	Displays...
Total number of BGP Attribute Entries	The number of routes contained in this Extreme device's BGP4 route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP through EGP. • IGP - The routes with this set of attributes came to BGP through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise zero. **Router-ID** shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE - Indicates information loss has occurred • FALSE - Indicates no information loss has occurred **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This field is used for internal Extreme debugging purposes only.
Hash	This field is used for internal Extreme debugging purposes only.
Reference Counts	This field is used for internal Extreme debugging purposes only.

Displaying dampened paths information for a specified VRF

To view BGP VPNv4 paths suppressed due to dampening for a specified VRF, enter the following command.

```
device# show ip bgp vrf green dampened-paths
```

Displaying filtered routes information for a specified VRF

To view BGP VPNv4 filtered paths information for a specified VRF, enter the following command.

```
device# show ip bgp vrf green filtered-routes
```

Displaying Flap statistics information for a specified VRF

To display flap statistics for a specified VRF, enter the following command at any level of the CLI.

```
device# show ip bgp vrf green flap-statistics
```

Syntax: **show ip bgp flap-statistics** [**regular-expression** *regular-expression* | *address /mask* [**longer-prefixes**] | **neighbor** *ip-addr* | **filter-list** *num...*]

The *address / mask* parameter specifies a particular route. When the user also uses the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, when the user specifies **10.157.0.0 longer** , then all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

The **as-path-filter***num* parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filter or filters are displayed.

The **neighbor** *ip-addr* parameter displays route flap dampening statistics only for routes learned from the specified neighbor. The user also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor** *ip-addr***flap-statistics** .

The **regular-expression** *regular-expression* parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters.

This display shows the following information.

TABLE 67 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

The user can also display all the dampened routes by entering the following command:

show ip bgp dampened-paths .

Displaying BGP neighbor information for a specified VRF

To view BGP4 configuration information and statistics for a specified VRF's neighbors, enter the following command.

```
    device# show ip bgp vrf black neighbor
  Total number of BGP Neighbors: 3
1  IP Address: 10.10.2.3, AS: 206 (EBGP), RouterID: 10.10.2.3, VRF: black
   State: ESTABLISHED, Time: 14h31m51s, KeepAliveTime: 60, HoldTime: 180
     KeepAliveTimer Expire in 4 seconds, HoldTimer Expire in 135 seconds
   Messages:    Open    Update  KeepAlive Notification Refresh-Req
     Sent   : 1      4       871       0          0
     Received: 1      1       873       0          0
   Last Update Time: NLRI       Withdraw        NLRI       Withdraw
             Tx: ---        ---        Rx: ---       ---
   Last Connection Reset Reason:Unknown
   Notification Sent:     Unspecified
   Notification Received: Unspecified
   Neighbor NLRI Negotiation:
```

```
       Peer Negotiated IPV4  unicast  capability
       Peer configured for IPV4 unicast  Routes
    TCP Connection state: ESTABLISHED
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   17543, Received: 16814
       Local host:  10.10.2.1, Local  Port: 8135
       Remote host: 10.10.2.3, Remote Port: 179
       ISentSeq:    3301937 SendNext:    3319481  TotUnAck:         0
       TotSent:       17544 ReTrans:           0  UnAckSeq:   3319481
       IRcvSeq:   466270178 RcvNext:   466286993  SendWnd:       6432
       TotalRcv:      16815 DupliRcv:        285  RcvWnd:       65000
       SendQue:           0 RcvQue:            0  CngstWnd:      1456
```

This example shows how to display information for a specific VRFs neighbor. None of the other display options are used; thus, all of the information is displayed for the neighbor. The number in the far left column indicates the neighbor for which information is displayed. When the user lists information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the device's Transmission Control Block (TCB) for the TCP session between the Extreme device and its neighbor. These fields are described in detail in section *3.2 of RFC 793, "Transmission Control Protocol Functional Specification"* .

Syntax: show ip bgp vrf *vrf-name* **neighbors** [*ip-addr* [**advertised-routes** [**detail** [*ip-addr/mask-bits*]]]] | [**attribute-entries** [**detail**]] | [**flap-statistics**] | [**last-packet-with-error**] | [**received prefix-filter**] | [**received-routes**] | [**routes** [**best**] | [**detail** [**best**] | [**not-installed-best**] | [**unreachable**]] | [**rib-out-routes** [*ip-addr/mask-bits* | *ip-addr /net-mask* | **detail**]] | [**routes-summary**]]

The *vrf-name* parameter specifies the VRF whose neighbor the user wants to display information about.

The *ip-addr* option lets the user narrow the scope of the command to a specific neighbor. The display is the same as that for the command without this option except that it is limited to only the neighbor specified.

The **advertised-routes** option displays only the routes that the Extreme device has advertised to the neighbor during the current BGP4 neighbor session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet's contents are displayed in decoded (human-readable) format.

The **received extended-community** option

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **received-routes** option lists all the route information received in route updates from the neighbor because the soft reconfiguration feature was enabled.

The **routes** option lists the routes received in UPDATE messages from the neighbor. The user can specify the following additional options:

- **best** - Displays the routes received from the neighbor that the Extreme device selected as the best routes to their destinations.
- **not-installed-best** - Displays the routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes.
- **unreachable** - Displays the routes that are unreachable because the Extreme device does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** - Displays detailed information for the specified routes. The user can refine the information request by also specifying one of the options above (best, not-installed-best, or unreachable).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. The user can display all the routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this Extreme device from the neighbor
- Number of routes this Extreme device filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor.

This display shows the following information.

TABLE 68 BGP4 neighbor information

This field...	Displays...
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP - The neighbor is in another AS. • EBGP_Confed - The neighbor is a member of another sub-AS in the same confederation. • IBGP - The neighbor is in the same AS.
RouterID	The neighbor's router ID.
Description	The description the user gave the neighbor when the user configured it on the Extreme device.

State	The state of the Extreme device's session with the neighbor. The states are from this Extreme device's perspective of the session, not the neighbor's perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each router:
	
	• IDLE - The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process.
	    – A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• ADMND - The neighbor has been administratively shut down.
	    – A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes.
	• CONNECT - BGP4 is waiting for the connection process for the TCP neighbor session to be completed.
	• ACTIVE - BGP4 is waiting for a TCP connection from the neighbor.
	
	    NOTE
	    When the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.
	
	• OPEN SENT - BGP4 is waiting for an Open message from the neighbor.
	• OPEN CONFIRM - BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. When the Extreme device receives a KEEPALIVE message from the neighbor, the state changes to

TABLE 68 BGP4 neighbor information (continued)

This field...	Displays...
	Established. When the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED - BGP4 is ready to exchange UPDATE messages with the neighbor. – When there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. **NOTE** When the user displays information for the neighbor using the **show ip bgp neighbor***ip-addr* command, the TCP receiver queue value is greater than 0.
Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this Extreme device sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the Extreme device waits for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead.
PeerGroup	The name of the peer group the neighbor is in, when applicable.
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.
MaximumPrefixLimit	Lists the maximum number of prefixes the Extreme device accepts from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this Extreme device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, when configured.
Filter-list	Lists the filter list parameters, when configured.
Prefix-list	Lists the prefix list parameters, when configured.
Route-map	Lists the route map parameters, when configured.
Messages Sent	The number of messages this Extreme device has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this Extreme device has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

TABLE 68 BGP4 neighbor information (continued)

This field...	Displays...

Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following:
	
	• Reasons described in the BGP specifications:
	  – Message Header Error
	  – Connection Not Synchronized
	  – Bad Message Length
	  – Bad Message Type
	  – OPEN Message Error
	  – Unsupported Version Number
	  – Bad Peer AS Number
	  – Bad BGP Identifier
	  – Unsupported Optional Parameter
	  – Authentication Failure
	  – Unacceptable Hold Time
	  – Unsupported Capability
	  – UPDATE Message Error
	  – Malformed Attribute List
	  – Unrecognized Well-known Attribute
	  – Missing Well-known Attribute
	  – Attribute Flags Error
	  – Attribute Length Error
	  – Invalid ORIGIN Attribute
	  – Invalid NEXT_HOP Attribute
	  – Optional Attribute Error
	  – Invalid Network Field
	  – Malformed AS_PATH
	  – Hold Timer Expired
	  – Finite State Machine Error
	  – Rcv Notification

Last Connection Reset Reason (cont.)	• Reasons specific to the implementation:
	  – Reset All Peer Sessions
	  – User Reset Peer Session
	  – Port State Down
	  – Peer Removed
	  – Peer Shutdown
	  – Peer AS Number Change
	  – Peer AS Confederation Change
	  – TCP Connection KeepAlive Timeout
	  – TCP Connection Closed by Remote
	  – TCP Data Stream Error Detected

Notification Sent	When the Extreme device receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.
	
	• Message Header Error:
	  – Connection Not Synchronized
	  – Bad Message Length
	  – Bad Message Type
	  – Unspecified
	• Open Message Error:
	  – Unsupported Version
	  – Bad Peer As
	  – Bad BGP Identifier
	  – Unsupported Optional Parameter
	  – Authentication Failure

TABLE 68 BGP4 neighbor information (continued)

This field...	Displays...

	– Unacceptable Hold Time
	    – Unspecified
	• Update Message Error:
	    – Malformed Attribute List
	    – Unrecognized Attribute
	    – Missing Attribute
	    – Attribute Flag Error
	    – Attribute Length Error
	    – Invalid Origin Attribute
	    – Invalid NextHop Attribute
	    – Optional Attribute Error
	    – Invalid Network Field
	    – Malformed AS Path
	    – Unspecified
	• Hold Timer Expired
	• Finite State Machine Error
	• Cease
	• Unspecified

Notification Received	See above.
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN – Waiting for a connection request. • SYN-SENT – Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED – Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED – Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 – Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 – Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT – Waiting for a connection termination request from the local user. • CLOSING – Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK – Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT – Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED – There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the Extreme device.
Local port	The TCP port the Extreme device is using for the BGP4 TCP session with the neighbor.

TABLE 68 BGP4 neighbor information (continued)

This field...	Displays...
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the Extreme device.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the Extreme device that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the Extreme device retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying advertised routes for a specified VRF neighbor

To display the routes the Extreme device has advertised to a specific VRFs neighbor, enter a command such as the following at any level of the CLI.

```
R3-2547# show ip bgp vrf black neighbor 10.10.2.3 advertised-routes
      There are 154 routes advertised to neighbor 10.10.2.3
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
      Prefix              Next Hop        Metric      LocPrf     Weight Status
1     10.100.101.30/32    10.10.3.3                              0      BE
         AS_PATH: 311
2     10.100.101.29/32    10.10.3.3                              0      BE
         AS_PATH: 311
3     10.100.101.28/32    10.10.3.3                              0      BE
         AS_PATH: 311
4     10.100.101.27/32    10.10.3.3                              0      BE
         AS_PATH: 311
5     10.100.101.26/32    10.10.3.3                              0      BE
         AS_PATH: 311
6     10.100.101.25/32    10.10.3.3                              0      BE
         AS_PATH: 311
7     10.100.101.24/32    10.10.3.3                              0      BE
         AS_PATH: 311
8     10.100.101.23/32    10.10.3.3                              0      BE
         AS_PATH: 311
9     10.100.101.22/32    10.10.3.3                              0      BE
         AS_PATH: 311
10    10.100.101.21/32    10.10.3.3                              0      BE
         AS_PATH: 311
```

Syntax: **show ip bgp vrf** *vrf-name* **neighbor** *ip-addr* **advertised-routes** [*ip-addr/prefix*]

Displaying neighbor attribute entries for a specified VRF

The neighbor attribute entries table lists the sets of BGP4 attributes stored in the router's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the Extreme device typically has fewer route attribute entries than routes. To display the route-attribute entries table for a specified VRF, enter the following command.

```
        device# show ip bgp vrf black neighbor 10.10.2.3 attribute-entries
        Total number of BGP Attribute Entries: 2
1       Next Hop  :10.10.2.3        Metric   :0               Origin:IGP
        Originator:0.0.0.0          Cluster List:None
        Aggregator:AS Number :0     Router-ID:0.0.0.0         Atomic:None
        Local Pref:100              Communities:Internet
        AS Path   :310
        Address: 0x2470139c  Hash:223 (0x0100036e) Reference Counts: 30:0:60
2       Next Hop  :10.2.2.2                                         Metric   :2
Origin:INCOMP
        Originator:0.0.0.0                                   Cluster List:None
        Aggregator:AS Number :0     Router-ID:0.0.0.0         Atomic:None
        Local Pref:100              Communities:Internet
        Extended Community: RT 600:1 OSPF DOMAIN ID:0.0.0.0 OSPF RT 0:5:1 OSPF ROUTER ID:0.0.0.0
        AS Path   :
        Address: 0x24702310  Hash:992 (0x03000000) Reference Counts: 0:0:90
```

Syntax: show ip bgp *vrf-name* **attribute-entries**

This display shows the following information.

TABLE 69 BGP4 route-attribute entries information

This field...	Displays...
Total number of BGP Attribute Entries	The number of routes contained in this Extreme device's BGP4 route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP through EGP. • IGP - The routes with this set of attributes came to BGP through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • **AS Number** shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • **Router-ID** shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE - Indicates information loss has occurred

TABLE 69 BGP4 route-attribute entries information (continued)

This field...	Displays...
	• FALSE - Indicates no information loss has occurred **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
Extended Community	The VRF's extended community attributes.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.

Displaying flap statistics for a specified VRF neighbor by IP address

To display flap-statistics for routes learned from the specified VRF neighbor, enter the following command at any level of the CLI.

```
R3-2547# show ip bgp vrf black neighbor 10.10.2.3 flap-statistics
Total number of flapping routes: 0
```

Syntax: **show ip bgp vrf** *vrf-name* **neighbor** *ip-addr* **flap-statistics**

The *vrf-name* parameter specifies the VRF whose neighbor the user wants to display flap-statistics for.

The *address / mask* parameter specifies a particular route. When the user also uses the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, when the user specifies **10.157.0.0 longer** , then all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

This display shows the following information.

TABLE 70 Route flap dampening statistics

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4 route table that have changed state and thus have been marked as flapping routes.

Displaying received ORF information for a specified VRF neighbor

To view BGP4 VPNv4 configuration information and statistics for a specified VRFs neighbor, enter the following command.

```
      device # show ip bgp vrf black neighbor 10.10.2.3 received extended-community
Extended-community ORF capability was not negotiated

device# show ip bgp vrf black neighbor 10.10.2.3 received prefix-filter
 No Prefix filter ORF received from neighbor 10.10.2.3!
```

Displaying received routes for a specified VRF neighbor

To view the BGP4 VPNv4 configuration and statistics for specified VRFs neighbor, enter the following command.

```
      device# show ip bgp vrf black neighbor 10.10.2.3 received-routes
Inbound soft reconfiguration not enabled for neighbor 10.10.2.3
```

Displaying a specified VRF neighbor routes

To view the route table for a specified VRF's neighbor, enter the following command.

```
        device# show ip bgp vrf black neighbor 10.10.2.3 routes
        There are 30 accepted routes from neighbor 10.10.2.3
 Searching for matching routes, use ^C to quit...
 Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
        E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
        Prefix            Next Hop        Metric      LocPrf      Weight Status
 1      10.100.100.1/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 2      10.100.100.2/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 3      10.100.100.3/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 4      10.100.100.4/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 5      10.100.100.5/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 6      10.100.100.6/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 7      10.100.100.7/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 8      10.100.100.8/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
 9      10.100.100.9/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
```

Syntax: **show ip bgp vrf** *vrf-name* **neighbor** *ip-addr* **routes**

Displaying the best routes

To display the routes received from a specific neighbor that are the "best" routes to their destinations, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black neighbor 192.168.4.211 routes best
```

Syntax: **show ip bgp vrf** *vrf-name* **neighbor** *ip-addr* **routes best**

Displaying the best routes that were nonetheless not installed in the IP route table

To display the BGP4 routes received from a specific neighbor that are the "best" routes to their destinations but are not installed in the Extreme device's IP route table, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black neighbor 192.168.4.211 routes not-installed-best
```

Each of the displayed routes is a valid path to its destination, but the Extreme device received another path from a different source (such as OSPF, RIP, or a static route) that has a lower administrative distance. The Extreme device always selects the path with the lowest administrative distance to install in the IP route table.

Syntax: **show ip bgp vrf** *vrf-name* **neighbor** *ip-addr* **routes not-installed-best**

For information about the fields in this display, refer to Displaying VRF route information on page 478.

Displaying the routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black neighbor 192.168.4.211 routes unreachable
```

Syntax: **show ip bgp vrf** *vrf-name* **neighbor** *ip-addr* **routes unreachable**

Displaying the Adj-RIB-Out for a VRF neighbor

To display the Extreme device's current BGP4 Routing Information Base (Adj-RIB-Out) for a specific VRF neighbor and a specific destination network, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black neighbor 10.10.2.3 rib-out-routes
        There are 154 RIB_out routes for neighbor 10.10.2.3
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
        Prefix              Next Hop        Metric      LocPrf      Weight Status
1       10.100.101.30/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
2       10.100.101.29/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
3       10.100.101.28/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
4       10.100.101.27/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
5       10.100.101.26/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
6       10.100.101.25/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
7       10.100.101.24/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
8       10.100.101.23/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
9       10.100.101.22/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
10      10.100.101.21/32    10.10.3.3                   100         0      BE
            AS_PATH: 311
```

The Adj-RIB-Out contains the routes that the Extreme device either has most recently sent to the VRF neighbor or is about to send to the neighbor.

Syntax: **show ip bgp vrf** *vrf-name* **neighbor** *ip-addr* **rib-out-routes** [*ip-addr/prefix*]

Displaying VPNv4 routes summary for a specified VRF neighbor

To view the route table for a specified VRFs neighbor, enter the following command.

```
        device# show ip bgp vrf black neighbor 10.10.2.3 routes-summary
1   IP Address: 10.10.2.3
Routes Accepted/Installed:30,  Filtered/Kept:0,  Filtered:0
    Routes Selected as BEST Routes:30
        BEST Routes not Installed in IP Forwarding Table:0
    Unreachable Routes (no IGP Route for NEXTHOP):0
    History Routes:0
NLRIs Received in Update Message:30,  Withdraws:0 (0),  Replacements:0
    NLRIs Discarded due to
        Maximum Prefix Limit:0, AS Loop:0
        Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
        Duplicated Originator_ID:0, Cluster_ID:0
Routes Advertised:154,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:154,  Withdraws:0,  Replacements:0
Peer Out of Memory Count for:
    Receiving Update Messages:0, Accepting Routes(NLRI):0
    Attributes:0, Outbound Routes(RIB-out):0 Outbound Routes Holder:0
```

This display shows the following information.

TABLE 71 BGP4 route summary information for a VRF neighbor

This field...	Displays...
Routes Received	How many routes the Extreme device has received from the neighbor during the current BGP4 session. • Accepted or Installed - Indicates how many of the received routes the Extreme device accepted and installed in the BGP4 route table. • Filtered - Indicates how many of the received routes the Extreme device did not accept or install because they were denied by filters on the device.
Routes Selected as BEST Routes	The number of routes that the Extreme device selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Extreme device received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the Extreme device does not have a valid RIP, OSPF, or static route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages. • **Withdraws** - The number of withdrawn routes the Extreme device has received. • **Replacements** - The number of replacement routes the Extreme device has received.
NLRIs Discarded due to	Indicates the number of times the Extreme device discarded an NLRI for the neighbor due to the following reasons: • **Maximum Prefix Limit** - The Extreme device's configured maximum prefix amount had been reached. • **AS Loop** - An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • **Invalid Nexthop** - The next hop value was not acceptable. • **Duplicated Originator_ID** - The originator ID was the same as the local Extreme device ID. • **Cluster_ID** - The cluster list contained the local cluster ID, or contained the local Extreme device ID (see above) when the cluster ID is not configured.
Routes Advertised	The number of routes the Extreme device has advertised to this neighbor: • **To be Sent** - The number of routes the Extreme device has queued to send to this neighbor. • **To be Withdrawn** - The number of NLRIs for withdrawing routes the Extreme device has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the Extreme device has sent to this neighbor in UPDATE messages: • **Withdraws** - The number of routes the Extreme device has sent to the neighbor to withdraw. • **Replacements** - The number of routes the Extreme device has sent to the neighbor to replace routes the neighbor already has.

TABLE 71 BGP4 route summary information for a VRF neighbor (continued)

This field...	Displays...
Peer Out of Memory Count for	Statistics for the times the Extreme device has run out of BGP4 memory for the neighbor during the current BGP4 session: • **Receiving Update Messages** - The number of times UPDATE messages were discarded because there was no memory for attribute entries. • **Accepting Routes (NLRI)** - The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • **Attributes** - The number of times there was no memory for BGP4 attribute entries. • **Outbound Routes (RIB-out)** - The number of times there was no memory to place a "best" route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised.

Displaying summary route information for a specified VRF

To display summary statistics for all the VPNv4 routes in the device's BGP route table for a specified VRF, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black routes summary
Total number of BGP routes (NLRIs) Installed    : 184
Distinct BGP destination networks               : 184
Filtered bgp routes for soft reconfig           : 0
Routes originated by this router                : 4
Routes selected as BEST routes                  : 184
BEST routes not installed in IP forwarding table : 0
Unreachable routes (no IGP route for NEXTHOP)   : 0
IBGP routes selected as best routes             : 90
EBGP routes selected as best routes             : 90
```

Syntax: **show ip bgp vrf** *vrf-name* **routes summary**

This display shows the following information.

TABLE 72 BGP VPNv4 summary route information

This field...	Displays...
Total number of BGP VPNv4 routes (NLRIs) Installed	The number of BGP VPNv4 routes the Extreme device has installed in the BGP route table.
Distinct BGP VPNv4 destination networks	The number of destination networks the installed routes represent. The BGP route table can have multiple routes to the same network.
Filtered BGP VPNv4 routes for soft reconfig	The number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained.
Routes originated by this Extreme device	The number of VPNv4 routes in the BGP route table that this Extreme device originated.
Routes selected as BEST routes	The number of VPNv4 routes in the BGP route table that this Extreme device has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	The number of BGP VPNv4 routes that are the best BGP VPNv4 routes to their destinations but were not installed in the IP route table because the Extreme device received better routes from other sources.
Unreachable routes (no IGP route for NEXTHOP)	The number of routes in the BGP route table whose destinations are unreachable because the next hop is unreachable.
IBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are IBGP routes.

TABLE 72 BGP VPNv4 summary route information (continued)

This field...	Displays...
EBGP routes selected as best routes	The number of "best" routes in the BGP VPNv4 route table that are EBGP routes.

Displaying a VRF BGP4 route table

When the user wants to view all the BGP routes in a VRF, the user can display the VRFs BGP route table using the following method.

To view a VRF BGP4 route table, enter the following command.

```
        device# show ip bgp vrf black routes
Total number of BGP Routes: 184
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix            Next Hop        Metric      LocPrf      Weight Status
1      10.7.7.7/32                                   0.0.0.0         0          100        32768  BL
          AS_PATH:
2      10.10.2.0/24      0.0.0.0         0           100         32768  BL
          AS_PATH:
3      10.10.3.0/24      0.0.0.0         0           100         32768  BL
          AS_PATH:
4      10.10.4.0/24      0.0.0.0         0           100         32768  BL
          AS_PATH:
5      10.100.100.1/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
6      10.100.100.2/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
7      10.100.100.3/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
8      10.100.100.4/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
9      10.100.100.5/32   10.10.2.3                   100         0      BE
          AS_PATH: 310
10     10.100.100.6/32   10.10.2.3                   100         0      BE
```

The following table shows the field descriptions.

TABLE 73 BGP4 VRF information

This field...	Displays...
Total number of BGP Routes:	The number of BGP routes.
Status or Status Codes	The route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE** When the "b" is shown in lowercase, the software was not able to install the route in the IP route table. • b - NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes. • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation.

TABLE 73 BGP4 VRF information (continued)

This field...	Displays...
	• D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable.
	• H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now.
	• I - INTERNAL. The route was learned through BGP4.
	• L - LOCAL. The route originated on this Extreme device.
	• M - MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B".
	NOTE When the "m" is shown in lowercase, the software was not able to install the route in the IP route table.
	• S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
	NOTE This field appears only when the user enters the **route** option.
Origin code	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
RD	The Route Distinguisher. A unique ID that is prepended on any address being routed or advertised from a VRF. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ":" and a unique arbitrary number.For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ":" and a unique arbitrary number.
Network	IP address or mask of the destination network of the route.
Next Hop	The next-hop router for reaching the network from this Extreme device.
Metric	The value of the route's MED attribute. When the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares route on the basis of local preference, the route with the higher local preference is chosen. The preference can have a value from 0-4294967295
Weight	The value that this route associates with routes from a specific neighbor. For example, when the Extreme device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The routes AS path.

Syntax: show ip bgp vrf *vrf-name* **routes** [*ip-addr*] | *num* | [**age** *secs*] | [**as-path-access-list** *num*] | [**as-path-filter** *num,num,...*] | [**best**] | [**cidr-only**] | [**community** *num* | **no-export** | **no-advertise** | **internet** | **local-as**] | [**community-access-list** *num*] | **community-filter** *num* | **community-reg-expression** *regular-expression* | **detail** | **local** | **neighbor** *ip-addr* [**next-hop** *ip-addr*] | [**no-best**] | [**not-installed-best**] | [**prefix-list** *string*] | [**r egular-expression regular-expression**] | [**route-map** *map-name*] | [**summary**] | [**unreachable**]

The *vrf-name* parameter specifies the VRF whose neighbor the user wants to display information about.

The *ip-addr* option displays routes for a specific network.

The *num* option specifies the table entry with which the user wants the display to start. For example, when the user wants to list entries beginning with table entry 100, specify 100.

The **age***secs* parameter displays only the routes that have been received or updated more recently than the number of seconds the user specifies.

The **as-path-access-list***num* parameter filters the display using the specified AS-path ACL.

The **best** parameter displays the routes received from the neighbor that the Extreme device selected as the best routes to their destinations.

The **cidr-only** option lists only the routes whose network masks do not match their class network length.

The **community** option lets the user display routes for a specific community. The user can specify **local-as** , **no-export,no-advertise** , **internet** , or a private community number. The user can specify the community number as either two five-digit integer values of up to 1-65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list***num* parameter filters the display using the specified community ACL.

The **community-filter** option lets the user display routes that match a specific community filter.

The c**ommunity regular-expression** *regular-expression* option filters the display based on a specified community regular expression.

The **local** option .

The **neighbor** *ip-addr* option displays the number of accepted routes from the specified BGP neighbor.

The **detail** option lets the user display more details about the routes. The user can refine the request by also specifying one of the other display options after the **detail** keyword.

The **next-hop***ip-addr* option displays the routes for a given next-hop IP address.

The **no-best** option displays the routes for which none of the routes to a given prefix were selected as the best route.

The **not-installed-best** option displays the routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the rib-route-limit (or RTM route table size limit) and always-propagate option to allow the propagating those best BGP routes.

The **prefix-list***string* parameter filters the display using the specified IP prefix list.

The **regular-expression** *regular-expression* option filters the display based on a regular expression.

The **route-map***map-name* parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** option displays summary information for the routes.

The **unreachable** option displays the routes that are unreachable because the Extreme device does not have a valid RIP, OSPF, or static route to the next hop.

Displaying the best BGP4 routes

To display all the BGP4 routes in the device's BGP4 route table that are the best routes to their destinations, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black routes best
  Searching for matching routes, use ^C to quit...
  Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
        E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
        Prefix            Next Hop        Metric      LocPrf      Weight Status
  1     10.7.7.7/32                                   0.0.0.0         0          100         32768  BL
```

```
             AS_PATH:
2       10.10.2.0/24        0.0.0.0          0          100        32768  BL
             AS_PATH:
3       10.10.3.0/24        0.0.0.0          0          100        32768  BL
             AS_PATH:
4       10.10.4.0/24        0.0.0.0          0          100        32768  BL
             AS_PATH:
5       10.100.100.1/32     10.10.2.3                   100        0      BE
             AS_PATH: 310
6       10.100.100.2/32     10.10.2.3                   100        0      BE
             AS_PATH: 310
7       10.100.100.3/32     10.10.2.3                   100        0      BE
             AS_PATH: 310
8       10.100.100.4/32     10.10.2.3                   100        0      BE
             AS_PATH: 310
9       10.100.100.5/32     10.10.2.3                   100        0      BE
             AS_PATH: 310
```

Syntax: **show ip bgp vrf** *vrf-name* **routes best**

Displaying best BGP4 routes that are not in the IP route table

When the Extreme device has multiple routes to a destination, the Extreme device selects the route with the lowest administrative distance as the best route, and installs that route in the IP route table.

To display the BGP4 routes for a specified VRF that are the "best" routes to their destinations but are not installed in the device's IP route table, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black routes not-installed-best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix            Next Hop        Metric      LocPrf      Weight Status
1       10.0.0.0/8                                   192.168.4.106           100        0      BE
       AS_PATH: 65001 4355 701 80
```

Each of the displayed routes is a valid path to its destination, but the Extreme device received another path from a different source that has a lower administrative distance. The Extreme device always selects the path with the lowest administrative distance to install in the IP route table.

> **NOTE**
> To display the routes that the Extreme device has selected as the best routes and installed in the IP route table, display the IP route table using the **show ip route** command.

Displaying BGP4 routes whose destinations are unreachable

To display BGP routes for a specified VRF whose destinations are unreachable using any of the paths in the BGP route table, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black routes unreachable
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
Route Distinguisher: 4:1
       Prefix            Next Hop        Metric      LocPrf      Weight Status
1       10.0.0.0/8                                   192.168.4.106           100        0      BE
       AS_PATH: 65001 4355 701 80
```

Displaying information for a specific route

To display BGP VPNv4 route information for a specified VRF by specifying an IP address within the network, enter a command such as the following at any level of the CLI.

```
        device# show ip bgp vrf black routes 10.8.1.0/24
Route Distinguisher: 4:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix            Next Hop        Metric     LocPrf    Weight Status
1      10.8.1.0/24       10.2.2.2                             2          100        0      I
         AS_PATH:
Route Distinguisher: 5:1
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix            Next Hop        Metric     LocPrf    Weight Status
1      10.8.1.0/24       10.4.4.4                             3          100        0      I
         AS_PATH:
```

Displaying route details

Here is an example of the information displayed when the user uses the **detail** option. In this example, the information for one route is shown.

```
        device# show ip bgp vrf black routes detail
Total number of BGP Routes: 288
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
        E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
Route Distinguisher: 4:1
1      Prefix: 10.6.1.0/24,  Status: I,  Age: 15h36m10s
       NEXT_HOP: 10.2.2.2,  Learned from Peer: 10.2.2.2 (1)
       Out-Label: 500000
         LOCAL_PREF: 100,  MED: 3,  ORIGIN: incomplete,  Weight: 0
         AS_PATH:
         Extended Community: RT 300:1 OSPF DOMAIN ID:0.0.0.0 OSPF RT 0:1:0 OSPF ROUTER ID:0.0.0.0
```

The following table shows some of the field descriptions.

TABLE 74 BGP VPNv4 route information

This field...	Displays...
Prefix	The network address and prefix.
Age	The last time an update occurred.
Learned from Peer	The IP address of the neighbor that sent this route.
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 – 4294967295.
MED	The route's metric. When the route has no metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP – The routes with this set of attributes came to BGP through EGP. • IGP – The routes with this set of attributes came to BGP through IGP. • INCOMPLETE – The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP.

TABLE 74 BGP VPNv4 route information (continued)

This field...	Displays...
	When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Atomic	Whether network information in this route has been aggregated and this aggregation has resulted in information loss. **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The router that originated this aggregator.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the Extreme device learned the route.
Admin Distance	The administrative distance of the route.
Adj_RIB_out	The number of neighbors to which the route has been or is advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.
Extended Community	The extended communities the route is in.

Displaying additional BGP or MPLS VPN information

The user can display the following additional information about a BGP or MPLS configuration on the device:

Displaying VRF information

To display IP Information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show vrf
Total number of VRFs configured: 1
Status Codes - A:active, D:pending deletion, I:inactive
Name                  Default RD                IFL ID  vrf|v4|v6             Routes Interfaces
a                     1:1               131071   A | A| A                         14

Total number of IPv4 unicast route for all non-default VRF is 12
Total number of IPv6 unicast route for all non-default VRF is 2

device# show vrf a
VRF a, default RD 1:1, Table ID 1 IFL ID 131071
Label: (Not Allocated),  Label-Switched Mode: OFF
IP Router-Id: 0.0.0.0
  No interfaces
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No export route-map

  Address Family IPv4
    Max Routes: 5120
    Number of Unicast Routes: 12
    No Export VPN route-target communities
    No Import VPN route-target communities
Address Family IPv6
    Max Routes: 128
    Number of Unicast Routes: 2
    No Export VPN route-target communities
    No Import VPN route-target communities
```

Syntax: **show vrf** *vrf-name*

The *vrf-name* parameter specifies the VRF that the user wants to display IP information for.

TABLE 75 Output from the show VRF command

This field…	Displays…
VRF Name	The name of the VRF.
Default RD	The default route distinguisher for the VRF.
Table ID	The table ID for the VRF.
Routes	The total number of IPv4 and IPv6 Unicast routes configured on this VRF.
Label	Display the unique VRF label that has been assigned to the specified VRF.
Label Switched Mode	Displays when Label Switched Mode is ON or OFF.
Max routes	The maximum number of routes that can be configured on this VRF.
Number of Unicast Routes	The number of Unicast routes configured on this VRF.
Interfaces	The interfaces from this Extreme device that are configured within this VRF.
Export VPN route-target communities:	The export route-targets that are configured for this VRF.

TABLE 75 Output from the show VRF command (continued)

This field...	Displays...
Import VPN route-target communities	The import route-targets that are configured for this VRF.
Import route-map	The name of the import route-map when any that is configured for this VRF.
Export route-map	The name of the export route-map when a route-map has been configured for this VRF.

Displaying IP network information for a VRF

To display IP network information for a specified VRF, use the following command at any level of the CLI.

```
        device# show ip network vrf green
  Total IP and IPVPN Cache Entry Usage on LPs:
     Module          Host            Network               Free         Total
       2        26                      0                204774      204800
       5        28      240              204532      204800
```

Syntax: show ip network vrf *vrf-name*

This display shows the following information.

TABLE 76 BGP VPNv4 summary route information

This field...	Displays...
Module	The slot number of the module.
Host	The number of host cache entries.
Network	The number of network cache entries
Free	The number of cache entries that are unused.
Total	The total number of cache entries used and unused.

Displaying the IP route table for a specified VRF

To display the IP routes for a specified VRF, enter the following command at any CLI level.

```
        device# show ip route vrf green
  Total number of IP routes: 99
   Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination        Gateway          Port        Cost        Type
  1        10.5.1.0/24        192.168.201.2   eth 6/3     110/2       O
  2        10.6.1.0/24        10.4.4.4                                       lsp toR4   200/0      B
  3        10.8.1.0/24        10.2.2.2                                       lsp toR2   200/0      B
  4        10.30.1.1/32       192.168.201.2   eth 6/3     110/3       O1
  5        10.30.1.2/32       192.168.201.2   eth 6/3     110/3       O1
  6        10.30.1.3/32       192.168.201.2   eth 6/3     110/3       O1
  7        10.30.1.4/32       192.168.201.2   eth 6/3     110/3       O1
  8        10.30.1.5/32       192.168.201.2   eth 6/3     110/3       O1
  9        10.30.1.6/32       192.168.201.2   eth 6/3     110/3       O1
  10       10.30.1.7/32       192.168.201.2   eth 6/3     110/3       O1
  11       10.30.1.8/32       192.168.201.2   eth 6/3     110/3       O1
```

Syntax: show ip route vrf *vrf-name*

The *vrf-name* parameter specifies the VRF that the user wants to display IP routes for.

The following table lists the information displayed by the **show ip route vrf** command.

TABLE 77 CLI display of IP route table

This field...	Displays...
Total number of IP routes	The total number of IP routes that are in the specified VRP routing table.
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The port through which this Extreme device sends packets to reach the route's destination.
Cost	The route's cost.

Type	The route type, which can be one of the following:
	
	• B - The route was learned from BGP.
	• D - The destination is directly connected to this Extreme device.
	• R - The route was learned from RIP.
	• S - The route is a static route.
	• * - The route is a candidate default route.
	• O - The route is an OSPF route. Unless the user uses the **ospf** option to display the route table, "O" is used for all OSPF routes. When the user does use the **ospf** option, the following type codes are used:
	   – O - OSPF intra area route (within the same area).
	   – IA - The route is an OSPF inter area route (a route that passes from one area into another).
	   – E1 - The route is an OSPF external type 1 route.
	   – E2 - The route is an OSPF external type 2 route.

Displaying ARP VRF information

To display the ARP information for a specified VRF, enter the following command.

```
        device# show arp vrf green
Total number of ARP entries: 9
Entries in VRF green:
     IP Address          MAC Address         Type       Age         Port
1    192.168.201.2       2001:DB8.52cf.e840    Dynamic     0           6/3
```

Syntax: show arp vrf *vrf-name* [**number**] [**ip-address**] [**ethernet** *slot/port*] [**mac-address** *mac-addr*]

The *vrf-name* parameter specifies the VRF that the user wants to display arp entries for.

To clear the ARP table.

```
        device# clear arp vrf green
```

Syntax: clear arp vrf *vrf-name*

Displaying OSPF information for a VRF

To display the OSPF Information for a specified VRF, enter the following command at any CLI level.

```
        device# show ip ospf vrf green
OSPF Version Number          Version 2
Router Id                    192.168.201.1
Domain Id                    10.2.2.2
Domain Tag                   10.2.2.2
ASBR Status                  Yes
ABR Status                   Yes        (1)
```

```
Redistribute Ext Routes from  BGP
External LSA Counter          96
Originate New LSA Counter     1738
Rx New LSA Counter            173
External LSA Limit            14447047
Database Overflow Interval    0
Database Overflow State :     NOT OVERFLOWED
RFC 1583 Compatibility :                      Enabled
```

Syntax: **show ip ospf vrf** *vrf-name* [**area** [*area-id* | **area-ipaddress**]] [**border-routers** *router-id*] [**config**] [**database** [**database-summary** | **external-link-state** [**advertise** *number*]] | **extensive** | **link-state-id** *id-number* | **router-id** *advertising-router-id* | **sequence-number** *HEX*] [**link-state** [**advertise** *number*]] | **sabre**

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF information for.

Displaying OSPF area information for a VRF

To display OSPF Area Information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green area
Indx Area             Type  Cost      SPFR      ABR    ASBR   LSA    Chksum(Hex)
1    0                normal 0         6         0      0      6      00039ba2
2    1                normal 0         6         0      2      6      0003af4b
```

Syntax: **show ip ospf vrf** *vrf-name* **area** [*area-id*] | [*ip-address*]

The *vrf-name* parameter specifies the VRF that the user wants to the OSPF area information for.

The *area-id* parameter shows information for the specified area.

The *ip-address* parameter displays the entry that corresponds to the IP address the user enters.

Displaying OSPF ABR and ASBR information for a VRF

To display OSPF ABR and ABSR Information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green border-routers
     router ID         router type next hop router outgoing interface Area
1    10.2.10.2                             ASBR          192.168.201.2   6/3            1
1    10.5.1.3          ASBR                              192.168.201.2   6/3                  1
```

Syntax: **show ip ospf vrf** *vrf-name* **border-routers** *router-id*

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF ABR and ABSR information for.

The *router-id* parameter specifies the display of OSPF ABR and ABSR information for the router with the specified router ID.

Displaying general OSPF configuration information for a VRF

To display OSPF ABR and ABSR Information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green config
Router OSPF: Enabled
Redistribution: Enabled
Default OSPF Metric: 10
OSPF Auto-cost Reference Bandwidth: Disabled

OSPF Redistribution Metric: Type2

OSPF External LSA Limit: 14447047

OSPF Database Overflow Interval: 0
```

```
      RFC 1583 Compatibility: Enabled

      Router id: 192.168.201.1
      Interface State Change Trap:                     Enabled
      Virtual Interface State Change Trap:             Enabled
      Neighbor State Change Trap:                      Enabled
      Virtual Neighbor State Change Trap:              Enabled
      Interface Configuration Error Trap:              Enabled
      Virtual Interface Configuration Error Trap:      Enabled
      Interface Authentication Failure Trap:           Enabled
      Virtual Interface Authentication Failure Trap:   Enabled
      Interface Receive Bad Packet Trap:               Enabled
      Virtual Interface Receive Bad Packet Trap:       Enabled
      Interface Retransmit Packet Trap:                Disabled
      Virtual Interface Retransmit Packet Trap:        Disabled
      Originate LSA Trap:                              Disabled
      Originate MaxAge LSA Trap:                       Disabled
      Link State Database Overflow Trap:               Disabled
      Link State Database Approaching Overflow Trap:   Disabled

      OSPF Area currently defined:
      Area-ID         Area-Type Cost
      0               normal    0
      1               normal    0
```

Syntax: show ip ospf vrf *vrf-name* **config**

The *vrf-name* parameter specifies the VRF that the user wants to display general OSPF configuration information for.

Displaying OSPF external link state information for a VRF

To display OSPF External Link State Information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green database external-link-state
Index Aging  LS ID          Router          Netmask  Metric   Flag
1     491    10.30.1.6      10.5.1.3        ffffffff 00000001 0000
2     1005   10.40.1.30     192.168.201.1   ffffffff 8000000a 0000
3     765    10.60.1.10     192.168.201.1   ffffffff 8000000a 0000
4     1005   10.40.1.9      192.168.201.1   ffffffff 8000000a 0000
5     491    10.30.1.19     10.5.1.3        ffffffff 00000001 0000
6     765    10.60.1.23     192.168.201.1   ffffffff 8000000a 0000
7     1005   10.40.1.22     192.168.201.1   ffffffff 8000000a 0000
8     765    10.60.1.2      192.168.201.1   ffffffff 8000000a 0000
9     1005   10.40.1.1      192.168.201.1   ffffffff 8000000a 0000
10    491    10.30.1.11     10.5.1.3        ffffffff 00000001 0000
11    765    10.60.1.15     192.168.201.1   ffffffff 8000000a 0000
12    1005   10.40.1.14     192.168.201.1   ffffffff 8000000a 0000
13    491    10.30.1.24     10.5.1.3        ffffffff 00000001 0000
14    491    10.30.1.3      10.5.1.3        ffffffff 00000001 0000
```

Syntax: show ip ospf vrf *vrf-name* **database external-link-state** [**advertise** *num*] | [**extensive**] | [**link-state-id** *ip-addr*] | [**router-id** *ip-addr*] | [**sequence-number** *num(Hex)*] | [**status** *num*]

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF external link state information for.

The **advertise***num* parameter displays the data in the specified LSA packet. The *num* parameter identifies the LSA packet by its position in the Extreme device's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf vrf** *vrf-name***external-link-state** command to display the table.

The **extensive** option displays the data in the LSAs in decrypted format.

The **link-state-id***ip-addr* parameter displays the External LSAs for the LSA source specified by *IP-addr* .

The **router-id***ip-addr* parameter shows the External LSAs for the specified OSPF router.

The **status***num* option shows status information.

The **sequence-number***num (Hex)* parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

Displaying OSPF link state information for a VRF

To display OSPF Link State Information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green database link-state
  Index Area ID          Type LS ID          Adv Rtr          Seq(Hex) Age  Cksum
  1     0                Summ 10.2.10.2                       192.168.201.1   8000001b 1145 0x03fb
  2     0                Summ 192.168.201.0  192.168.201.1    8000001b 1145 0x4d8d
  3     0                Summ 10.8.1.0       192.168.201.1    8000001b 905  0xadc5
  4     0                Summ 10.5.1.0       192.168.201.1    8000001b 1145 0xea12
  5     0                ASBR 10.2.10.2                       192.168.201.1   8000001b 1145 0xf409
  6     0                ASBR 10.5.1.3       192.168.201.1    8000001b 1145 0xbe3a
  7     1                Rtr  192.168.201.1  192.168.201.1    80000088 1145 0xf304
  8     1                Rtr  10.2.10.2                       10.2.10.2
  800000eb 581   0x503d
  9     1                Rtr  10.5.1.3       10.5.1.3         8000005e 1470 0xf8b0
  10    1                Net  192.168.201.1  192.168.201.1    8000001f 1145 0xb5da
  11    1                Net  10.5.1.1       10.2.10.2                        8000004e 1792 0x0fbb
  12    1                Summ 10.8.1.0       192.168.201.1    8000001b 905  0xadc5
```

Syntax: **show ip ospf vrf** *vrf-name* **database link-state** [**advertise** *num*] | [**asbr**] | [**extensive**] | [**link-state-id** *ip-addr*] | [**network**] | [**nssa**] | [**opaque-area**] | [**router**] | [**router-id** *ip-addr*] | [**sequence-number** *num(Hex)*] | [**status** *num*] | [**summary**]

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF link state information for.

The **advertise** *num* parameter displays the hexadecimal data in the specified LSA packet. The *num* parameter identifies the LSA packet by its position in the Extreme device's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf vrf** *vrf-name***external-link-state** command to display the table.

The **asbr** option shows ASBR information.

The **extensive** option displays the LSAs in decrypted format.

The **link-state-id***ip-addr* parameter displays the External LSAs for the LSA source specified by *IP-addr* .

The **network** option shows network information.

The **nssa** option shows network information.

The **opaque-area** option shows information for opaque areas.

The **router-id***ip-addr* parameter shows the External LSAs for the specified OSPF router.

The **sequence-number***num (Hex)* parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

The **status***num* option shows status information.

The **summary** option shows summary information.

Displaying OSPF interface information

To display OSPF interface information for a specified VRF, enter the following command at any CLI level.

```
        device# show ip ospf vrf green interface
ethernet 6/3,OSPF enabled
    IP Address 192.168.201.1, Area 1
    OSPF state DR, Pri 1, Cost 1, Options 2, Type broadcast Events 3
    Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
    DR:  Router ID 192.168.201.1    Interface Address 192.168.201.1
    BDR: Router ID 1.2.10.2         Interface Address 192.168.201.2
    Neighbor Count = 1, Adjacent Neighbor Count= 1
    Neighbor:        192.168.201.2
```

```
        Authentication-Key:None
        MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

Syntax: show ip ospf vrf *vrf-name* **interface** [*ip-addr*]

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF interface information for.

The *ip-addr* parameter displays the OSPF interface information for the specified IP address.

Displaying OSPF neighbor information for a VRF

To display OSPF neighbor information for a specified VRF, enter the following command at any CLI level.

```
        device# show ip ospf vrf green neighbor

  Port  Address         Pri State     Neigh Address  Neigh ID        Ev Opt Cnt
  6/3   192.168.201.1   1   FULL/BDR  192.168.201.2  10.2.10.2                       6  2   0
```

Syntax: show ip ospf vrf *vrf-name* **neighbor** [**router-id** *ip-addr*] | [*num*] | [**detail**]

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF neighbor information for.

The **router-id***ip-addr* parameter displays only the neighbor entries for the specified router.

The *num* parameter displays only the entry in the specified index position in the neighbor table. For example, when the user enters "1", only the first entry in the table is displayed.

The **detail** parameter displays detailed information about the neighbor routers.

Displaying the routes that have been redistributed into OSPF

The user can display the routes that have been redistributed into OSPF for a VRF. To display the redistributed routes, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green redistribute route
  10.6.1.0 10.255.255.0 bgp
  10.8.1.0 10.255.255.0 bgp
  10.40.1.1 10.255.255.255 bgp
  10.40.1.2 10.255.255.255 bgp
```

In this example, four routes have been redistributed from BGP routes.

Syntax: show ip ospf vrf *vrf-name* **redistribute route**

The *vrf-name* parameter specifies the VRF that the user wants to display routes redistributed into OSPF for.

Displaying OSPF route information for a VRF

To display the OSPF route information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green routes
  OSPF Area 0x00000001 ASBR Routes 2:
      Destination     Mask           Path_Cost Type2_Cost Path_Type
      10.2.10.2                      10.255.255.255     1          0           Intra
      Adv_Router      Link_State     Dest_Type State     Tag       Flags
      10.2.10.2                      10.2.10.2                                 Asbr      Valid
  0                                  0000
      Paths Out_Port  Next_Hop       Type      State
      1     6/3       192.168.201.2  OSPF      00 00
```

In this example, four routes have been redistributed from BGP routes.

Syntax: **show ip ospf vrf** *vrf-name* **routes** [*ip-addr*]

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF routes for.

The *ip-addr* parameter specifies a destination IP address. When the user uses this parameter, only the route entries for that destination are shown.

Displaying OSPF sham links

To display the OSPF sham links information for a VRF, enter the **show ip ospf vrf** *vrf-name***sham-links** command at any level of the CLI, as in the following example.

```
        device# show ip ospf vrf CustomerA sham-links
  Sham Link in OSPF instance CustomerA to 10.1.1.2 is UP, Established over lsp(LDP)
   Area 1 source address 10.1.1.1
     Link cost 1 Transmit Delay is 1 sec, State ptpt
     Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
     Adjacency State UP, number of interface events 417
```

Syntax: **show ip ospf vrf** *vrf-name* **sham-links**

The *vrf-name* variable identifies the VRF for which the user wants to display OSPF sham links information.

Displaying OSPF trap status for a VRF

To display the state (enabled or disabled) of the OSPF traps for a specified VRF, enter the following command at any CLI level.

```
        device# show ip ospf vrf green trap
Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                        Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Disabled
Virtual Interface Retransmit Packet Trap:         Disabled
Originate LSA Trap:                               Disabled
Originate MaxAge LSA Trap:                        Disabled
Link State Database Overflow Trap:                Disabled
Link State Database Approaching Overflow Trap:    Disabled
```

Syntax: **show ip ospf vrf** *vrf-name* **trap**

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF trap status for.

Displaying OSPF virtual links for a VRF

To display the OSPF virtual links information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip ospf vrf green virtual-link
  No ospf virtual-link entries available
```

Syntax: **show ip ospf vrf** *vrf-name* **virtual-link** [*num*]

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF virtual links information for.

The *num* parameter displays the table beginning at the specified entry number.

Displaying OSPF virtual neighbor information for a VRF

To display the OSPF virtual neighbor information for a specified VRF, enter the following command at any level of the CLI.

```
device# show ip ospf vrf green virtual neighbor
```

Syntax: **show ip ospf vrf** *vrf-name* **virtual neighbor** [*num*]

The *vrf-name* parameter specifies the VRF that the user wants to display OSPF virtual neighbor information for.

The *num* parameter displays the table beginning at the specified entry number.

Displaying IP extcommunity list information

To display the IP Extcommunity information, enter the following command at any level of the CLI.

```
device# show ip extcommunity-list
ip extcommunity access list 20:
  permit RT 100:1
```

Syntax: **show ip extcommunity-list**

For information about the fields, refer to the following.

TABLE 78 Output of show IP extcommunity list

This field...	Displays...
ip extcommunity access list	The contents of all extended community lists on the Extreme device.

Displaying the IP static route table for a VRF

To display the IP static route table for a VRF, enter the following command at any level of the CLI.

```
device# show ip static route vrf green
IP Static Routing Table - entries:
  IP Prefix                Next Hop              Interface              Dis/Met/
Tag                Name
  10.22.66.0/24                10.22.66.0                -              1/1/0
green
```

Syntax: **show ip static route vrf** *vrf-name*

The *vrf-name* parameter specifies the VRF that the user wants to display the static route table for.

Show run displays the entire name of the static IP route. The **show ip static route** command displays an asterisk (*) after the first twelve characters when the assigned name is thirteen characters or more. The **show ipv6 static route** command displays an asterisk after the first two characters when the assigned name is three characters or more.

Displaying the static ARP table for a VRF

To display the static ARP table for a VRF, enter the following command at any level of the CLI.

```
device# show ip static-arp vrf green
Static ARP table size: 2048, configurable from 2048 to 4096
  Index            IP Address        MAC Address        Port
    1            10.95.6.111              2001:DB8.093b.d210      1/1
    3            10.95.6.123              2001:DB8.093b.d211      1/1
```

Syntax: **show ip static-arp vrf** *vrf-name*

The *vrf-name* parameter specifies the VRF that the user wants to display the static ARP table for.

To clear the static ARP table in a VRF, enter the following command.

```
device# clear arp vrf blue
```

Syntax: **clear arp vrf** *vrf-name*

Displaying TCP connections for a VRF

The **show ip tcp vrf connections** command displays information about each TCP connection on the VRF, including the local IP address, local port number, remote IP address, remote port number and the state of the connection. For example.

```
device# show ip tcp vrf green connections
Local IP address:port <-> Remote IP address:port TCP state    (hdl itc cln pdn)
0.0.0.0:179 <-> 0.0.0.0:0 LISTEN (000100bf: 13, 0, 0)
Total 1 TCP connections
```

Syntax: **show ip tcp vrf** *vrf-name* **connections**

The *vrf-name* parameter specifies the VRF that the user wants to display TCP connections for.

Displaying MPLS statistics for a VRF

> NOTE
> Displaying MPLS statistics for a VRF is supported only on the XMR Series and MLX Series.

To display MPLS statistics on a per-interface basis for a specified VRF, enter the following command at any level of the CLI.

```
device# show mpls statistics vrf green
VRF Name       In-Port(s)    Endpt Out-Pkt    Tnl Out-Pkt
green          e3/1          0                0
               e3/2          0                0
               e3/3          0                0
               e3/4          0                0
               e6/1          0                0
               e6/2          4367535952       0
               e6/3          0                4366414365
               e6/4          0                0
```

Syntax: **show mpls statistics vrf** *vrf-name*

The *vrf-name* parameter specifies the VRF that the user wants to display MPLS statistics for.

For information about the fields in this display, refer the following.

TABLE 79 Output from the show MPLS statistics VRF command

This field...	Displays...
VRF Name	The name of the VRF MPLS statistics are being collected for.
In-Ports	The port where the traffic is received.
Endpt Out-Pkt	The number of packets transmitted out of local endpoints.
Tnl Out-Pkt	The number of packets transmitted out of lsp tunnels.

To clear the MPLS statistics counters.

```
device# clear mpls statistics
```

Syntax: **clear mpls statistics** [**label** | **tunnel** | **vpls** | **vrf**]

The **label** parameter clears in-label statistics.

The **tunnel** parameter clears MPLS tunnel statistics.

The **vpls** parameter clears VPLS statistics.

The **vrf** parameter clears vrf statistics.

Displaying IP route information for a VRF

Display IP route information for a specified VRF by entering the following command.

> NOTE
> When BGP and static routes use an MPLS tunnel as the outgoing interface, the Gateway field displays DIRECT in the output of the **show ip route vrf** *vrf-name* command. This is only applicable when displaying IPv4 routes.

```
        device# show ip route vrf yellow
Total number of IP routes: 2
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination        Gateway          Port        Cost       Type
1       10.8.8.8/32                                      DIRECT           loopback 1  0/0        D
2       10.9.9.8/32                                      DIRECT           lsp to1     200/0      B
```

Syntax: **show ip route vrf** *vrf-name* [*num* | *ip-addr* | **bgp** | **connected** | **isis** | **ospf** | **rip** | **static** | **tags**]

The *vrf-name* parameter specifies the VRF that the user wants to display IP route information for.

Displaying RIP information for a VRF

To display RIP Information for a specified VRF, enter the following command at any level of the CLI.

```
        device# show ip rip vrf black
RIP Summary
  Default port 520
     Administrative distance is 120
     Updates every 30 seconds, expire after 180
     Holddown lasts 180 seconds, garbage collect after 120
     Last broadcast  27, Next Update 29
     Need trigger update  0, Next trigger broadcast 3
     Minimum update  interval 25, Max update Interval 5
     Split horizon is on; poison reverse is off
     Import metric 1
     Prefix List, Inbound : Not set
     Prefix List, Outbound : Not set
     Route-map, Inbound : Not set

     Route-map, Outbound : Not set
     Redistribute: BGP  Metric : 0  Routemap : Not Set
```

Syntax: **show ip rip vrf** *vrf-name* [**interface ethernet** *slot/port* | **route** *ip-address*]

The *vrf-name* parameter specifies the VRF that the user wants to display IP route information for.

To clear all RIP routes from a specified VRF, enter the following command.

```
        device# clear ip rip routes vrf blue
```

Syntax: **clear ip rip routes vrf** *vrf-name*

To clear all local RIP routes from a specified VRF, enter the following command.

```
        device# clear ip rip local routes vrf blue
```

Syntax: **clear ip rip local routes vrf** *vrf-name*

BGP or MPLS VPN sample configurations

This section presents examples of typical MPLS configurations. The following sample configurations are presented:

Basic configuration example for IBGP on the PEs

PE routers use IBGP to exchange VRF routes. As in all BGP configurations, this is accomplished by configuring BGP neighbors where the user wants to exchange routes. When the neighbors are configured in the same AS, it is an IBGP configuration. In addition, because MPLS LSPs are made between router loopback addresses, the [update-source loopback] parameters must be used. The following diagram shows two PE routers (PE 1 and PE 4) that are configured as BGP neighbors.

FIGURE 83 IBGP example

MPLS Layer-3 VPN
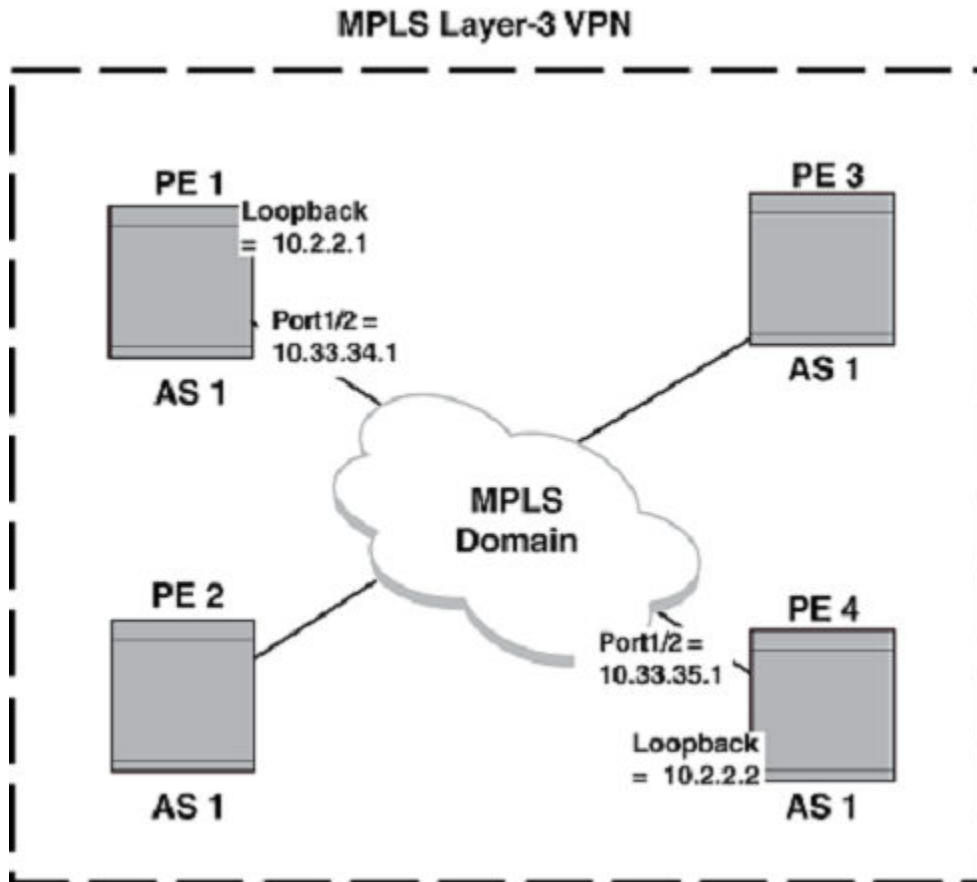
To configure IBGP on a Provider Edge router (PE) of a BGP or MPLS VPN network, the user must perform the configuration steps listed below.

1. Assigning an AS number to a PE on page 514
2. Assigning a loopback interface on page 515
3. Configuring an IBGP neighbor on a PE on page 515

Assigning an AS number to a PE

In the IBGP configuration used in a BGP or MPLS VPN, all PEs are configured with the same AS number. To assign the local AS number 1 to the PE 1 router as shown in Figure 84 on page 514, enter the following commands.

```
        device(config)# router bgp
device(config-bgp)# local-as 1
```

Assigning a loopback interface

A loopback interface is used as the termination for address for BGP sessions. This allows BGP to stay up even when the outbound interface is down as long as an alternate path is available. To install the loopback interface on PE 1 as shown in Figure 84 on page 514, enter the following commands.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.1/32
```

Configuring an IBGP neighbor on a PE

Other PEs that the user wants to exchange IBGP routes with must be configured as BGP neighbors. In addition, the neighbor must be set to enable the BGP to update the loopback address. To assign an IBGP neighbor with the IP address 10.33.35.1, a remote AS number of 1, and an update-source to loopback 1 of the PE 1 router shown in Figure 84 on page 514, enter the following commands.

```
        device(config-bgp)# neighbor 10.2.2.2 remote-as 1
device(config-bgp)# neighbor 10.2.2.2 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.2 activate
```

Configuring EBGP on a CE router

Allows route exchanges between a CE router and its associated PE router by enabling BGP on a customer edge (CE) router and configuring an associated premises edge (PE) router as a BGP neighbor.

The following task shows the steps required for enabling BGP on a CE device and assigning a PE device as a BGP neighbor. For an example of a full configuration required to exchange routes in an external BGP (EBGP) network, see the EBGP for route exchange task.

1. On a CE device, enter the **configure terminal** command.

```
        device# configure terminal
```

2. Enter the **router bgp** command to enable BGP routing.

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

3. Enter the **local-as** command to configure the AS number.

```
        device(config-bgp)# local-as 2
```

4. Configure a PE BGP neighbor using the **neighbor remote-as** command.

```
        device(config-bgp)# neighbor 10.33.33.3 remote-as 1
```

The following example enables BGP on a CE and assigns a PE as a BGP neighbor.

```
device# configure terminal
device(config)# router bgp
device(config-bgp)# local-as 2
device(config-bgp)# neighbor 10.33.33.3 remote-as 1
```

Configuring EBGP on a PE router

Allows route exchange between a VRF on a PE router and its associated customer edge (CE) router by enabling BGP on the appropriate VRF of the premises edge (PE) router and configuring the associated CE router as a BGP neighbor.

In this task, a CE is assigned as a BGP neighbor to the VRF VPN1 on a PE device. For an example of a full configuration required to exchange routes in an external BGP (EBGP) network, see the EBGP for route exchange task.

1. On a PE device, enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **router bgp** command to enable BGP routing.

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

3. Enter the **address-family ipv4 unicast** command to assign a VRF.

```
device(config-bgp)# address-family ipv4 unicast vrf VPN1
```

4. Enter the **neighbor remote-as** command to configure a BGP neighbor (a CE device) to the VRF.

```
device(config-bgp-ipv4u-vrf)# neighbor 10.33.33.2 remote-as 2
```

The following example assigns a CE device as a BGP neighbor to the VRF VPN1 on a PE device.

```
device# configure terminal
device(config)# router bgp
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# neighbor 10.33.33.2 remote-as 2
```

EBGP for route exchange

External BGP (EBGP) can be used to exchange routes from CE routers to PE routers.

To exchange routes, a BGP neighbor must be configured on both CE and PE routers. In the diagram shown below, the CE 1 router is configured to exchange routes with the PE 1 router and the CE 5 router is configured to exchange routes with the PE 4 router.

FIGURE 84 EBGP to CE network example
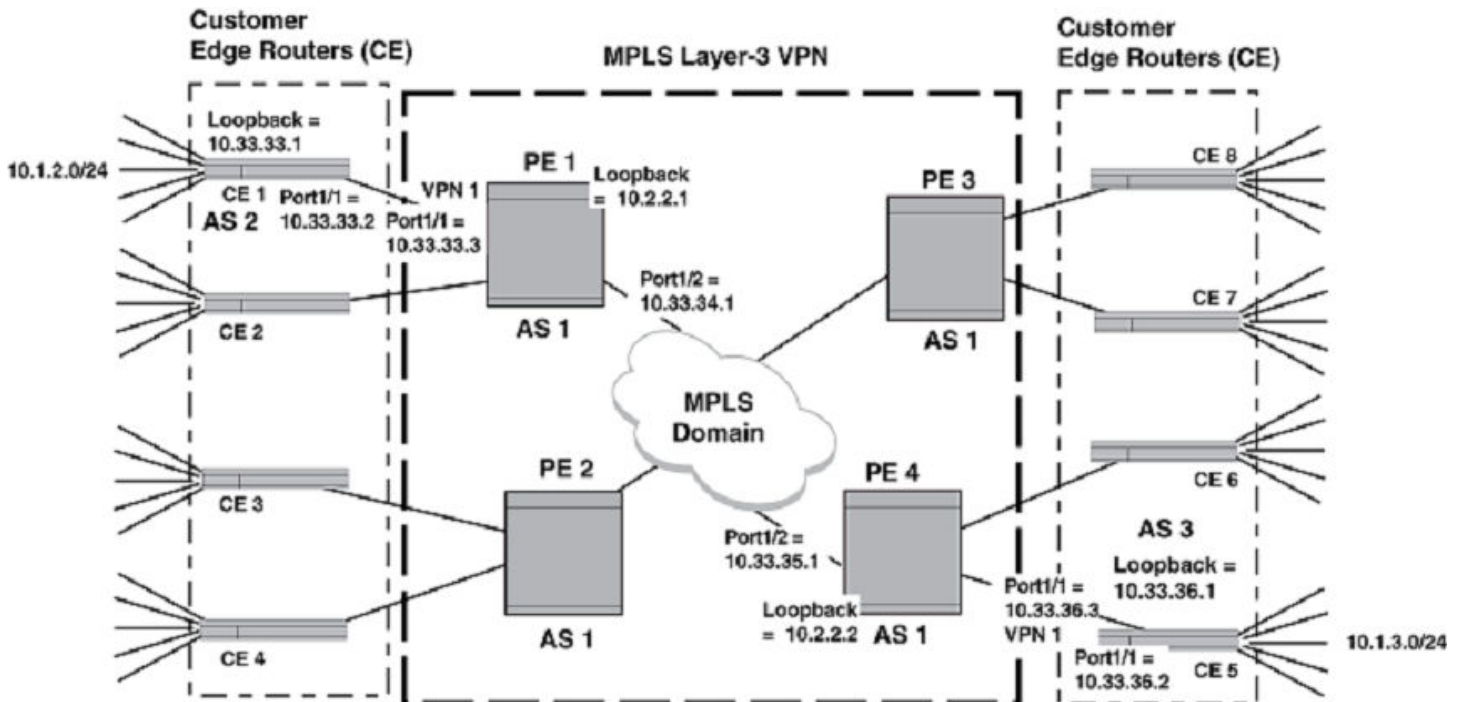
EBGP to CE network example

In the example shown in the diagram above, the network is configured to use EBGP to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS domain. The diagram above contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers, which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in the diagram above. In this example, a static route is configured between the external network (10.1.2.0/24) and the loopback interface. EBGP is configured between CE 1 and PE 1, and the static route is redistributed through this connection.

```
device(config)# ip route 10.1.2.0/24 10.33.33.1
device(config)# router bgp
device(config-bgp)# local-as 2
device(config-bgp)# neighbor 10.33.33.3 remote-as 1
device(config-bgp)# redistribute static
device(config-bgp)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip address 10.33.33.2/24
device(config-if-e10000-1/1)# exit
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in the diagram above. In this example, a static route is configured between the external network (10.1.3.0/24) and the loopback interface. EBGP is configured between CE 5 and PE 4, and the static route is redistributed through this connection.

```
device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.33.36.1/32
device(config-lbif-1)# exit
device(config)# ip route 10.1.3.0/24 10.33.36.1
device(config)# router bgp
device(config)# local-as 3
device(config-bgp)# neighbor 10.33.36.3 remote-as 1
device(config-bgp)# redistribute static
device(config-bgp)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip address 10.33.36.2/24
device(config-if-e10000-1/1)# exit
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. EBGP is configured between VPN1 and CE 1. IBGP with extended community attributes is configured between PE 1 and PE 4.

The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signaled LSP named "tunnel1" to PE 4.

```
device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.1/32
device(config-lbif-1)# ip ospf area 0
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:1
device(config-vrf-vpn1)# route-target export 100:1
device(config-vrf-vpn1)# route-target import 100:2
device(config-vrf-vpn1)# exit-vrf

device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.2 remote-as 1
device(config-bgp)# neighbor 10.2.2.2 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.2 activate
device(config-bgp-vpnv4u)# exit

device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# neighbor 10.33.33.2 remote-as 2
device(config-bgp-ipv4u-vrf)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# exit

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip address 10.33.34.1/24
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.2
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.33.3/24
```

PE 4 configuration

This configuration example describes what is required to operate the PE 2 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. EBGP is configured between VPN1 and CE 5. IBGP with extended community attributes is configured between PE 4 and PE 1.

The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signaled LSP named "tunnel1" to PE 1.

```
device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.2/32
device(config-lbif-1)# ip ospf area 0
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:2
device(config-vrf-vpn1)# route-target export 100:2
device(config-vrf-vpn1)# route-target import 100:1
device(config-vrf-vpn1)# exit-vrf

device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.1 remote-as 1
device(config-bgp)# neighbor 10.2.2.1 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.1 activate
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# neighbor 10.33.36.2 remote-as 2
device(config-bgp-ipv4u-vrf)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# exit

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip address 10.33.35.1/24
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.1
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.36.3/24
device(config-if-e10000-1/1)# exit
```

Static routes for route exchange

Static routes can be used to exchange routes between CE routers and PE routers.

To exchange routes, a default static route must be configured on a CE router to its associated PE router. A static route must also be configured between the PE router and the network (or networks) that the PE wants to advertise as available through a VRF. In this task, the network shown below is configured for a default static route to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS domain. The diagram below contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

FIGURE 85 Static route to CE network example

CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in the diagram above. In this example, a default static route is configured between the CE 1 router and the attached interface of PE 1.

```
        device(config)# ip route 0.0.0.0/0 10.33.33.3
    device(config)# interface ethernet 1/1
    device(config-if-e10000-1/1)# ip address 10.33.33.2
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in the diagram above. In this example, a default static route is configured between the CE 5 router and the attached interface of PE 4.

```
        device(config)# ip route 0.0.0.0/0 10.33.36.3
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip address 10.33.34.2
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. A static route is configured between this router and the network connected to CE 1. IBGP with extended community attributes is configured between PE 1 and PE 4.

The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signaled LSP named "tunnel1" to PE 4.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.1/24
device(config-lbif-1)# ip ospf area 0
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:1
device(config-vrf-vpn1)# route-target export 100:1
device(config-vrf-vpn1)# route-target import 100:2
device(config-vrf-vpn1)# exit-vrf

device(config)# ip route vrf VPN1 10.1.2.0/24 10.33.33.2
device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.2 remote-as 1
device(config-bgp)# neighbor 10.2.2.2 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.2 activate
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute static
device(config-bgp-ipv4u-vrf)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# exit

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip address 10.33.34.1/24
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.2
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.33.3/24
device(config-if-e10000-1/1)#
```
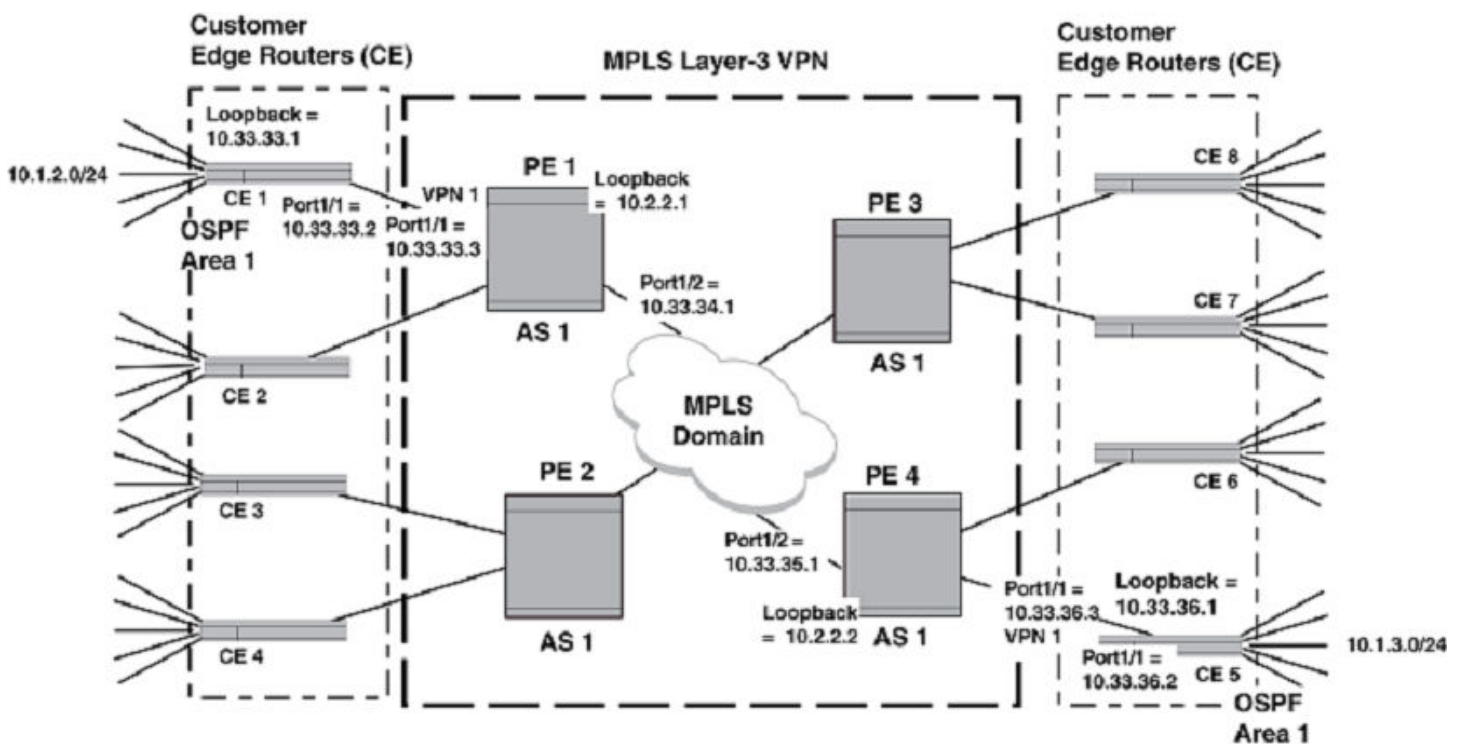
PE 4 configuration

This configuration example describes what is required to operate the PE 4 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. A static route is configured between this router and the network connected to CE 5.

IBGP with extended community attributes is configured between PE 1 and PE 4. The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signalled LSP named "tunnel1" to PE 1.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.2/32
device(config-lbif-1)# ip ospf area 0
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:2
device(config-vrf-vpn1)# route-target export 100:2
device(config-vrf-vpn1)# route-target import 100:1
device(config-vrf-vpn1)# exit-vrf

device(config)# ip route vrf VPN1 10.1.2.0/24 10.33.36.2
device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.1 remote-as 1
device(config-bgp)# neighbor 10.2.2.1 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.1 activate
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute static
device(config-bgp-ipv4u-vrf)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# exit

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip address 10.33.35.1/24
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.1
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.36.3/24
device(config-if-e10000-1/1)# exit
```

Configuring a static default route on a CE router

To allow route exchange between a CE router and its associated PE router, a static default route must be created to the interface on the associated PE router where the VPN is enabled. In this example, the PE 1 router has the VRF "VPN1" enabled on port 1/1, which has the IP address 10.33.33.3. To create a default static route from CE 1 to this interface on PE 1, enter the following command.

```
        device(config)# ip route 0.0.0.0 10.33.33.3
```

Configuring a static default route on a PE router

To allow route exchange between a PE router and its associated CE router, a static route must be created to the route that the user wants to provide access to with a next hop consisting of the IP address of the interface that is connected to the VRF. In this example, the IP address of the connected port on the CE router is 10.33.33.2, and the address on the CE that is provided access from the PE's VRF is 10.1.2.0/24. To create a static route from PE 1 to CE 1, enter the following command.

```
device(config)# ip route vrf VPN1 10.1.2.0/24 10.33.33.2
```

RIP for route exchange

RIP can be used to exchange routes between CE routers and PE routers. In this situation, RIP must be enabled on the CE router and enabled on the interface that is connected to the interface of the PE that is associated with the VRF that the user wants to advertise RIP routes on. On the PE router, the VRF must be enabled to redistribute RIP routes, and RIP must be enabled for the VRF and configured to redistribute routes from BGP in the VRF. The following diagram provides an example of a network where RIP is used to exchange routes between CE routers and PE routers.

FIGURE 86 RIP to CE network example
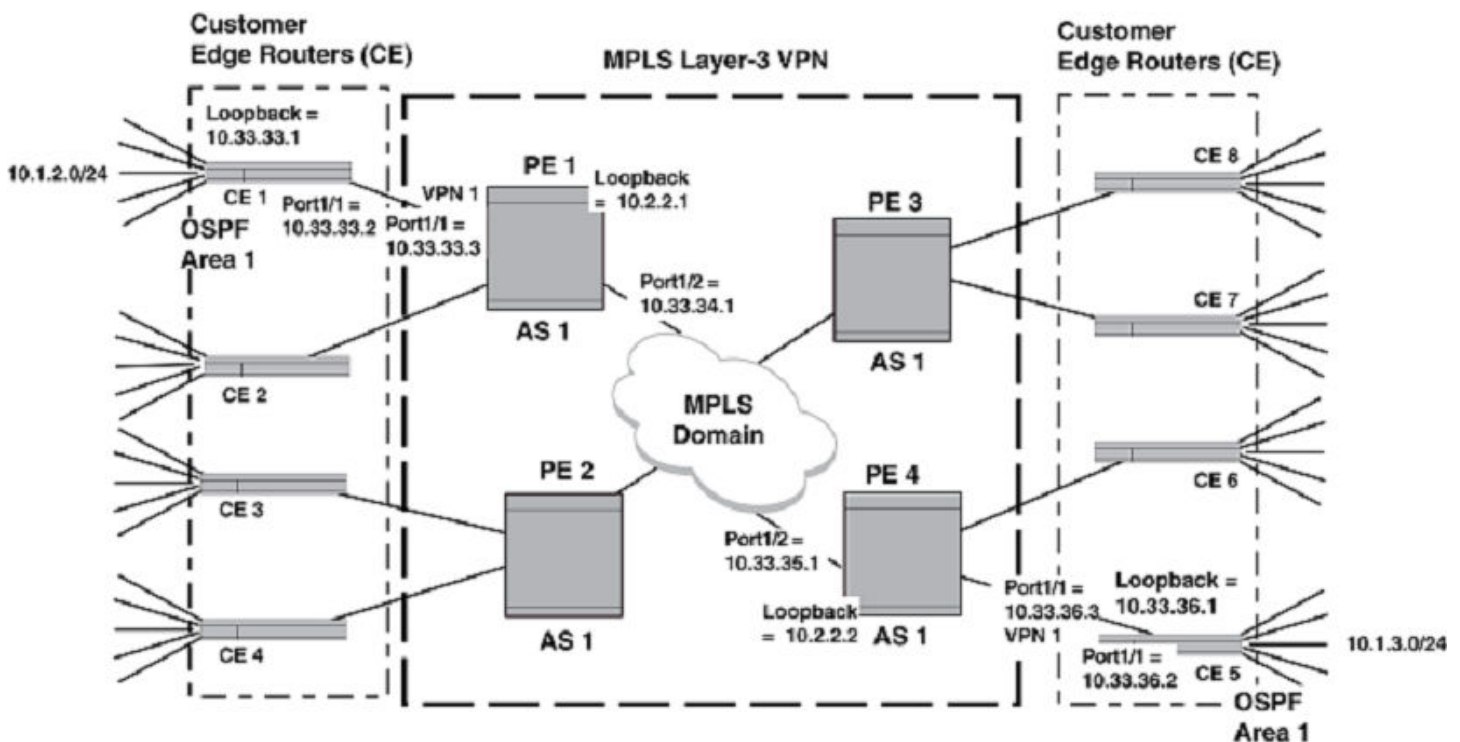
To configure RIP to exchange routes between PE routers and CE routers, the user must perform the configuration steps listed below.

1. Configuring RIP on the CE router on page 525
2. Enabling RIP on the CE router's interface on page 525
3. Configuring the VRF on the PE router to redistribute RIP routes on page 525
4. Configuring RIP on the PE router to redistribute BGP-VPNv4 routes on page 525
5. Enabling RIP on the PE router interface on page 525

Configuring RIP on the CE router

To allow RIP route exchange between a CE router and its associated PE router, RIP must be enabled on the CE router. To configure RIP on the CE 1 router in RIP for route exchange on page 524 and enable it to redistribute static routes through RIP, enter the following commands.

```
        device(config)# router rip
device(config-rip-router)# redistribute static
```

Enabling RIP on the CE router's interface

To allow RIP route exchange between a CE router and its associated PE router, RIP must be enabled on the interface that connects to the VRF-enabled interface of its associated PE router. To configure RIP on the interface of the CE 1 router in RIP for route exchange on page 524 that is connected to the VRF VPN1 associated interface on PE 1, enter the following commands.

```
        device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip rip v2-only
device(config-if-e10000-1/1)# ip address 10.33.33.2
```

Configuring the VRF on the PE router to redistribute RIP routes

To allow RIP route exchange between a specified VRF on a PE router and its associated CE router, the VRF must be enabled redistribute RIP routes. To enable the VRF VPN1 on PE 1 router in RIP for route exchange on page 524 to redistribute RIP routes, enter the following commands.

```
        device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute rip
```

Configuring RIP on the PE router to redistribute BGP-VPNv4 routes

To allow RIP route exchange between a specified VRF on a PE router and its associated CE router, RIP must be configured to redistribute BGP routes from the local AS. To enable RIP on PE 1 in RIP for route exchange on page 524 and configure it to redistribute BGP-VPNv4 routes into RIP, enter the following commands.

```
        device(config)# router rip vrf VPN1
device(config-rip-router)# redistribute bgp
```

Enabling RIP on the PE router interface

To allow RIP route exchange between a PE router and its associated CE router, RIP must be enabled on the PE's interface that is associated with the VRF and connected to the PE router. To configure RIP on the interface of the PE 1 router that is associated with VRF VPN1 in RIP for route exchange on page 524 to CE 1, enter the following commands.

```
        device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.33.3/24
device(config-if-e10000-1/1)# ip rip v2-only
```

RIP to CE configuration example

In this example, the network shown in diagram below is configured for RIP to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS domain. The diagram below contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers, which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

FIGURE 87 RIP to CE network example
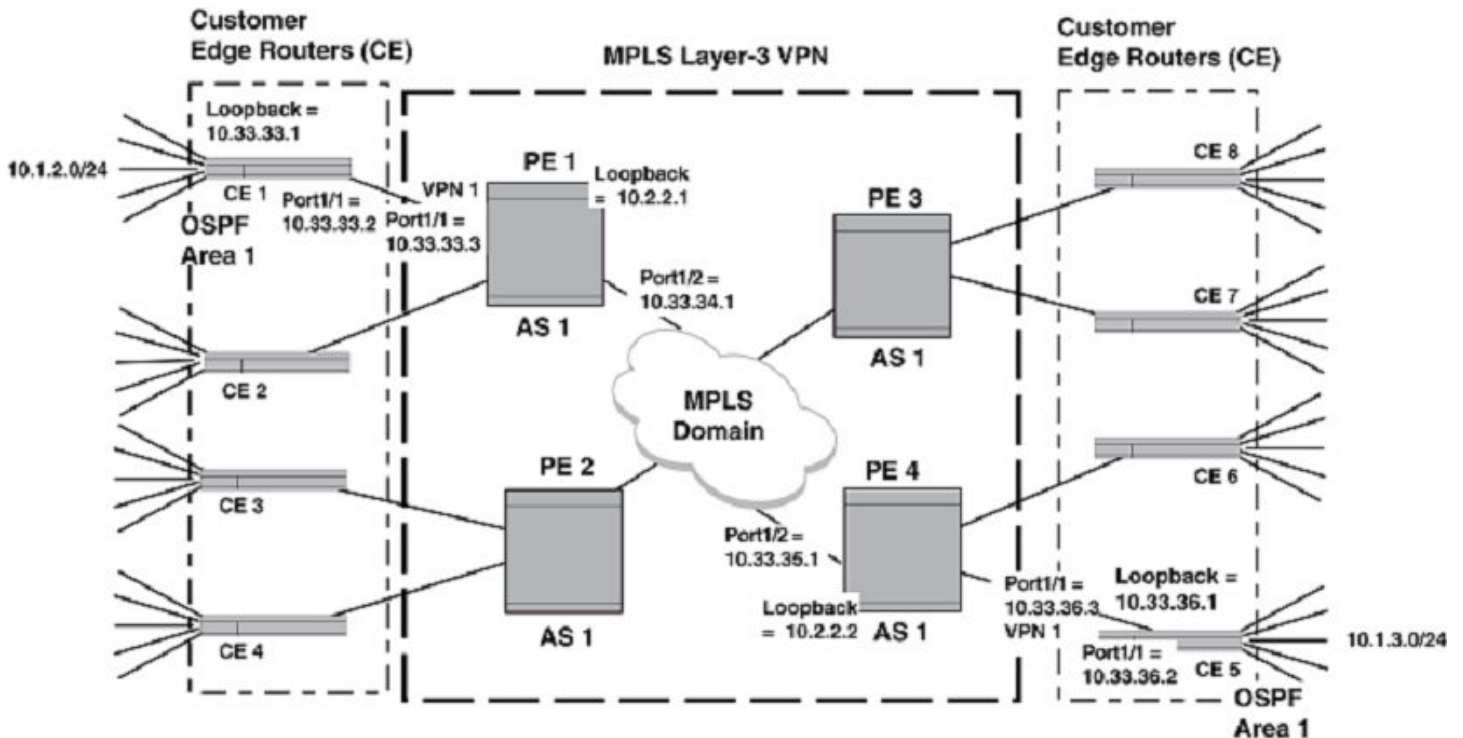
CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in the diagram above. In this example, a static route is configured between the external network (10.1.2.0/24) and the loopback interface. RIP is configured to redistribute static routes between the CE 1 router and the attached interface of PE 1.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.33.33.1/32
device(config-lbif-1)# exit

device(config)# ip route 10.1.2.0/24 10.33.33.1
device(config)# router rip
device(config-lbif-1)# redistribute static
device(config-lbif-1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip rip v2-only
device(config-if-e10000-1/1)# ip address 10.33.33.2
device(config-if-e10000-1/1)# exit
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in the diagram above. In this example, a static route is configured between the external network (10.1.3.0/24) and the loopback interface. RIP is configured to redistribute static routes between the CE 5 router and the attached interface of PE 4.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.33.36.1/32
device(config-lbif-1)# exit

device(config)# ip route 10.1.3.0/24 10.33.36.1
```

```
device(config)# router rip
device(config-rip-router)# redistribute static
device(config-rip-router)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip rip v2-only
device(config-if-e10000-1/1)# ip address 10.33.36.2
device(config-if-e10000-1/1)# exit
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. RIP is configured on the VRF named VPN1 to exchange routes with CE 1 and to redistribute routes from across the MPLS domain.

IBGP with extended community attributes is configured between PE 1 and PE 4. The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signaled LSP named "tunnel1" to PE 1.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.1/32
device(config-lbif-1)# ip ospf area 0
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:1
device(config-vrf-vpn1)# route-target export 100:1
device(config-vrf-vpn1)# route-target import 100:2
device(config-vrf-vpn1)# exit-vrf

device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.2 remote-as 1
device(config-bgp)# neighbor 10.2.2.2 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.2 activate
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute rip
device(config-bgp-ipv4u-vrf)# exit

device(config)# router rip vrf VPN1
device(config-rip-router)# redistribute bgp
device(config-rip-router)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)#

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip address 10.33.34.1/24
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.2
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.33.3/24
device(config-if-e10000-1/1)# ip rip v2-only
device(config-if-e10000-1/1)# exit
```

PE 4 configuration

This configuration example describes what is required to operate the PE 4 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. RIP is configured on the VRF named VPN1 to exchange routes with CE 1 and to redistribute routes from across the MPLS domain.

IBGP with extended community attributes is configured between PE 4 and PE 1. The OSPF area is specified as 0, and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signaled LSP named "tunnel1" to PE 1.

```
      device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.2/24
device(config-lbif-1)# ip ospf area 0
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:2
device(config-vrf-vpn1)# route-target export 100:2
device(config-vrf-vpn1)# route-target import 100:1
device(config-vrf-vpn1)# exit-vrf

device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.1 remote-as 1
device(config-bgp)# neighbor 10.2.2.1 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.1 activate
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute rip
device(config-bgp-ipv4u-vrf)# exit

device(config)# router rip vrf VPN1
device(config-rip-router)# redistribute bgp
device(config-rip-router)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# exit

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip address 10.33.35.1/24
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.1
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.36.3/24
device(config-if-e10000-1/1)# ip rip v2-only
device(config-if-e10000-1/1)# exit
```

OSPF for route exchange

OSPF can be used to exchange routes between CE routers and PE routers. In this situation, OSPF must be enabled on the CE router with a local area and enabled on the interface that is connected to the interface of the PE that is associated with the VRF that the user wants to advertise OSPF routes on. On the PE router, the VRF must be enabled in BGP to redistribute OSPF routes, and OSPF must be enabled for the VRF and configured to redistribute routes from BGP-VPNv4. The diagram below provides an example of a network where OSPF is used to exchange routes between CE routers and PE routers.

FIGURE 88 OSPF to CE network example
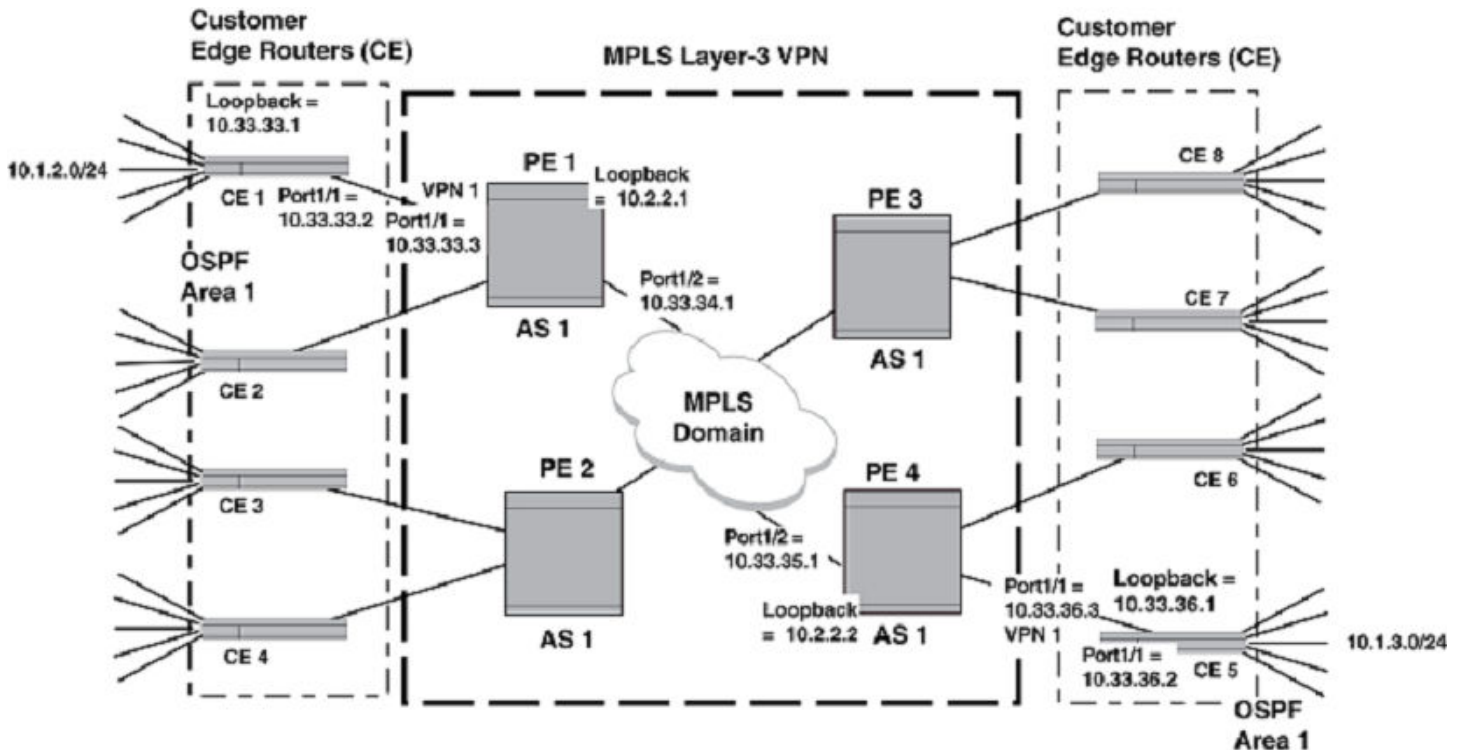
To configure OSPF to exchange routes between PE routers and CE routers, the user must perform the configuration steps listed below.

1.   Configuring OSPF on the CE router on page 529.
2.   Enabling OSPF on the CE router interface on page 530.
3.   Configuring the VRF on the PE router to redistribute OSPF routes on page 530.
4.   Configuring OSPF on the PE router to redistribute BGP-VPNv4 routes on page 530.
5.   Enabling OSPF on the PE router interface on page 530.

Configuring OSPF on the CE router

To allow OSPF route exchange between a CE router and its associated PE router, OSPF must be enabled on the CE router. To configure OSPF on the CE 1 router for local area 1 in Figure 89 on page 529 and enable it to redistribute static routes through OSPF, enter the following commands.

```
        device(config)# router ospf
device(config-ospf-router)# area 1
device(config-ospf-router)# redistribute static
```

Enabling OSPF on the CE router interface

To allow OSPF route exchange between a CE router and its associated PE router, OSPF must be enabled on the interface that connects to the VRF-enabled interface of its associated PE router. To configure OSPF on the interface of the CE 1 router in Figure 89 on page 529 that is connected to the VRF VPN1 associated interface on PE 1, enter the following commands.

```
        device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip ospf area 1
device(config-if-e10000-1/1)# ip address 10.33.33.2
```

Configuring the VRF on the PE router to redistribute OSPF routes

To allow OSPF route exchange between a specified VRF on a PE router and its associated CE router, the VRF must be enabled to redistribute OSPF routes. To enable the VRF VPN1 on PE 1 router in Figure 89 on page 529 to redistribute OSPF routes, enter the following commands.

```
        device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute ospf match internal
device(config-bgp-ipv4u-vrf)# redistribute ospf match external1
device(config-bgp-ipv4u-vrf)# redistribute ospf match external2
```

Configuring OSPF on the PE router to redistribute BGP-VPNv4 routes

To allow OSPF route exchange between a specified VRF on a PE router and its associated CE router, OSPF must be configured to redistribute BGP routes from the local AS. To enable OSPF on PE 1 in Figure 89 on page 529 and configure it to redistribute BGP-VPNv4 routes into OSPF, enter the following commands.

```
        device(config)# router ospf vrf VPN1
device(config-ospf-router)# domain-id 0.0.0.100
device(config-ospf-router)# domain-tag 1200
device(config-ospf-router)# area 1
device(config-ospf-router)# redistribute bgp
```

Enabling OSPF on the PE router interface

To allow OSPF route exchange between a PE router and its associated CE router, OSPF must be enabled on the PE's interface that is associated with the VRF and connected to the PE router. To configure OSPF on the interface of the PE 1 router that is associated with VRF VPN1 in OSPF for route exchange on page 529 to CE 1, enter the following commands.

```
        device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.33.3/24
device(config-if-e10000-1/1)# ip ospf area 1
```

OSPF to CE configuration example

In this example, the network shown in the diagram below is configured for OSPF to forward routes between the networks attached to the CE routers and the PE routers. IBGP is used to forward routes between the PE routers and an LSP tunnel is configured across the MPLS domain. The diagram below contains all of the network addresses and AS numbers required to perform this configuration. The configurations are shown for only the CE 1, CE 5, PE 1 and PE 5 routers, which demonstrates what is required to make VPN1 work. The configurations for VPN2, VPN3, and VPN 4 would essentially be the same.

FIGURE 89 OSPF to CE network example

CE 1 configuration

This configuration example describes what is required to operate the CE 1 router in the diagram above. In this example, a static route is configured between the external network (10.1.2.0/24) and the loopback interface. OSPF is configured to redistribute static routes between the CE 1 router and the attached interface of PE 1.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.33.33.1/32
device(config-lbif-1)# exit

device(config)# ip route 10.1.2.0/24 10.33.33.1
device(config)# router ospf
device(config-ospf-router)# area 1
device(config-ospf-router)# redistribute static
device(config-ospf-router)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip address 10.33.33.2
device(config-if-e10000-1/1)# ip ospf area 1
device(config-if-e10000-1/1)# exit
```

CE 5 configuration

This configuration example describes what is required to operate the CE 5 router in the diagram above. In this example, a static route is configured between the external network (10.1.3.0/24) and the loopback interface. OSPF is configured to redistribute static routes between the CE 5 router and the attached interface of PE 4.

```
       device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.33.36.1/32
device(config-lbif-1)# exit

device(config)# ip route 10.1.3.0/24 10.33.36.1
device(config)# router ospf
device(config-ospf-router)# area 1
device(config-ospf-router)# redistribution static
device(config-ospf-router)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip address 10.33.36.2
device(config-if-e10000-1/1)# ip ospf area 1
device(config-if-e10000-1/1)# exit
```

PE 1 configuration

This configuration example describes what is required to operate the PE 1 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 1. OSPF is configured on the VRF named VPN1 to exchange routes with CE 1 and to redistribute routes from across the MPLS domain.

IBGP with extended community attributes is configured between PE 1 and PE 4. The OSPF area is specified as 0 and MPLS is set up for OSPF traffic engineering. In addition, MPLS is configured with a signalled LSP named "tunnel1" to PE 1.

```
      device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.1/24
device(config-lbif-1)# ip ospf area 0
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:1
device(config-vrf-vpn1)# route-target export 100:1
device(config-vrf-vpn1)# route-target import 100:2
device(config-vrf-vpn1)# exit-vrf

device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.2 remote-as 1
device(config-bgp)# neighbor 10.2.2.2 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.2 activate
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute ospf
device(config-bgp-ipv4u-vrf)# exit

device(config)# router ospf vrf VPN1
device(config-ospf-router)# domain-id 10.0.0.100
device(config-ospf-router)# domain-tag 10.0.0.100
device(config-ospf-router)# area 1
device(config-ospf-router)# redistribute bgp
device(config-ospf-router)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# exit

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip address 10.33.34.1/24
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.2
device(config-mpls-lsp-tunnel1)# enable
device(config-if-e10000-1/2)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.33.3/24
device(config-if-e10000-1/1)# ip ospf area 1
device(config-if-e10000-1/1)# exit
```

PE 4 configuration

This configuration example describes what is required to operate the PE 4 router in the diagram above. In this example, the VRF VPN1 is created with a unique route descriptor consisting of the BGP AS number (1) and a random other number (2), and route targets are set for import and export. The VRF (VPN1) is defined on the interface that connects to CE 5. OSPF is configured on the VRF named VPN1 to exchange routes with CE 5 and to redistribute routes from across the MPLS domain.

```
        device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.2/32
device(config-lbif-1)# ip ospf area 1
device(config-lbif-1)# exit

device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 2:1
device(config-vrf-vpn1)# route-target export 100:2
device(config-vrf-vpn1)# route-target import 100:1
device(config-vrf-vpn1)# exit-vrf

device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.1 remote-as 1
device(config-bgp)# neighbor 10.2.2.1 update-source loopback 1
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.2.2.1 activate
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# redistribute ospf
device(config-bgp-ipv4u-vrf)# exit

device(config)# router ospf vrf VPN1
device(config-ospf-router)# domain-id 10.0.0.100
device(config-ospf-router)# domain-tag 10.0.0.100
device(config-ospf-router)# area 1
device(config-ospf-router)# redistribute bgp
device(config-ospf-router)# exit

device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# exit

device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# ip ospf area 0
device(config-if-e10000-1/2)# ip address 10.33.35.1/24
device(config-if-e10000-1/2)# exit

device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# traffic-engineering ospf
device(config-mpls)# mpls-interface eth 1/2
device(config-mpls)# lsp tunnel1
device(config-mpls-lsp-tunnel1)# to 10.2.2.1
device(config-mpls-lsp-tunnel1)# enable
device(config-mpls-lsp-tunnel1)# exit

device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding VPN1
device(config-if-e10000-1/1)# ip address 10.33.36.3/24
device(config-if-e10000-1/1)# ip ospf area 1
device(config-if-e10000-1/1)# exit
```

Cooperative route filtering

The Cooperative Route Filtering feature allows the user to move the filtering function of a route-target import filter to a peer. In this situation, an Outbound Route Filter (ORF) is derived from the contents of all of the **route-target import** commands of BGP configured VRFs on a PE and shared with a peer PE. This ORF is then used to exclude any routes that are blocked by that ORF from being sent by the peer PE to the PE with the **route-target import** commands from which the ORF was derived. For example, in the diagram below the routes that are admitted into VPN1 and VPN2 have route targets of 1:1 and 2:2. The user can use the cooperative route filtering feature to send an ORF that is derived from the route-target import commands on PE 1 to PE 2 to only accept these routes.

FIGURE 90 Cooperative route filtering example
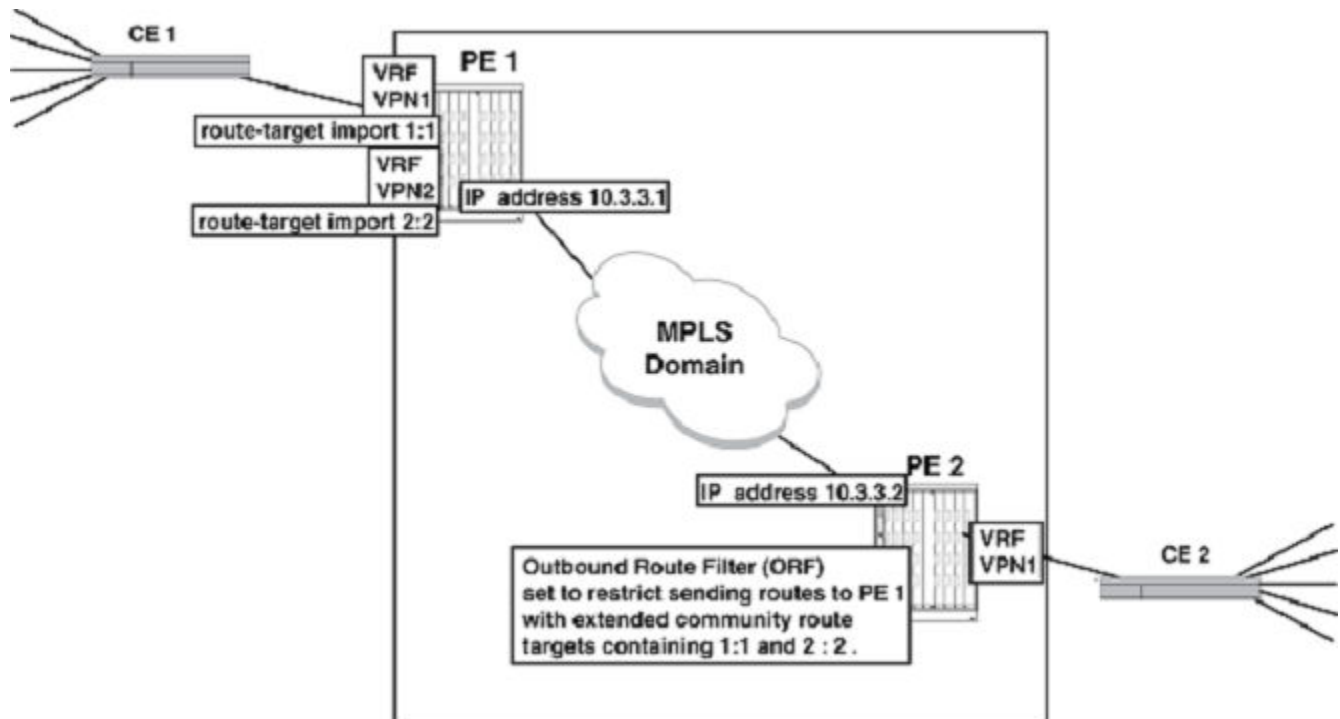
The following example shows the commands required to configure VRF VPN1 on PE 1 in the diagram above with an import route-target of 1:1 and VRF VPN2 on PE 1 with an import route-target import of 2:2.

```
        device(config)# vrf VPN1
device(config-vrf-VPN1)# route-target import 1:1
device(config-vrf-VPN1)# exit-vrf

device(config)# vrf VPN2
device(config-vrf-VPN2)# route-target import 2:2
device(config-vrf-VPN2)# exit-vrf
```

The following commands configure PE 1 to send the filter derived from the import route-target commands in VPN1 and VPN2 to PE 2.

```
        device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.3.3.2 capability orf extended-community send-vrf-filter
```

The following commands configure PE 2 to receive the filter derived from the import route-target commands in VPN1 and VPN2 on PE 1.

```
        device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-ivpnv4u)# neighbor 10.3.3.1 capability orf extended-community receive
```

Using an IP extcommunity variable with route map

In the diagram below, the VRF named "VPN1" on PE 1 is set to import routes with RT 100:14, 100:20 and 100:80. The VRF named "VPN1" on PE 4 is configured to export routes with RT 100:20 and 100:14. The VRF named "VPN2" on PE 4 is configured to export routes with RT 100:6 and 100:20. A route-map is configured from a BGP neighbor command on PE 1 to not install all routes from PE 4 with RT 100:6. This blocks all routes from VPN2 being sent to PE 1.

FIGURE 91 IP Extcommunity and route-map usage
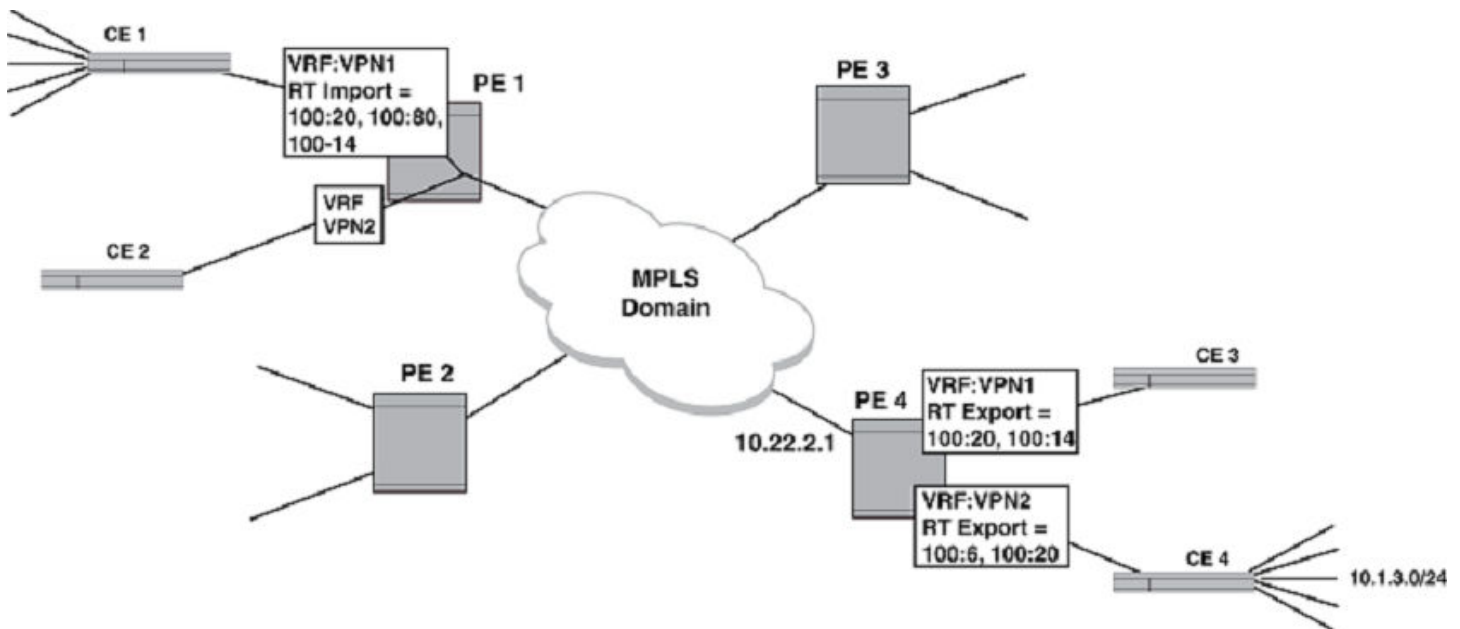
The following example shows the configuration commands required on the PE 1 router for the example shown in the diagram above. In this example, the **route-map ExcludeRoute** has an *extcommunity* value that references the extcommunity 20. The **ip extcommunity-list** command specifies that routes with RT 100:6 are to be denied. The **neighbor route-map** command exports the ExcludeRoute route-map to the BGP neighbor PE 4. Consequently, PE 4 blocks the export or route-target 100:6 to PE 1. This blocks all routes from VPN2 on PE 4 from being sent to PE 1.

```
        device(config)# router bgp
device(config-bgp)# local-as 100
device(config-bgp)# neighbor 10.22.2.1 remote-as 100
device(config-bgp)# address-family vpnv4 unicast
device(config-bgp-vpnv4u)# neighbor 10.22.2.1 activate
device(config-bgp-vpnv4u)# neighbor 10.22.2.1 route-map in ExcludeRoute
device(config-bgp-vpnv4u)# neighbor 10.22.2.1 send-community extended
device(config-bgp-vpnv4u)# exit

device(config)# route-map ExcludeRoute permit 10
device(config-routemap ExcludeRoute)# match extcommunity 20
device(config-routemap ExcludeRoute)# exit

device(config)# ip extcommunity-list 20 deny RT 100:6
device(config)# vrf VPN1
device(config-vrf-vpn1)# rd 1:1
device(config-vrf-vpn1)# route-target import 100:20
device(config-vrf-vpn1)# route-target import 100:80
device(config-vrf-vpn1)# route-target import 100:14
device(config-vrf-vpn1)# exit-vrf
```

Autonomous system number override

In the example shown in the diagram below the service providers network is in AS1 and the customer wants both of his CE routers at different sites to use AS 2. When a route is sent from CE 1 to CE 2, it contains an AS_PATH attribute containing AS 2. When CE 2 sees that the AS_PATH attribute contains its own AS number, it rejects the route.

FIGURE 92 AS number override example
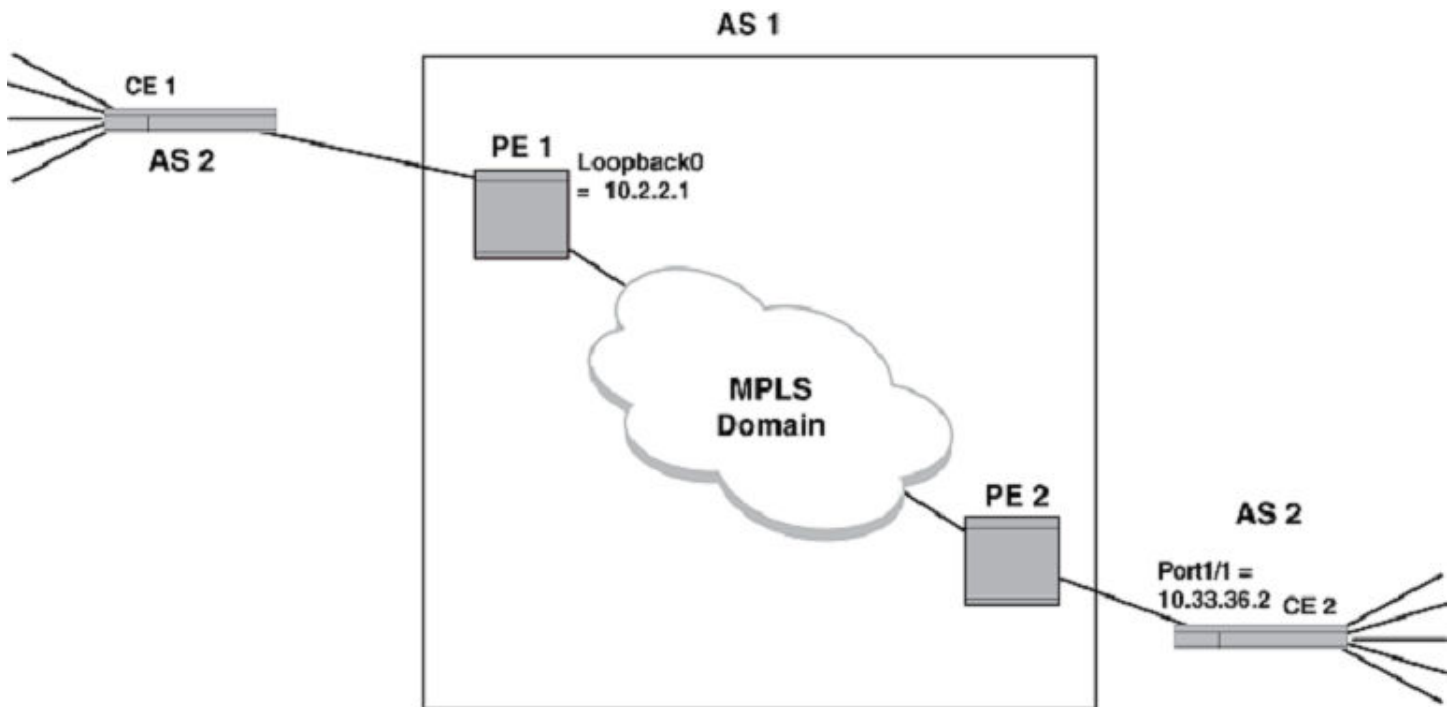
One solution to this problem is to configure PE 2 to override the AS_PATH attribute that contains AS 2. When this is enabled, the PE router determines when the AS_PATH attribute in a route intended for a neighbor CE contains the same AS number as the CE. When this is determined, the PE router substitutes its own AS number for the CE's in the AS_PATH attribute. The CE is then able to receive the route. The following addition conditions apply when this feature is in effect:

The following example describes the configuration of PE 2 required to enable Autonomous System number override for the BGP neighbor CE 2.

```
        device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# neighbor 10.2.2.1 remote-as 1
device(config-bgp)# neighbor 10.2.2.1 update-source loopback0
device(config-bgp)# address-family ipv4 unicast vrf VPN1
device(config-bgp-ipv4u-vrf)# neighbor 10.33.36.2 remote-as 2
device(config-bgp-ipv4u-vrf)# neighbor 10.33.36.2 as-override
```

Setting an LSP for each VRF on a PE

The diagram below provides an example of assigning a different LSP for each VRF on a PE. In this example, PE 1 contains two VRFs: VPN1 and VPN2. It also contains two loopback interfaces with the following IP addresses: Loopback 1 = 10.2.2.1 and Loopback 2 = 10.2.2.2. Nexthop addresses for VPN1 and VPN2 can be created separately to Loopback 1 and Loopback 2. Then, different LSPs are assigned to each of the Loopback addresses.

FIGURE 93 Support per-VRF BGP nexthop
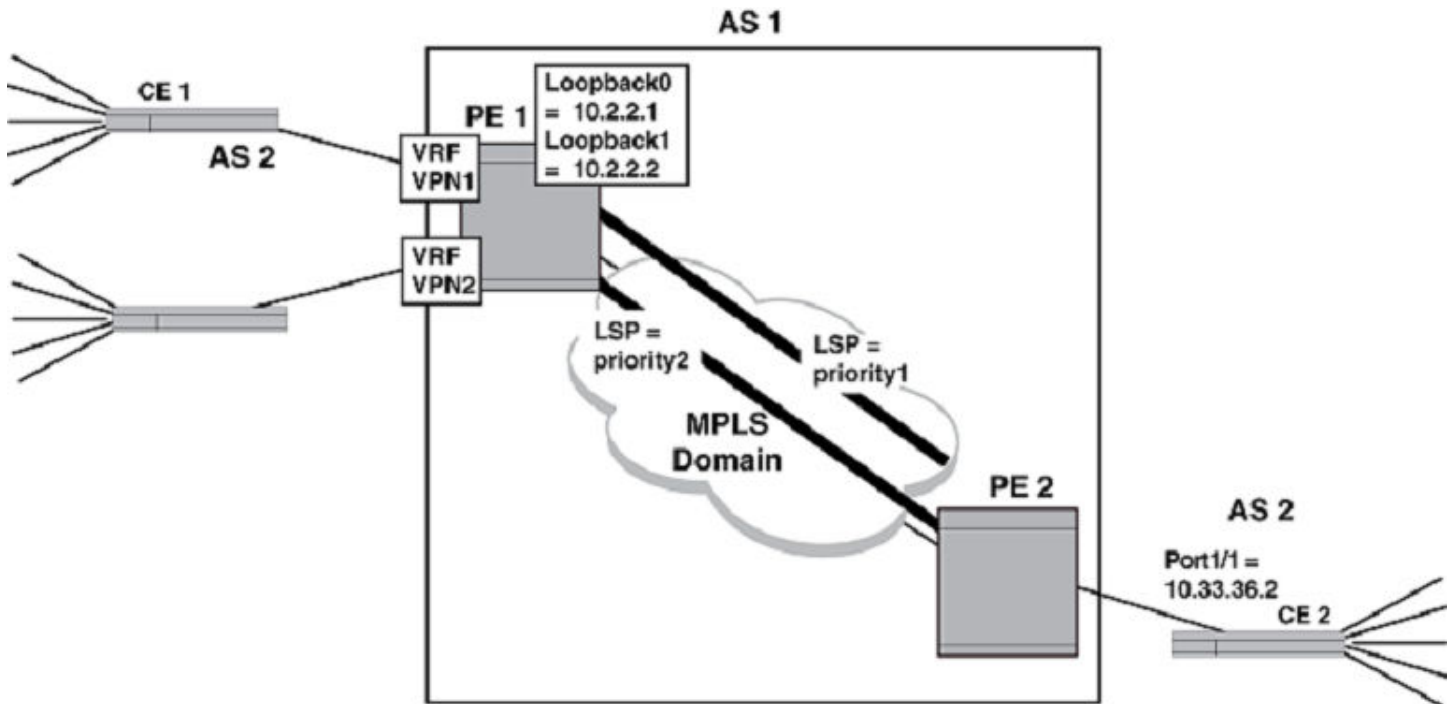
The following configuration example shows the elements in the PE 2 configuration required to make this example operate.

```
        device(config)# vrf VPN1
device(config-vrf-vpn1)# bgp next-hop loopback 1
device(config-vrf-vpn1)# exit-vrf
device(config)# vrf VPN2
device(config-vrf-vpn2)# bgp next-hop loopback 2
device(config-vrf-vpn2)# exit-vrf

device(config)# router mpls
device(config-mpls)# mpls-interface ethernet 1/1
device(config-mpls)# lsp priority1
device(config-mpls-lsp-priority1)# to 10.2.2.2
device(config-mpls-lsp-priority1)# primary-path prim-path1
device(config-mpls-lsp-priority1)# secondary-path sec-path1
device(config-mpls-lsp-priority1)# enable
device(config-mpls)# lsp priority2
device(config-mpls-lsp-priority2)# to 10.2.2.1
device(config-mpls-lsp-priority2)# primary prim-path2
device(config-mpls-lsp-priority2)# secondary sec-path2
device(config-mpls-lsp-priority2)# enable
```

OSPF sham links

In the example shown in the figure below, CE 2 and CE 3 are both in OSPF Area 1 and connect to the same service provider network through different PEs. An additional backdoor connection is configured between them over another network. OSPF recognizes the backdoor connection as an Intra-area connection and the connection through the service provider network as an Inter-network connection. Because OSPF favors Intra-area routes over Inter-network routes, most traffic between CE 2 and CE 3 travels across the backdoor link. When this is the preferred link in the network, the configuration is as it should be. However, when the user prefers traffic between the two networks to be routed across the service provider network, this configuration can causes problems.

FIGURE 94 BGP or MPLS VPN with OSPF backdoor link
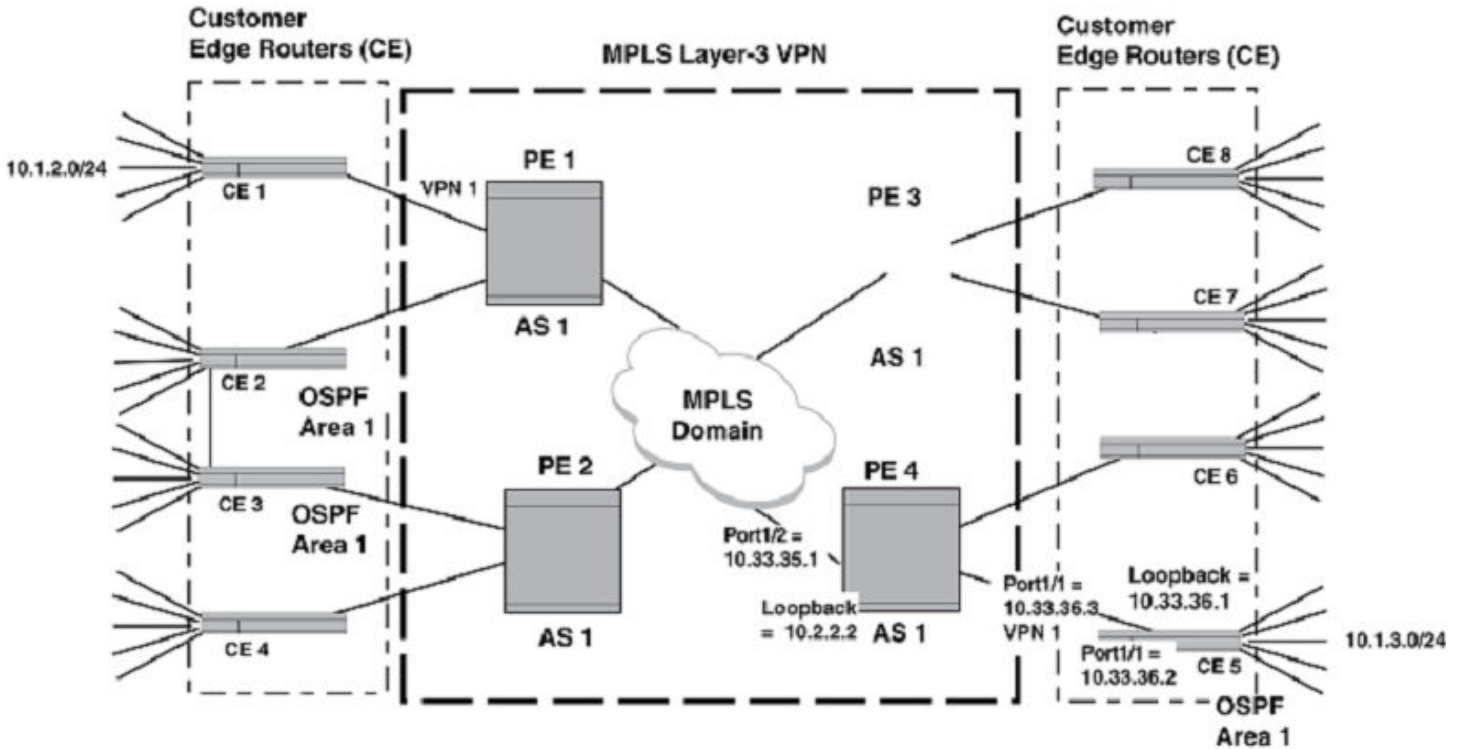
Problems can be avoided by creating a virtual intra-area OSPF link between two PEs. This virtual link is called a sham link. A sham link directs OSPF to treat the route through the service provider network as an intra-area link. A cost is assigned to the sham link to help the OSPF network determine when to route over the sham link route and when to use the backdoor link. Because this virtual link (sham-link) is an intra-area link, the OSPF areas in which each of the PEs reside must be the same.

> **NOTE**
> For sham links to work, OSPF cannot be configured on the loopback interface in the applicable area.

FIGURE 95 BGP or MPLS VPN with OSPF including Sham link and backdoor link
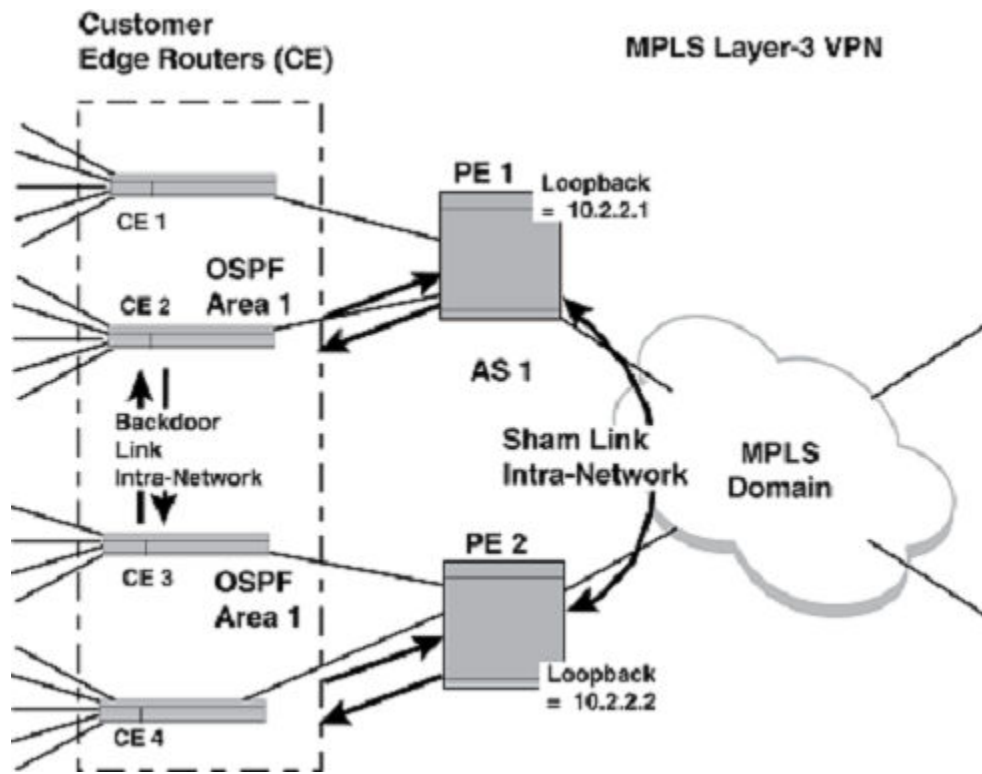
This configuration example describes the additional configuration required to create a sham link between PE 1 and PE 2 in the example shown in the figure above. In this example, the VRF VPN1 is added to the loopback interface configuration, and a sham link with a cost of 10 is created between the loopback interfaces on PE 1 and PE 2.

After this configuration is implemented, routes between CE 2 and CE 3 over the service provider network is preferred to the backdoor link that exists between these CEs.

PE 1 configuration

```
       device(config)# interface loopback 1
device(config-lbif-1)# vrf forwarding VPN1
device(config-lbif-1)# ip address 10.2.2.1/24
device(config)# vrf VPN1
device(config)# router ospf vrf VLAN1
device(config-ospf-router)# area 1 sham-link 10.2.2.1 10.2.2.2 cost 10
device(config-ospf-router)# redistribution bgp
```

PE 2 configuration

```
       device(config)# interface loopback 1
device(config-lbif-1)# vrf forwarding VPN1
device(config-lbif-1)# ip address 10.2.2.2/24
device(config)# vrf VPN1
device(config)# router ospf vrf VLAN1
device(config-ospf-router)# area 1 sham-link 10.2.2.2 10.2.2.1 cost 10
device(config-ospf-router)# redistribution bgp
```

PE 1 configuration

```
        device(config)# interface loopback 1
device(config-lbif-1)# vrf forwarding VPN1
device(config-lbif-1)# ip address 10.2.2.1/24
device(config)# vrf VPN1
device(config)# router ospf vrf VLAN1
device(config-ospf-router)# area 1 sham-link 10.2.2.1 10.2.2.2 cost 10
device(config-ospf-router)# redistribution bgp
```

PE 2 configuration

```
        device(config)# interface loopback 1
device(config-lbif-1)# vrf forwarding VPN1
device(config-lbif-1)# ip address 10.2.2.2/24
device(config)# vrf VPN1
device(config)# router ospf vrf VLAN1
device(config-ospf-router)# area 1 sham-link 10.2.2.2 10.2.2.1 cost 10
device(config-ospf-router)# redistribution bgp
```

IPv4 L3 VPN CAM optimization overview

This document reviews the functional specification for optimization of CAM programming for L3VPN CAM for IPv4. This scheme is similar to the existing CAM programming of L3VPN CAM for IPv6. After this optimization, L3VPN CAM look up for IPv4 packets is based on (VPN-ID, IP) instead of the current implementation (port, VLAN, IP). This optimization brings efficient CAM usage as it saves multiple CAM entries for same routes for the VPNs having multiple interfaces on the same PPCR.

In releases prior to NetIron R05.5.00, IP VPN CAM lookup is based on {port, VLAN, DST-IP} for ingress traffic from CE side (VRF-lite). This is an inefficient way of CAM space utilization as it requires each route to replicate for all endpoints of VRFs on the PPCR. IP VPN CAM optimization resolves this issue by programming {VPN-ID, DST-IP} as CAM lookup keys.

The VPN-ID is derived from IFL-CAM (Service CAM) at pre-lookup (lookup step prior to route lookup) stage where {port, VLAN} programs as its lookup keys. In this scheme, each route of a VRF consumes only one CAM entry on a PPCR.

Network diagram

The following figure shows the topology of the IP VPN network.

FIGURE 96 IP VPN network
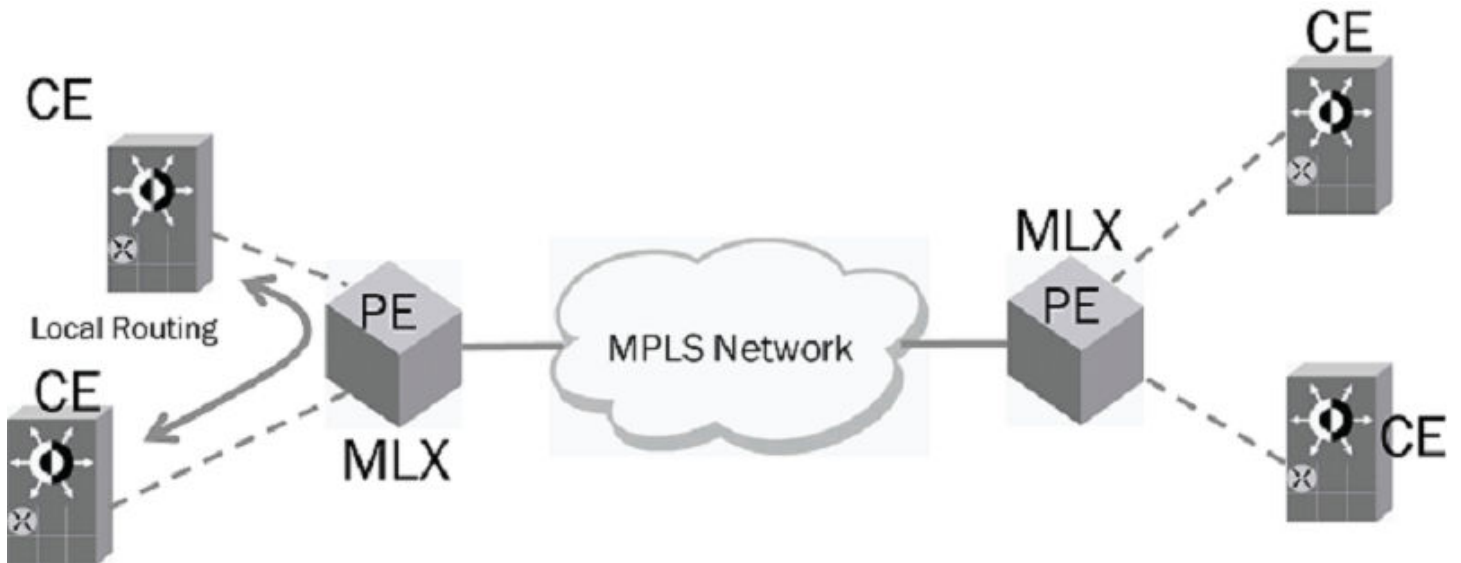
IPv4 Packet Flow for IPVPN after ingress side optimization

In VRF-lite, ingress direction packets come from CE devices to IP VPN endpoints on PE routers and enter to the MPLS domain or route back to CE (local routing). The following figure depicts the successful processing of the ingress packets after optimization.

FIGURE 97 IP VPN CAM Optimization: ingress packet processing
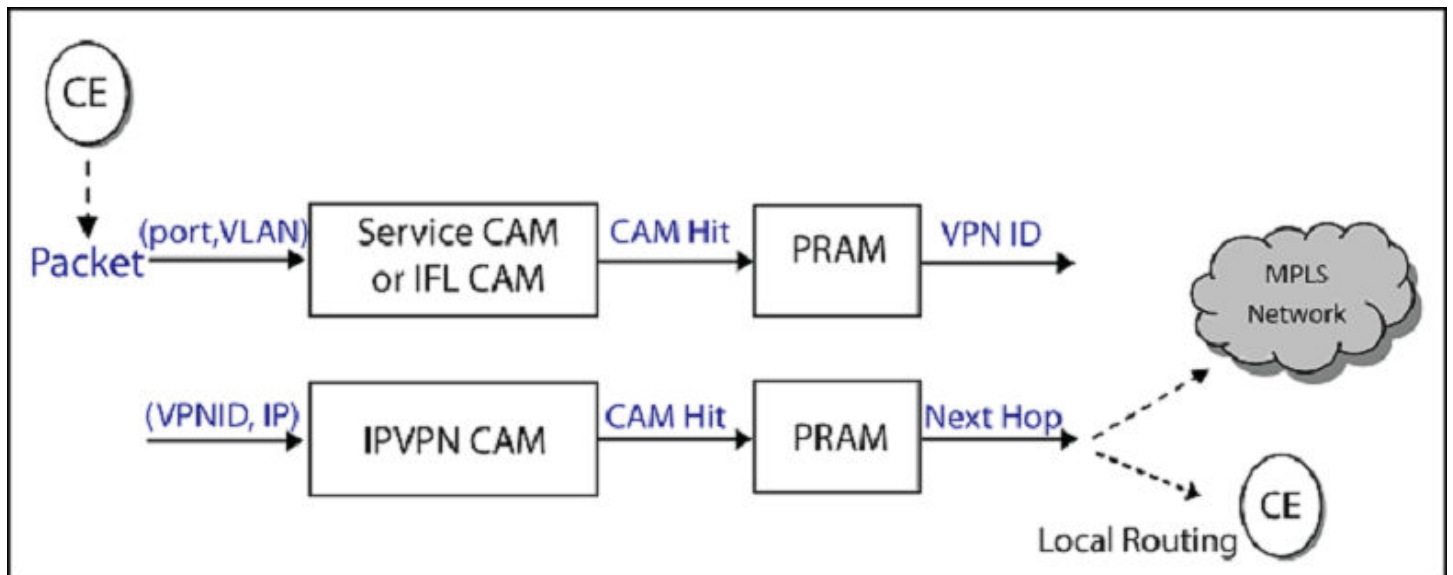
IPv4 L3 VPN CAM optimization requirements

The IPv4 L3 VPN CAM optimization feature is designed to optimize CAM usage for IPVPN routes where each route of a VRF only consumes one CAM entry on a PPCR.

From the user's point of view, there is no change in IPVPN behavior other than the option to configure an increased number of ingress routes entries.

Configuring BGP-Based Auto-Discovery for VPLS

Overview

This chapter describes how to configure the Extreme device to automatically discover *Virtual Private LAN Services (VPLS)* endpoints that are part of the same VPLS domain.

VPLS is a method for carrying Layer 2 frames between *Customer Edge (CE)* devices across a *Multi-Protocol Label Switched (MPLS)* domain. Information about VPLS, how it works, and how to manually configure it is discussed in the Overview.

With this feature, the implementation of *BGP-based auto-discovery for VPLS* (also called *VPLS auto-discovery*) eliminates the need for manual configuration of VPLS peers for every VPLS instance configured on the device. The implementation complies with the Internet draft, draft-ietf-l2vpn-signaling-08. Using the services of *Border Gateway Protocol version 4 (BGP4)* and *Label Distribution Protocol (LDP)*, VPLS auto-discovery enables a device to automatically discover other VPLS *Provider Edge (PE)* devices that are part of the same VPLS domain, and to detect and converge when other PE routers are added to or removed from the VPLS domain.

Terms introduced in this chapter

BGP-based auto-discovery for VPLS - Also called *VPLS auto-discovery*, this feature enables automatic discovery of VPLS Provider Edge (PE) devices that are part of the same VPLS domain, and the ability to detect and converge when other PE routers are added to or removed from the VPLS domain.

BGP L2VPN VPLS Routing Information Base (RIB) - Also called the *BGP L2VPN RIB*, this is the database that contains information about VPLS endpoints that are automatically discovered through VPLS auto-discovery.

L2VPN VPLS address family or *L2VPN address family* - This is the BGP-based auto-discovery mechanism used to distribute information about VPLS endpoints. Information is stored in the BGP L2VPN VPLS Routing Information Base.

Label Switch Router (LSR) ID - This is the router ID. LDP assigns the default loopback address as the router ID. Because VPLS auto-discovery uses the services of LDP, a valid loopback address must be configured on the Extreme device before VPLS auto-discovery can be enabled.

Route Distinguisher (RD) - The address qualifier used within a single *Internet Service Provider's (ISPs) Multi-Protocol Label Switching (MPLS)* network. The qualifier is used to distinguish the distinct *Virtual Private Network (VPN)* routes of separate customers who connect to the service provider.

Route Target (RT) Extended Community – Defines the import and export policies applied to a VPLS instance. Each VPLS instance is associated with one or more route target extended communities.

Subsequent Address Family Identifier (SAFI) – An ID number that provides additional information about the NLRI type for a given attribute.

VPLS Virtual Circuit Identifier (VPLS VCID) or *VPLS ID* – Identifies the endpoints of a VPLS instance. All Provider Edge (PE) routers that are part of the same VPLS instance have the same VPLS VCID.

How BGP-based auto-discovery for VPLS works

The devices use the services of LDP and BGP4 to discover automatically VPLS endpoints that are part of the same VPLS domain. To enable the Extreme device to distribute information about VPLS endpoints, the user must configure a L2VPN VPLS address family, activate BGP peering on the L2VPN VPLS address family, then enable BGP-based auto-discovery for VPLS. When BGP L2VPN VPLS update messages are exchanged between PE routers, the device can start discovering VPLS peer addresses.

For every VPLS instance on which BGP-based auto-discovery is enabled, the device automatically generates a *Route Distinguisher (RD)* value based on the BGP *Autonomous System (AS)* number and the VPLS *Virtual Circuit Identifier (VCID)* for PE routers. The RD Is an address qualifier used by the PE router to distinguish VPN routes of separate customers. Also, when not manually configured, the device automatically generates import and export route targets that define the policies that each VPLS instance uses. A local VPLS endpoint *Network Layer Reachability Information (NLRI)* and import route-target tree are also created and sent to BGP peers with the L2VPN VPLS capability.

When the device receives information about a VPLS endpoint, it checks when its extended community matches any locally-configured VPLS import route targets. When a match is found, information about the VPLS endpoint is stored in the BGP L2VPN routing table and a notification is sent to VPLS for peering information. Once VPLS receives the information, it creates a VPLS peer and starts a peering session.

When VPLS auto-discovery is disabled for a VPLS instance, the system removes all auto-discovered peers for the VPLS instance from the configuration. It then removes the route (local VPLS endpoint address) from the BGP L2VPN route table and sends a "withdrawn" message to VPLS peers, prompting them to remove the route and to disable VPLS auto-discovery. Finally, the system updates the route target tree and sends a route refresh message for the L2VPN VPLS address family.

About the L2VPN VPLS address family

The *L2VPN VPLS address family* is an integral part of BGP-based auto-discovery for VPLS, in that it is the mechanism used by BGP4 to distribute information about VPLS endpoints. The L2VPN address family is configured at the BGP configuration level of the CLI and supports the VPLS *Subsequent Address Family Identifier (SAFI)*, an address qualifier that provides additional information about the *Network Layer Reachability Information (NLRI)* type for a given attribute.

BGP4 uses the L2VPN address family to build the BGP L2VPN *Routing information Base (RIB)*. The L2VPN database updates each time a Layer 2 *Virtual Forwarding Instance (VFI)* is configured.

Information about configuring the L2VPN Address Family is in the section

Feature limitations and configuration notes

Consider the following feature limitations and configuration notes:

- VPLS must not be used when FDP is enabled.
- VPLS auto-discovery and manual configuration of VPLS peers are supported together on the same device. However, they are not supported together on the same VPLS instance.
- VPLS auto-discovery is not compatible in multi vendor environment. It must only be used among MLX Series series and XMR Series devices.

Scalability

The following section describes the scalability:

- The maximum number of BGP4 peers that can support the L2VPN VPLS address family is equal to the maximum number of BGP4 peers supported on the device.
- The maximum number of VPLS instances that can support BGP-based auto-discovery for VPLS is equal to the maximum number of VPLS instances supported on the device.
- The maximum number of BGP-based auto-discovered peers supported per VPLS instance is equal to the maximum number of unique VPLS or VLL peers or number of VPLS peers supported on the device. When the system exceeds the default or manually-configured maximum number of VPLS peers supported on the device, any new peering for VPLS auto-discovery is rejected.

Configuring BGP-based auto-discovery for VPLS

It is recommended that the user performs the configuration tasks in the order listed in Table 80. Performing the tasks in the recommended sequence minimizes CPU consumption and route flapping. Except where noted as "optional", the configuration tasks in the table are required for VPLS auto-discovery.

TABLE 80 Configuration tasks for VPLS auto-discovery

Configuration task		See...
1.	Configure a loopback address	Configuring a loopback interface on page 548
1.	Enable BGP4 and assign a local *Autonomous System (AS)* number	Configuring BGP4 to support VPLS auto-discovery on page 549
1.	Enable MPLS and configure LDP	To configure MPLS, refer to the Configuring BGP-Based Auto-Discovery for VPLS on page 545. To configure LDP, refer to the Configuring BGP-Based Auto-Discovery for VPLS on page 545.
1.	Configure VPLS: • Create a VPLS instance • Define the route target (optional) • Enable load balancing (optional)	Configuring VPLS to support auto-discovery on page 550
1.	Enable VPLS auto-discovery	Enabling VPLS auto-discovery on page 554

TABLE 80 Configuration tasks for VPLS auto-discovery (continued)

Configuration task		See...
1.	Configure the L2VPN VPLS address family and activate BGP4 peering	Configuring the L2VPN VPLS address family and activating the BGP4 peering session on page 554

After performing the configuration steps listed in Table 80, the user can observe the L2VPN VPLS address family routes, neighbor summary, and VPLS auto-discovery peering. Refer to Displaying VPLS auto-discovery information on page 558.

> **NOTE**
> This behavior does not apply to CES 2000 Series or CER 2000 Series devices when the IPv6 packet has the following format IPv6 header + IPv6 Hop-by-Hop Extension Header + IPv6 Routing Header (with type 0).

Configuring a loopback interface

The user must configure a loopback address on the Extreme device before enabling VPLS auto-discovery.

This section contains the following topics:

- About loopback interfaces and the router ID on page 548
- Changes that occur when a loopback interface is deletes on page 548
- Adding a loopback interface on page 549
- Viewing the loopback Interface on page 549

About loopback interfaces and the router ID

In most configurations, an Extreme device has multiple IP addresses, usually configured on different interfaces. As a result, an Extreme device's identity to other devices varies depending on the interface to which the other device is attached. BGP4 identifies an Extreme device by just one of the IP addresses configured on the device, regardless of the interfaces that connect the devices. This IP address is the *router ID* also known as the *Label Switched Router (LSR)* ID.

LDP uses the default loopback address as the router ID. Since VPLS auto-discovery uses the services of LDP, a valid loopback address must be configured on the Extreme device before VPLS auto-discovery can be enabled. When a loopback address is not configured, the LDP router ID is NULL and VPLS auto-discovery does not function.

When there are several loopback addresses configured on the device, the default loopback address is the IP address configured on the lowest-numbered loopback interface on the Extreme device. For example, when the user configures loopback interfaces 1, 2, and three as follows, the default router ID is 10.9.9.9/24.

```
Loopback interface 1, 10.9.9.9/24
Loopback interface 2, 10.4.4.4/24
Loopback interface 3, 10.1.1.1/24
```

Changes that occur when a loopback interface is deletes

When a loopback interface deletes while VPLS auto-discovery is enabled, and more than one loopback interface is configured on the device, the Extreme device uses the IP address configured on the next lowest numbered loopback interface as the router ID. For example, when loopback interface one deletes, the Extreme device uses loopback interface 2. Thus, the successive router ID is 10.4.4.4/24. The system removes all existing VPLS routes for 10.9.9.9/24 and obtains new routes for 10.4.4.4/24.

```
Loopback interface 1, 10.9.9.9/24
Loopback interface 2, 10.4.4.4/24
```

When a loopback interface is deleted from the configuration while VPLS auto-discovery is enabled, and there are no other valid loopback interfaces, the system disables LDP and VPLS auto-discovery.

Adding a loopback interface

To add a loopback interface, enter commands similar to the following:

```
device(config)# int loopback 1
device(config-lbif-1)# ip address 10.1.1.4/24
```

Syntax: [no] interface loopback num

Use the [**no**] form of the command to delete a loopback interface. Also refer to Changes that occur when a loopback interface is deletes on page 548in the following section.

The *num* value can be a number from 1 - 64.

Viewing the loopback Interface

Use the **show mpls ldp** command to view the loopback interface and router ID in use on the Extreme device. Refer to Displaying information about LDP on page 570.

Configuring BGP4 to support VPLS auto-discovery

BGP4 must be enabled on the device and a local *Autonomous System (AS)* number must be assigned before VPLS auto-discovery can be enabled.

This section includes configuration details for the following BGP-related tasks:

- How to enable BGP4 and assign the local AS number
- How to change or clear the local AS number when VPLS auto-discovery is enabled
- How to disable BGP4 when VPLS auto-discovery is enabled

    NOTE
    This section provides minimal information about configuring BGP4 neighbors, peer groups, and other essential BGP-related configuration tasks, because its focus is to provide information about configuring BGP to support VPLS auto-discovery.

Enabling BGP4 and assigning the local AS number

In a VPLS configuration, all PEs in the same VPLS domain must be configured with the same AS number. To assign a local AS number to the Extreme device, enter commands such as the following.

```
device(config)# router bgp
device(config-bgp)# local-as 10
device(config-bgp)# neighbor 10.1.1.1 remote-as 10
```

For additional information regarding this command, see *NetIron Command Line Interface (CLI) Reference Guide*.

Changing or clearing the local AS number when VPLS auto-discovery is enabled

When VPLS auto-discovery is enabled on the device and the user wants to clear or change the BGP local AS number, the user must first disable VPLS auto-discovery, then clear the local AS number. When the user attempts to clear or change the local AS number while VPLS auto-discovery is enabled, the console displays the following message:

```
device(config-bgp)# no router bgp
Error:  VPLS instances with BGP auto-discovery exist, remove auto-discovery configuration first!
```

To clear the BGP local AS number when VPLS auto-discovery is enabled, enter commands similar to the following.

```
device(config)# router mpls
device(config-mpls)# vpls c1 10
device(config-mpls-vpls-c1)# no auto-discovery
device(config-mpls-vpls-c1)# router bgp
device(config-bgp)# no local-as 10
BGP is no longer operational
```

Syntax: [no] auto-discovery

Syntax: [no] local-as num

Disabling BGP4 when VPLS auto-discovery is enabled

When VPLS auto-discovery is enabled on the device and the user wants to disable BGP4, first disable VPLS auto-discovery, then disable BGP4 at the VPLS instance level of the CLI. When the user attempts to disable BGP4 while VPLS auto-discovery is enabled, the console displays the following message:

```
device(config-bgp)# no router bgp
Error:  There are VPLS instances with BGP auto-discovery enabled, disable auto-discovery first!
```

> **NOTE**
> When VPLS auto-discovery is not enabled on the device, the use can disable BGP simply by entering the CLI command **no router bgp** at the global CONFIG level of the CLI.

To disable BGP4 when VPLS auto-discovery is enabled, enter commands such as the following.

```
device(config)# router mpls
device(config-mpls)# vpls c1 10
device(config-mpls-vpls-c1)# no auto-discovery
device(config-mpls-vpls-c1)# no router bgp
router bgp mode now disabled.  All bgp config data will be lost when writing to flash!!
```

Syntax: [no] auto-discovery

Syntax: [no] router bgp

> **NOTE**
> When BGP is disabled, the system also removes the BGP local AS number from the configuration.

Configuring VPLS to support auto-discovery

This section describes how to configure VPLS to support BGP-based auto-discovery. It includes the following configuration details:

- Creating a VPLS instance on page 551
- Defining the route target for a VPLS instance (optional) on page 551
- Enabling and disabling load balancing for a VPLS instance (optional) on page 552

This section provides minimal information about configuring VPLS and other related configuration tasks, because its focus is to provide information about configuring VPLS to support BGP-based auto-discovery.

Creating a VPLS instance

To create a VPLS instance, enter VPLS configuration statements on two or more PE routers.

On the PE routers, enter commands such as the following:

```
device(config)# router mpls
device(config-mpls)# mpls-interface ethernet 1/1
device(config-mpls)# vpls CustomerA 10
device(config-mpls-vpls-CustomerA)#
```

On the VPLS peers (when they are Extreme devices), enter commands similar to the following:

```
device(config)# router mpls
device(config-mpls)# mpls-interface ethernet 6/1
device(config-mpls)# vpls CustomerA 10
device(config-mpls-vpls-CustomerA)#
In the above configurations, the endpoints of the VPLS instance are associated by having the same Virtual
Circuit Identifier (VCID) of 10 on each PE router.
```

Syntax: [no] router mpls

Syntax: [no] mpls-interface ethernet [*slot_num / portnum*]

Syntax: [no] vpls name vpls-vcid

The **router mpls** command enables MPLS. Enter the [**no**] form of the command to disable it.

The **mpls-interface ethernet** command specifies the interface on which to create the VPLS instance.

The **vpls** *name* parameter specifies the VPLS instance name. The name can be up to 64 alphanumeric characters.

The *vpls-vcid* parameter specifies the VCID for the BGP L2VPNVPLS instance. The endpoints of a VPLS instance are associated by having the same VCID on each PE router. Enter a number in the range 1 - 4294967294.

Defining the route target for a VPLS instance (optional)

NOTE
When the user decides to manually define a route target, it is recommended that the user do so before enabling VPLS auto-discovery.

The **route target** extended community for VPLS auto-discovery defines the import and export policies that a VPLS instance uses. The export route target sets an extended community attribute number that is appended to all routes that are exported from the VPLS instance. The import route target value sets a filter that determines the routes that are accepted into the VPLS instance. Any route with a value in its import route target contained in its extended attributes field matching the value in the VPLS instance's import route target are accepted. Otherwise the route is rejected.

In a configuration with VPLS auto-discovery, configuring a route target is optional. When the user does not manually configure one, the system automatically generates the import and export route target for each VPLS instance configured on the Extreme device when VPLS auto-discovery is enabled. A manually-configured route target takes precedence over one that is automatically generated by VPLS auto-discovery. When all manually-configured route targets are removed from a VPLS instance while VPLS auto-discovery is enabled, the system automatically generates a new route target for the VPLS instance.

The Extreme device supports up to 16 unique import and export route targets per VPLS instance. When the user attempts to configure more than 16, the system displays the following error message:

```
Error: Maximum number of Import RT for a VPLS instance is 16!
Error: Maximum number of Export RT for a VPLS instance is 16!
```

To define an import route target of 3:6 and an export route target of 3:8 for a VPLS instance, enter commands similar to the following:

```
device(config)# router mpls
device(config-mpls)# vpls c1
device(config-mpls-vpls-c1)# route-target import 3:6
device(config-mpls-vpls-c1)# route-target export 3:8
```

Syntax: [no] route-target [both | import | export] [*ASN:num* | *IP-address:num*]

The **both** parameter specifies both import and export values apply to the specified route target for the VPLS instance where this command is applied. This is the default state and applies when no specific value for this parameter is set.

The **import** parameter specifies that routes with route-target extended community attributes matching the specified route-target can be imported into the VPLS instance where this command is applied.

The **export** parameter specifies the route-target extended community attributes that are attached to routes exported from the specified VPLS instance.

The *ASN:num* parameter identifies the route as an ASN relative. This number is the local ASN number followed by a colon (**:**) and a unique arbitrary number.

The *IP-address:num* parameter identifies the route as an IP-address relative. This number is the local IP address followed by a colon (**:**) and a unique arbitrary number.

Viewing the route target for a VPLS instance

Use the **show mpls vpls name** command to view the route targets for a VPLS instance. Refer to *NetIron Command Line Interface (CLI) Reference Guide*.

Enabling and disabling load balancing for a VPLS instance (optional)

This section describes how to enable and disable load balancing for a VPLS instance on which VPLS auto-discovery is enabled. When load balancing is enabled, the Extreme device automatically load balances traffic to all auto-discovered peers.

The user can configure a VPLS instance to load balance known unicast traffic sent to auto-discovered VPLS peers across multiple tunnel LSPs. The CLI commands for enabling and disabling load balancing differ depending on whether VPLS auto-discovery is enabled on the VPLS instance. Follow the appropriate procedures in this section.

> **NOTE**
> The Extreme device load balances traffic for auto-discovered VPLS peers, the same as for manually-created VPLS peers.

Enabling load balancing when VPLS auto-discovery is disabled

To enable VPLS auto-discovery and load balancing of traffic sent to auto-discovered VPLS peers, enter commands such as the following:

NOTE
Before enabling VPLS auto-discovery, make sure the user has completed the configuration tasks listed in Configuring BGP-based auto-discovery for VPLS on page 547.

```
device(config)# router mpls
device(config-mpls)# vpls c1 10
device(config-mpls-vpls-c1)# auto-discovery load-balance
```

Syntax: [no] auto-discovery load-balance

To disable load balancing, refer to Disabling load balancing on page 553.

Enabling load balancing when VPLS auto-discovery is enabled

When VPLS auto-discovery is enabled for a VPLS instance and the user wishes to enable load balancing, the user must first disable VPLS auto-discovery, then re-enable it with the **load-balancing** option. When the user attempts to enable load balancing when VPLS auto-discovery is enabled, the console displays the following message:

```
device(config-mpls-vpls-c1)# auto-discovery load-balance
Error:  Please disable auto-discovery before make change!
```

To enable load balancing for a VPLS instance that has VPLS auto-discovery enabled, enter commands similar to the following:

```
device(config)# router mpls
device(config-mpls)# vpls c1 10
device(config-mpls-vpls-c1)# no auto-discovery
device(config-mpls-vpls-c1)# auto-discovery load-balance
```

The above commands disable VPLS auto-discovery for VPLS instance "c1", then re-enable VPLS auto-discovery with the **load-balance** option.

Syntax: [no] auto-discovery

Syntax: [no] auto-discovery load-balance

Disabling load balancing

To disable load balancing when VPLS auto-discovery is enabled on the device, first disable VPLS auto-discovery, then re-enable it without the **load-balancing** option.

```
device(config)# router mpls
device(config-mpls)# vpls c1 10
device(config-mpls-vpls-c1)# no auto-discovery load-balance
device(config-mpls-vpls-c1)# auto-discovery
```

Syntax: [no] auto discovery load-balance

Syntax: [no] auto-discovery

Viewing the load balancing configuration

Use the **show mpls vpls name** command to view when VPLS traffic to the peer is load balanced across tunnel LSPs, and to the tunnel LSPs used to reach the peer. Refer to *NetIron Command Line Interface (CLI) Reference Guide.*

Enabling VPLS auto-discovery

> **NOTE**
> Before enabling VPLS auto-discovery, make sure the user has completed the configuration tasks listed in Configuring BGP-based auto-discovery for VPLS on page 547.

To enable auto-discovery for a VPLS instance, enter commands similar to the following:

```
device(config)# router mpls
device(config-mpls)# vpls c1
device(config-mpls-vpls-c1)# auto-discovery
```

These commands enable MPLS, then change the CLI configuration level from the global MPLS level to the configuration level for the VPLS instance "c1". The **auto-discovery** command enables auto-discovery for this VPLS instance.

Syntax: [no] auto-discovery

Use the [**no**] form of the command to disable VPLS auto-discovery.

Configuration notes

Consider the following configuration notes while enabling VPLS auto-discovery:

- When the user attempts to enable VPLS auto-discovery without first adding a loopback interface, the following error message displays on the console.

```
device(config-mpls-vpls-c2)# auto-discovery
Error: Please configure a loopback address for LDP first!
```

To add a loopback interface, follow the configuration instructions in Configuring a loopback interface on page 548.

- When the user attempts to enable VPLS auto-discovery without first configuring the BGP AS number, the following error message displays on the console.

```
device(config-mpls-vpls-c2)# auto-discovery
Error: Cannot configure auto-discovery before configuring BGP-AS number!
```

To configure the BGP AS number, follow the configuration instructions in Configuring BGP4 to support VPLS auto-discovery on page 549.

Configuring the L2VPN VPLS address family and activating the BGP4 peering session

This section describes how to configure the L2VPN VPLS address family and activate BGP4 peering. More information about the L2VPN VPLS address family is in the section About the L2VPN VPLS address family on page 546.

> **NOTE**
> It is recommended that the user activates peering on the L2VPN VPLS address family after performing steps 1 - 5 in Configuring BGP-based auto-discovery for VPLS on page 547. Otherwise, the user needs to clear the entire peering session.

To configure the L2VPN VPLS address family, enter commands such as the following.

```
device(config)# router bgp
device(config-bgp)# address-family l2vpn vpls
device(config-bgp-l2vpn-vpls)# neighbor 10.10.1.1 activate
device(config-bgp-l2vpn-vpls)# exit
```

Syntax: [no] address-family l2vpn vpls

Syntax: [no] neighbor *IPv4-address* [| **peer group** *name* **activate** | **remote-as** | **send-community extended**]

The **activate** option enables the exchange and updating of routes within the L2VPN VPLS address family.

The **send-community extended** command enables the sending of extended community attributes to this neighbor.

Clearing the BGP L2VPN route table

The user can clear routes from the BGP L2VPN route table with or without resetting the BGP session. Use the appropriate commands in this section.

Clearing the BGP L2VPN route table and resetting BGP

> NOTE
> This section describes how to clear routes from the BGP L2VPN route table and reset the BGP session. When the user does not want to reset the BGP session while clearing routes, refer to Clearing the BGP L2VPN route table without resetting the BGP session on page 555.

The user can clear routes from the BGP L2VPN route table that were exchanged by the Extreme device and:

- All BGP4 neighbors
- A specific neighbor
- A specific peer group

To clear and reset all BGP4 routes from the BGP L2VPN route table, enter a command similar to the following:

```
device# clear ip bgp l2vpn vpls neighbor all
```

To clear and reset BGP4 routes exchanged by the Extreme device and a *specific neighbor*, enter a command similar to the following.

```
device# clear ip bgp l2vpn vpls neighbor 10.10.10.1
```

To clear and reset BGP4 routes exchanged by the Extreme device and a *specific peer group*, enter a command similar to the following:

```
device# clear ip bgp l2vpn vpls neighbor peergroup1
```

Syntax: clear ip bgp l2vpn vpls neighbor [**ip-addr** | **peer-group-name**]

The *peer-group-name* | *as-num* specifies the neighbor.

The *ip-addr* parameter specifies a neighbor by its IP interface with the Extreme device.

The *peer-group-name* specifies all neighbors in a specific peer group.

Clearing the BGP L2VPN route table without resetting the BGP session

When clearing all BGP4 routes from the BGP L2VPN route table, the user can place policy changes into effect without resetting the BGP session. To do so, enter a command such as the following.

```
device(config-bgp)# clear ip bgp l2vpn vpls neighbor all soft in
```

This command updates the inbound routes in the BGP L2VPN route table by comparing the route policies against the route updates that the Extreme device has stored. The command does not request additional updates from the neighbor or otherwise affect the session with the neighbor.

Syntax: clear ip bgp l2vpn vpls neighbor all [*ip-addr* | *peer-group-name*] **soft** [**in** | **out**]

The **soft** parameter performs a soft reset of the neighbor session, which does not affect the session with the neighbor.

The **in** parameter updates inbound routes.

The **out** parameter updates outbound routes.

> **NOTE**
> When the user does not specify "in", the command applies to both inbound and outbound updates.

Example configuration

The following shows a typical VPLS auto-discovery configuration.

device1 configuration

The following commands are entered on device1.

```
device1(config)# int loopback 1
device1(config-lbif-1)# ip address 10.1.1.1/24
device1(config-lbif-1)# exit
device1(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
device1(config-bgp)# local-as 10
device1(config-bgp)# neighbor 10.1.1.2 remote-as 10
device1(config-bgp)# exit
device1(config)# router mpls
device1(config-mpls)# mpls-interface ethernet 1/1
device1(config-mpls)# vpls C1 10
device1(config-mpls-vpls-C1)# auto-discovery
device1(config-mpls)# exit
device1(config-mpls)# vpls C2 20
device1(config-mpls-vpls-C2)# auto-discovery
device1(config-mpls-vpls-C2)# exit
device1(config-mpls)# exit
device1(config)# router bgp
device1(config-bgp)# address-family l2vpn vpls
device1(config-bgp-l2vpn-vpls)# neighbor 10.1.1.2 activate
device1(config-bgp-l2vpn-vpls)# exit-address-family
device1(config-bgp)# exit
device1(config)#
```

device2 configuration

The following commands are entered on device2, a peer of device1.

```
device2(config)# int loopback 1
device2(config-lbif-1)# ip address 10.1.1.2/24
device2(config-lbif-1)# exit
device2(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
device2(config-bgp)# local-as 10
device2(config-bgp)# neighbor 10.1.1.1 remote-as 10
device2(config-bgp)# exit
device2(config)# router mpls
device2(config-mpls)# mpls-interface ethernet 1/1
device2(config-mpls)# vpls C1 10
device2(config-mpls-vpls-C1)# auto-discovery
device2(config-mpls)# exit
device2(config-mpls)# vpls C2 20
device2(config-mpls-vpls-C2)# auto-discovery
device2(config-mpls-vpls-C2)# exit
device2(config-mpls)# exit
```

```
device2(config)# router bgp
device2(config-bgp)# address-family l2vpn vpls
device2(config-bgp-l2vpn-vpls)# neighbor 10.1.1.1 activate
device2(config-bgp-l2vpn-vpls)# exit-address-family
device2(config-bgp)# exit
device2(config)#
```

After applying the above commands, the user can use various show commands to display information about the VPLS auto-discovery configuration. In the show command examples that follow, the lines in bold type indicate the information specific to the VPLS auto-discovery configuration.

> **NOTE**
> The **show mpls vpls name** command has changed. The total VC labels allocated field is no longer displayed in the output of the **show mpls vpls name** command.

```
device1# show ip bgp nei 10.1.1.2
1    IP Address: 10.1.1.2, AS: 10 (IBGP), RouterID: 10.2.2.2, VRF: default-vrf
     State: ESTABLISHED, Time: 0h1m5s, KeepAliveTime: 60, HoldTime: 180
         KeepAliveTimer Expire in 34 seconds, HoldTimer Expire in 175 seconds
     Minimal Route Advertisement Interval: 0 seconds
         RefreshCapability: Received
     Messages:      Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1         1        2        0          0
       Received: 1         4        2        0          0
     Last Update Time: NLRI    Withdraw    NLRI        Withdraw
                   Tx: ---     ---         Rx: 0h1m5s  ---
     Last Connection Reset Reason: Hold Timer Expired
     Notification Sent:     Unspecified
     Notification Received: Unspecified
     Neighbor NLRI Negotiation:
       Peer Negotiated IPV4  unicast  capability
       Peer Negotiated VPNv4 unicast  capability
       Peer Negotiated L2VPN VPLS address family
       Peer configured for IPV4 unicast  Routes
       Peer configured for VPNv4 unicast  Routes
       Peer configured for L2VPN VPLS address family
     Neighbor Capability Negotiation:
     As-path attribute count: 3

device1# show ip bgp l2vpn vpls sum
  BGP4 Summary
  Router ID: 10.1.1.1   Local AS Number: 10
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 1, UP: 1
  Number of Routes Installed: 4, Uses 344 bytes
  Number of Routes Advertising to All Neighbors: 2, Uses 88 bytes
  Number of Attribute Entries Installed: 4, Uses 376 bytes
  Neighbor Address  AS#    State   Time      Rt:Accepted  Filtered  Sent   ToSend
  10.1.1.2          10     ESTAB   0h 7m21s  2            0         2      0

device1# show ip bgp l2vpn vpls
Total number of BGP L2VPN VPLS Routes: 4
Status codes: s suppressed, d damped, h history, * valid, > best, i  internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network       Next Hop    Metric    LocPrf    Weight    Path
Route Distinguisher: 10:10
*>  10.1.1.1/32   0.0.0.0     0         100       65535     i    << local VPLS endpoint for C1
*i  10.2.2.2/32   0.0.0.0     0         100       0         i        <<remote VPLS endpoint for C1
Route Distinguisher: 10:20
*>  10.1.1.1/32   0.0.0.0     0         100       65535     i
*i  10.2.2.2/32   0.0.0.0     0         100       0         i

device1# show mpls vpls name c1
VPLS c1, Id 10, Max mac entries: 8192
 Total vlans: 0, Tagged ports: 0 (0 Up), Untagged ports 0 (0 Up)
 Total VPLS peers: 1 (0 Operational)
 auto-discovery enabled, RD 10:10
```

```
 export RT    10:10
 import RT    10:10
 Peer address: 10.2.2.2 (auto-discovered)
, State: Wait for functional local ports
  Tnnl in use: None
  LDP session: Up, Local VC lbl: 983040, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0
 CPU-Protection: OFF
 Local Switching: Enabled

device1# show mpls vpls name c2
VPLS c2, Id 20, Max mac entries: 8192
 Total vlans: 0, Tagged ports: 0 (0 Up), Untagged ports 0 (0 Up)
 Total VPLS peers: 1 (0 Operational)
 auto-discovery enabled, RD 10:20
 export RT    10:20
 import RT    10:20
 Peer address: 10.2.2.2 (auto-discovered)
, State: Wait for functional local ports
  Tnnl in use: None
  LDP session: Up, Local VC lbl: 983072, Remote VC lbl: N/A
  Local VC MTU: 1500, Remote VC MTU: 0
 CPU-Protection: OFF
 Local Switching: Enabled
```

Displaying VPLS auto-discovery information

The user can display the following information about the VPLS auto-discovery configuration:

- L2VPN VPLS address family and associated routes
- VPLS auto-discovered peers
- Load balancing status for VPLS auto-discovered peers
- LDP configuration, including the loopback interface and router ID

Displaying information about BGP L2VPN VPLS routes

The user can use the **show ip bgp l2vpn vpls** command with the parameters listed in Table 81 to view information related to BGP L2VPN VPLS routes.

TABLE 81 Parameters for CLI command show ip bgp l2vpn vpls

Parameter	Displays...	For details, see...
A.B.C.D or *A.B.C.D/L* (route IP address)	The BGP L2VPN VPLS routes for a particular IP route address	Viewing BGP L2VPN VPLS routes for a particular IP route address on page 560
attribute-entries	AS-path attribute entries	Viewing BGP L2VPN VPLS route attribute entries on page 560
neighbors	Details about TCP and BGP neighbor connections	Viewing neighbor connections on page 562
rd	Details about the route distinguisher	Viewing information for a route distinguisher on page 567
routes	Information about BGP L2VPN VPLS routes	Viewing information about BGP L2VPN VPLS routes on page 568
summary	A summary of the BGP L2VPN VPLS neighbor status	Viewing a summary of BGP neighbor status on page 569

Viewing all BGP L2VPN VPLS routes

The **show ip bgp l2vpn vpls** command displays all of the BGP L2VPN VPLS routes. The following shows example output.

```
device1# show ip bgp l2vpn vpls
Total number of BGP L2VPN VPLS Routes: 4
Status codes: s suppressed, d damped, h history, * valid, > best, i  internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network        Next Hop    Metric    LocPrf    Weight    Path
Route Distinguisher: 10:10
*>  10.1.1.1/32    0.0.0.0     0         100       65535     i
*i  10.2.2.2/32    0.0.0.0     0         100       0         i
Route Distinguisher: 10:20
*>  10.1.1.1/32    0.0.0.0     0         100       65535     i
*i  10.2.2.2/32    0.0.0.0     0         100       0         i
```

Syntax: show ip bgp l2vpn vpls

Table 82 defines the fields shown in the above example output.

TABLE 82 Output for the show ip bgp l2vpn vpls command

Output field	Description
Total number of BGP L2VPN VPLS Routes	The number of BGP4 routes in the BGP L2VPN VPLS route table.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route: • **s** (suppressed) – This route was suppressed during aggregation and thus is not advertised to neighbors. • **d** (damped) – This route has been dampened (by the route dampening feature), and is currently unusable. • **h** (history) – Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • ***** (valid) – The next-hop of this route can be resolved by the routing table. • **>** (best) – BGP4 has determined that this is the optimal route to the destination. • **i** (internal) – The route was learned through BGP4. • **S** (stale) – This route is stale and is cleaned up.
Origin codes	A list of the characters the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin code can be one of the following: • **i – IGP** – The routes with this set of attributes came to BGP4+ through IGP • **e – EGP** – The routes with this set of attributes came to BGP4+ through EGP. • **? – incomplete** – The routes came from an origin other than IGP or EGP. For example, they may have been redistributed from OSPF or RIP. **NOTE** When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.

TABLE 82 Output for the show ip bgp l2vpn vpls command (continued)

Output field	Description
Route Distinguisher	A unique ID that is pre-pended on any address being routed or advertised from a *Virtual Routing and Forwarding (VRF)* instance. The RD can be defined as either ASN-relative or IP address-relative as described: • **ASN-relative** - Composed of the local ASN number followed by a ':' (colon) and a unique arbitrary number. For example: 3:6 • **IP address-relative** - Composed of the local IP address followed by a ':' (colon) and a unique arbitrary number.
Network	The IP address and network mask of the destination network of the route.
Next Hop	The IP address of the next-hop router.
Metric	The cost of the routes that have this set of attributes.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this route associates with routes from a specific neighbor. For example, when the Extreme device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight
Path	The AS path for the route.

Viewing BGP L2VPN VPLS routes for a particular IP route address

The **show ip bgp l2vpn vpls** *IP route address* command displays the BGP L2VPN VPLS routes for a particular IP route address.

```
device(config-lbif-1)# show ip bgp l2vpn vpls 10.1.1.1
Total number of BGP L2VPN VPLS Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network        Next Hop   Metric  LocPrf   Weight   Path Route Distinguisher: 10:10
10.3.3.3/32    10.0.0.0   0       100      65535    i
```

Syntax: show ip bgp l2vpn vpls *IP route address*

Field definitions for the **show ip bgp l2vpn vpls** *IP route address* command are the same as for **show ip bgp l2vpn vpls**. Refer to

Viewing BGP L2VPN VPLS route attribute entries

Use the **show ip bgp l2vpn vpls attribute-entries** command to view attribute entries for BGP L2VPN VPLS routes.

```
device1# show ip bgp l2vpn vpls attribute-entries
        Total number of BGP Attribute Entries: 4 (2)
1       Next Hop  :10.0.0.0            Metric  :0              Origin:IGP
        Originator:10.0.0.0           Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0        Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 10:10
        AS Path   : (length 0)
        Address: 0x1431f5a2  Hash:108 (0x01000000), PeerIdx 0
        Links: 0x00000000, 0x00000000, nlri: 0x143709e4
        Reference Counts: 1:0:0, Magic: 5
2       Next Hop  :10.0.0.0            Metric  :0              Origin:IGP
        Originator:10.0.0.0           Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0        Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 10:20
        AS Path   : (length 0)
```

```
            Address: 0x1431f608  Hash:620 (0x01000000), PeerIdx 0
            Links: 0x00000000, 0x00000000, nlri: 0x14370a42
            Reference Counts: 1:0:0, Magic: 6
3           Next Hop  :10.0.0.0           Metric   :0              Origin:IGP
            Originator:10.0.0.0           Cluster List:None
            Aggregator:AS Number :0     Router-ID:0.0.0.0         Atomic:None
            Local Pref:100               Communities:Internet
            Extended Community: RT 10:10
            AS Path   : (length 0)
              AsPathLen: 0  AsNum: 0, SegmentNum: 0, Neighboring As: 0, Source As 0
            Address: 0x1431f4d6  Hash:108 (0x01000000), PeerIdx 4000
            Links: 0x00000000, 0x00000000, nlri: 0x14370928
            Reference Counts: 1:0:1, Magic: 3
4           Next Hop  :10.0.0.0           Metric   :0              Origin:IGP
            Originator:10.0.0.0           Cluster List:None
            Aggregator:AS Number :0     Router-ID:0.0.0.0         Atomic:None
            Local Pref:100               Communities:Internet
            Extended Community: RT 10:20
            AS Path   : (length 0)
              AsPathLen: 0  AsNum: 0, SegmentNum: 0, Neighboring As: 0, Source As 0
            Address: 0x1431f53c  Hash:620 (0x01000000), PeerIdx 4000
            Links: 0x00000000, 0x00000000, nlri: 0x14370986
            Reference Counts: 1:0:1, Magic: 4
```

Syntax: show ip bgp l2vpn vpls attribute-entries

Table 83 defines the fields shown in the above example output.

TABLE 83 Output for the show ip **bgp l2vpn vpls attribute-entries** command

Field output	Description
Total number of BGP Attribute Entries	The number of routes contained in this device's BGP L2VPN VPLS route table.
Next Hop	The IP address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP through EGP. • IGP - The routes with this set of attributes came to BGP through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. **NOTE** When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • `AS Number` shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • `Router-ID` shows the router that originated this aggregator.
Atomic	Indicates whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss.

TABLE 83 Output for the show ip **bgp l2vpn vpls attribute-entries** command (continued)

Field output	Description
	NOTE Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities to which routes with this set of attributes belong.
Extended Community	The extended community attributes.
AS Path	The AS path through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	This field is an internal value used for debugging purposes only.
Links	This field in an internal value used for debugging purposes only.
Reference Counts	This field is an internal value used for debugging purposes only.

Viewing neighbor connections

Use the **show ip bgp l2vpn vpls neighbors** command to view the details of TCP and BGP neighbor connections.

```
device1# show ip bgp l2vpn vpls neighbors
    Total number of BGP Neighbors: 1
1   IP Address: 10.1.1.2, AS: 10 (IBGP), RouterID: 10.2.2.2, VRF: default-vrf
    State: ESTABLISHED, Time: 0h15m47s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 49 seconds, HoldTimer Expire in 148 seconds
    Minimal Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
    Messages:    Open    Update   KeepAlive Notification Refresh-Req
       Sent   : 3       2        19        0            0
       Received: 1       2        18        0            0
    Last Update Time: NLRI      Withdraw         NLRI       Withdraw
              Tx: 0h15m47s   ---            Rx: 0h15m47s   ---
    Last Connection Reset Reason: Hold Timer Expired
    Notification Sent:    Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer Negotiated L2VPN VPLS address family
      Peer configured for IPV4 unicast  Routes
      Peer configured for L2VPN VPLS address family

 Neighbor AS4 Capability Negotiation:
    As-path attribute count: 2
    TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
     Maximum segment size: 1460
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   604, Received: 585
       Local host:  10.1.1.1, Local  Port: 179
       Remote host: 10.1.1.2, Remote Port: 8018
       ISentSeq:  310843582  SendNext:  310844187  TotUnAck:        0
       TotSent:         605  ReTrans:          0  UnAckSeq:  310844187
       IRcvSeq:   310909513  RcvNext:   310910099  SendWnd:     64981
       TotalRcv:        586  DupliRcv:         0  RcvWnd:      65000
       SendQue:           0  RcvQue:           0  CngstWnd:     3102
```

Syntax: show ip bgp l2vpn vpls neighbors

TABLE 84 Output for the show ip **bgp l2vpn vpls neighbors** command

Output field	Description
Total Number of BGP Neighbors	The number of BGP neighbors configured.

TABLE 84 Output for the show ip **bgp l2vpn vpls neighbors** command (continued)

Output field	Description
IP Address	The IP address of the neighbor.
AS	The AS number to which the neighbor belongs.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP - The neighbor is in another AS. • EBGP_Confed - The neighbor is a member of another sub-AS in the same confederation. • IBGP - The neighbor is in the same AS.
RouterID	The neighbor's router ID.
VRF	• default-vrf - The L2VPN is only applicable to the global default VRF instance.
State	The state of the Extreme device's session with the neighbor. The states are from this device's perspective of the session, not the neighbor's perspective. The state values are based on the BGP4 state machine values described in *RFC 1771* and can be one of the following for each router: • IDLE - The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. • ADMND - The neighbor has been administratively shut down. • CONNECT - BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE - BGP4 is waiting for a TCP connection from the neighbor. **NOTE** When the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT - BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM - BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. When the Extreme device receives a KEEPALIVE message from the neighbor, the state changes to Established. When the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED - BGP4 is ready to exchange UPDATE messages with the neighbor.
	Operational States: Additional information regarding the operational states of the BGP states described above may be added as described in the following: • (+) - Indicates that there is more BGP data in the TCP receiver queue. • (–) - indicates that the session has gone down and the software is clearing or removing routes. • (*) - indicates that the inbound or outbound policy is being updated for the peer. • (s) - indicates that the peer has negotiated restart, and the session is in a stale state. • (r) - indicates that the peer is restarting the BGP connection, through restart.

TABLE 84 Output for the show ip **bgp l2vpn vpls neighbors** command (continued)

Output field	Description
	• (^) - On the standby MP indicates that the peer is in the ESTABLISHED state and has received restart capability (in the primary MP). • (<) - Indicates that the Extreme device is waiting to receive the "End of RIB" message from the peer.
Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this device sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the Extreme device waits for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is dead.
Minimal Route Advertisement Interval	The minimum time elapse between route advertisements to the same neighbor.
RefreshCapability	Indicates whether this Extreme device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
Messages Sent	The number of messages this device has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this device has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	The last time updates were sent and received for the following: • NLRIs • Withdraws

Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following:
	• Reasons described in the BGP specifications:
	  – Message Header Error
	  – Connection Not Synchronized
	  – Bad Message Length
	  – Bad Message Type
	  – OPEN Message Error
	  – Unsupported Version Number
	  – Bad Peer AS Number
	  – Bad BGP Identifier
	  – Unsupported Optional Parameter
	  – Authentication Failure
	  – Unacceptable Hold Time
	  – Unsupported Capability
	  – UPDATE Message Error
	  – Malformed Attribute List
	  – Unrecognized Well-known Attribute
	  – Missing Well-known Attribute
	  – Attribute Flags Error
	  – Attribute Length Error
	  – Invalid ORIGIN Attribute
	  – Invalid NEXT_HOP Attribute
	  – Optional Attribute Error
	  – Invalid Network Field

TABLE 84 Output for the show ip **bgp l2vpn vpls neighbors** command (continued)

Output field	Description

	– Malformed AS_PATH
	    – Hold Timer Expired
	    – Finite State Machine Error
	    – Rcv Notification

	• Reasons specific to the implementation:
	    – Reset All Peer Sessions
	    – User Reset Peer Session
	    – Port State Down
	    – Peer Removed
	    – Peer Shutdown
	    – Peer AS Number Change
	    – Peer AS Confederation Change
	    – TCP Connection KeepAlive Timeout
	    – TCP Connection Closed by Remote
	    – TCP Data Stream Error Detected

Notification Sent	When the device sends a NOTIFICATION message to the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.
	
	• Message Header Error:
	    – Connection Not Synchronized
	    – Bad Message Length
	    – Bad Message Type
	    – Unspecified
	• Open Message Error:
	    – Unsupported Version
	    – Bad Peer As
	    – Bad BGP Identifier
	    – Unsupported Optional Parameter
	    – Authentication Failure
	    – Unacceptable Hold Time
	    – Unspecified
	• Update Message Error:
	    – Malformed Attribute List
	    – Unrecognized Attribute
	    – Missing Attribute
	    – Attribute Flag Error
	    – Attribute Length Error
	    – Invalid Origin Attribute
	    – Invalid NextHop Attribute
	    – Optional Attribute Error
	    – Invalid Network Field
	    – Malformed AS Path
	    – Unspecified
	• Hold Timer Expired
	• Finite State Machine Error
	• Cease
	• Unspecified

Notification Received	When the device receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages.

TABLE 84 Output for the show ip **bgp l2vpn vpls neighbors** command (continued)

Output field	Description
Neighbor NLRI Negotiation	The state of the NLRI negotiation with the neighbor. For example: • Peer negotiated IPv4 unicast capability • Peer negotiated L2VPN VPLS address family • Peer configured for IPv4 unicast routes • Peer configured for L2VPN VPLS address family
Neighbor AS4 Capability Negotiation	Whether this neighbor enabled 4 bytes ASN capability.
As-path attribute count	The number of unique path attributes learned from this neighbor.
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN - Waiting for a connection request. • SYN-SENT - Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED - Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED - Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 - Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 - Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT - Waiting for a connection termination request from the local user. • CLOSING - Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK - Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT - Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED - There is no connection state.
Maximum segment size	The TCP maximum segment size.
TTL check	The TCP TTL check.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the Extreme device.
Local port	The TCP port the Extreme device is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the Extreme device.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the Extreme device that have not been acknowledged by the neighbor.

Output field	Description
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the Extreme device re-transmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Viewing information for a route distinguisher

Use the **show ip bgp l2vpn vpls rd** command to view information for a particular route distinguisher.

```
device# show ip bgp l2vpn vpls rd 10:10
Total number of BGP Routes: 2
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network        Next Hop    Metric    LocPrf    Weight    Path
*>  10.1.1.1/32    10.0.0.0    0         100       65535     i
*i  10.2.2.2/32    10.0.0.0    0         100       0         i
```

Syntax: show ip bgp l2vpn vpls rd

TABLE 85 Output for the show ip **bgp l2vpn vpls rd** command

This field...	Displays
Total number of BGP Routes	The number of BGP4 routes the Extreme device has installed in the BGP L2VPN VPLS route table.
Status codes	A list of the characters the display uses to indicate the route's status. Refer to Viewing all BGP L2VPN VPLS routes on page 559.
Origin codes	A list of the characters the display uses to indicate the route's origin. Refer to Viewing all BGP L2VPN VPLS routes on page 559.
Network	The IP address and network mask of the destination network of the route.
Next Hop	The IP address of the next-hop router.
Metric	The cost of the route.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this route associates with routes from a specific neighbor. For example, when the Extreme device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The AS path for the route.

Viewing information about BGP L2VPN VPLS routes

Use the **show ip bgp l2vpn vpls routes** command to view information about BGP L2VPN VPLS routes.

```
device1# show ip bgp l2vpn vpls routes
Total number of BGP Routes: 4
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix            Next Hop     Metric   LocPrf   Weight Status
Route Distinguisher: 10:10
1      10.1.1.1/32       10.0.0.0     0        100      65535  BL
          AS_PATH:
2      10.2.2.2/32       10.0.0.0     0        100      0      I
          AS_PATH:
Route Distinguisher: 10:20
3      10.1.1.1/32       10.0.0.0     0        100      65535  BL
          AS_PATH:
4      10.2.2.2/32       10.0.0.0     0        100      0      I
          AS_PATH:
```

Syntax: show ip bgp l2vpn vpls routes

TABLE 86 Output for the show ip bgp l2vpn vpls routes command

Output field	Description
Total number of BGP Routes	The number of BGP4 routes the Extreme device has installed in the BGP4 route table.
Status	A list of the characters the display uses to indicate the route's status. The status code appears in the last column of the display, to the right of each route. The route's status can be one or more of the following: • A: AGGREGATE - The route is an aggregate route for multiple networks. • B: BEST - BGP4 has determined that this is the optimal route to the destination. • b: NOT-INSTALLED-BEST - The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table because the Extreme device received better routes from other sources (such as OSPF, RIP, or static IP routes). • C: CONFED_EBGP - The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D: DAMPED - This route has been dampened (by the route dampening feature), and is currently unusable. • E: EBGP - The route was learned from another AS BGP neighbor. • F: FILTERED - The route was filtered from the BGP route table. • H: HISTORY - Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I: IBGP - The route was learned from the same AS BGP neighbor. • L: LOCAL - The route originated on this Extreme device. • M: MULTIPATH - BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". • m: NOT-INSTALLED-MULTIPATH - The software was not able to install the route in the IP route table. • S: SUPPRESSED - This route was suppressed during aggregation and thus is not advertised to neighbors.

TABLE 86 Output for the show ip bgp l2vpn vpls routes command (continued)

Output field	Description
	• s: STALE - This is a stale route and is cleaned up.
Route Distinguisher	A unique ID that is prepended on any address being routed or advertised from a Virtual Routing and Forwarding (VRF) instance. The RD can be defined as either ASN-relative or IP address-relative as described: • ASN-relative - Composed of the local ASN number followed by a ':' (colon) and a unique arbitrary number. For example: 3:6 • IP address-relative - Composed of the local IP address followed by a ':' (colon) and a unique arbitrary number.
Prefix	The IP address and network mask of the destination network of the route.
AS_PATH	The BGP AS_PATH path attribute.
Next Hop	The IP address of the next-hop router.
Metric	The cost of this route.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this route associates with routes from a specific neighbor. For example, when the Extreme device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Status	The route's status. Refer to Viewing information about BGP L2VPN VPLS routes.

Viewing a summary of BGP neighbor status

Use the **show ip bgp l2vpn vpls summary** command to view BGP4 summary information.

```
device1# show ip bgp l2vpn vpls summary
  BGP4 Summary
  Router ID: 10.1.1.1   Local AS Number: 10
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 1, UP: 1
  Number of Routes Installed: 4, Uses 344 bytes
  Number of Routes Advertising to All Neighbors: 2, Uses 88 bytes
  Number of Attribute Entries Installed: 4, Uses 376 bytes
  Neighbor Address  AS#     State   Time      Rt:Accepted   Filtered Sent  ToSend
  10.1.1.2          10      ESTAB   0h 7m21s  2             0        2     0
```

Syntax: show ip bgp l2vpn vpls summary

TABLE 87 Output for the show ip bgp l2vpn vpls **summary** command

Output field	Description
Router ID	The Extreme device's router ID.
Local AS Number	The BGP4 AS number to which the Extreme device belongs.
Confederation Identifier	The AS number of the confederation to which the Extreme device belongs.
Confederation Peers	The numbers of the local ASs contained in the confederation. This list matches the confederation peer list the user configures on the Extreme device.
Maximum Number of IP ECMP Paths Supported for Load Sharing	The maximum number of route paths across which the device can balance traffic to the same destination.

TABLE 87 Output for the show ip bgp l2vpn vpls **summary** command (continued)

Output field	Description
Number of Neighbors Configured	The number of BGP4 neighbors configured on this Extreme device, and currently in established state.
Number of Routes Installed	The number of BGP4 routes in the Extreme device's BGP4 route table and the route or path memory usage.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors and the amount of memory used by these routes.
Number of Attribute Entries Installed	The number of BGP4 route-attribute entries in the device's route-attributes table and the amount of memory used by these entries.
Neighbor Address	The IP addresses of this device's BGP4 neighbors.
AS#	The AS number.
State	Refer to Viewing neighbor connections on page 562.mmd.
Time	The time that has passed since the state last changed.
Rt: Accepted	The number of routes received from the neighbor that this device installed in the BGP4 route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this device filtered out some of the routes received in the UPDATE messages.
Filtered	The routes or prefixes that have been filtered out: • When soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4 route table) but retained in memory. • When soft reconfiguration is not enabled, this field shows the number of BGP4 routes that have been filtered out.
Sent	The number of BGP4 routes that the Extreme device has sent to the neighbor.
ToSend	The number of routes the Extreme device has queued to send to this neighbor.

Displaying information about LDP

To display information about LDP, including the router ID and loopback interface in use, enter the **show mpls ldp** command.

```
device(config)# show mpls ldp
Label Distribution Protocol version 1
  LSR ID: 10.2.2.2, using Loopback 1 (deleting it will stop LDP)
  Hello interval: Link 5 sec, Targeted 15 sec
  Hold time value sent in Hellos: Link 15 sec, Targeted 45 sec
  Keepalive interval: 6 sec, Hold time multiple: 6 intervals
```

For additional information concerning the **show mpls ldp** command, see *NetIron Command Line Interface (CLI) Reference Guide*.

VPLS LSP Load Balancing

Glossary

TABLE 88 Glossary of terms

Term	Meaning
MAC	Media Access Control

TABLE 88 Glossary of terms (continued)

Term	Meaning
LSP	Label Switched Path
VPLS	Virtual Private LAN Service

Feature overview

This functional specification documents the VPLS LSP load balancing which is to be incremented from four LSPs to eight LSPs in the XMR Series and MLX Series product lines.

Limitations and prerequisites

An existing limitation is that the hashing technique to load balance is not consistent.

For example, The following unique MAC addresses/unique VLANs still cause traffic forwarding NOT distributed across all maximum 8 x LSPs.

```
IXIA PORT 1: SMAC = 00-00-00-11-00-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8  VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
IXIA PORT 2: SMAC = 00-00-00-22-00-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8  VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
or
IXIA P1: SMAC = 00-00-00-12-00-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
IXIA P2: SMAC = 00-00-00-13-00-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
or
IXIA P1: SMAC = 00-00-00-3c-01-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
IXIA P2: SMAC = 00-00-00-3d-01-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
```

BUT the following unique mac addresses/unique VLANs are working:

```
IXIA P1: SMAC = 00-00-00-03-01-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
IXIA P2: SMAC = 00-00-00-03-02-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
or
IXIA P1: SMAC = 00-00-00-03-12-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
IXIA P2: SMAC = 00-00-00-03-13-xx where xx = 1, 2, 3, 4, 5, 6, 7, 8     VLAN = 10yy  where yy = 10, 11,
12, 13, 14, 15, 16, 17
```

Feature enhancement

VPLS LSP load balancing is increased to eight LSPs after this feature.

In the CES 2000 Series and CER 2000 Series, you would be able to assign up to eight LSPs to a VPLS peer but at any time. Only one of them is chosen for all traffic forwarding for this VPLS peer because load balancing is not supported in the CES 2000 Series and CER 2000 Series.

Assumptions and dependencies

The hashing decision has not changed for this feature support. It is based on the fields in each packet received.

VPLS can use both LDP and RSVP tunnels for load balancing as long as they all matched the CoS criteria. A tunnel reachable to a peer with the right CoS value is all that is required to be used as a candidate for VPLS tunnel load balancing. There are no preferences for particular types of tunnels dependencies.

VPLS LSP load balancing configuration scenarios

Load balance with dynamic LSP selection

Configuration in Ingress router

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# cspf-interface-constraint
device(config-mpls)# path ve100
device(config-mpls-path-ve100)# strict 10.19.2.3
device(config-mpls)# path ve100
device(config-mpls-path-ve100)# strict 10.19.3.2
device(config-mpls)# mpls-interface ve 100
device(config-mpls)# mpls-interface ve 10
device(config-mpls)# lsp lsp1
device(config-mpls-lsp-lsp1)# to 10.19.19.19
device(config-mpls-lsp-lsp1)# primary ve100
device(config-mpls-lsp-lsp1)# enable
device(config-mpls)# lsp lsp2
device(config-mpls-lsp-lsp2)# 10.19.19.19
device(config-mpls-lsp-lsp2)# primary ve10
device(config-mpls-lsp-lsp2)# enable
device(config-mpls)# vpls v1 100
device(config-mpls-vpls-v1-100)# vpls-peer 10.19.19.19 load-balance
device(config-mpls-vpls-v1-100)# vlan 10
device(config-mpls-vpls-v1-100-vlan-10)# tagged ethe 4/17
device(config-mpls-vpls-v1-100)# vlan 100
device(config-mpls-vpls-v1-100-vlan-100)# tagged ethe 4/1
```

Configuration in Egress router

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# cspf-interface-constraint
device(config-mpls)# path ve100
device(config-mpls-path-ve100)# strict 19.19.2.2
device(config-mpls)# path ve10
device(config-mpls-path-ve10)# strict 19.19.3.1
device(config-mpls)# mpls-interface ve 100
device(config-mpls)# mpls-interface ve 10
device(config-mpls)# lsp lsp1
device(config-mpls-lsp-lsp1)# to 18.18.18.18
device(config-mpls-lsp-lsp1)# primary ve100
device(config-mpls-lsp-lsp1)# enable
device(config-mpls)# lsp lsp2
device(config-mpls-lsp-lsp2)# to 18.18.18.18
device(config-mpls-lsp-lsp2)# primary ve10
device(config-mpls-lsp-lsp2)# enable
device(config-mpls)# vpls v1 100
device(config-mpls-vpls-v1-100)# vpls-peer 18.18.18.18 load-balance
device(config-mpls-vpls-v1-100)# vlan 10
device(config-mpls-vpls-v1-100-vlan-10)# tagged ethe 4/17
device(config-mpls-vpls-v1-100)# vlan 100
device(config-mpls-vpls-v1-100-vlan-100)# tagged ethe 4/1
```

Load balance with manual LSP Assignment

Configuration on the Ingress router

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# cspf-interface-constraint
device(config-mpls)# path ve5
device(config-mpls-path-ve5)# strict 10.19.2.3
device(config-mpls)# path ve10
device(config-mpls-path-ve10)# strict 10.19.3.2
device(config-mpls)# mpls-interface ve5
device(config-mpls)# mpls-interface ve10
device(config-mpls)# lsp lsp1
device(config-mpls-lsp-lsp1)# to 10.19.19.19
device(config-mpls-lsp-lsp1)# primary ve10
device(config-mpls-lsp-lsp1)# enable
device(config-mpls)# lsp lsp2
device(config-mpls-lsp-lsp2)# to     10.19.19.19
device(config-mpls-lsp-lsp2)# primary ve10
device(config-mpls-lsp-lsp2)# enable
device(config-mpls)# vpls v1 100
device(config-mpls-vpls-v1-100)# vpls-peer 10.19.19.19 load-balance lsp lsp1 lsp2
device(config-mpls-vpls-v1-100)# vlan 10
device(config-mpls-vpls-v1-100-vlan-10)# tagged ethe 4/17
device(config-mpls-vpls-v1-100)# vlan 100
device(config-mpls-vpls-v1-100-vlan-100)# tagged ethe 4/1
```

Configuration in Egress router

```
device(config)# router mpls
device(config-mpls)# policy
device(config-mpls-policy)# cspf-interface-constraint
device(config-mpls)# path ve5
device(config-mpls-path-ve5)# strict 10.19.2.2
device(config-mpls)# path ve10
device(config-mpls-path-ve10)# strict 10.19.3.1
device(config-mpls)# mpls-interface ve5
device(config-mpls)# mpls-interfave ve10
device(config-mpls)# lsp lsp1
device(config-mpls-lsp-lsp1)# to 10.18.18.18
device(config-mpls-lsp-lsp1)# primary ve5
device(config-mpls-lsp-lsp1)# enable
device(config-mpls)# lsp lsp2
device(config-mpls-lsp-lsp2)# to 10.18.18.18
device(config-mpls-lsp-lsp2)# primary ve10
device(config-mpls-lsp-lsp2)# enable
device(config-mpls)# v1 v100
device(config-mpls-v1-v100)# vpls-peer 10.18.18.18 load-balance lsp lsp1 lsp2
device(config-mpls-v1-v100)# vlan 10
device(config-mpls-v1-v100-vlan-10)# tagged ethe 4/17
device(config-mpls-v1-v100)# vlan 100
device(config-mpls-v1-v100-vlan-100)# tagged ethe 4/1
```

VPLS static MAC

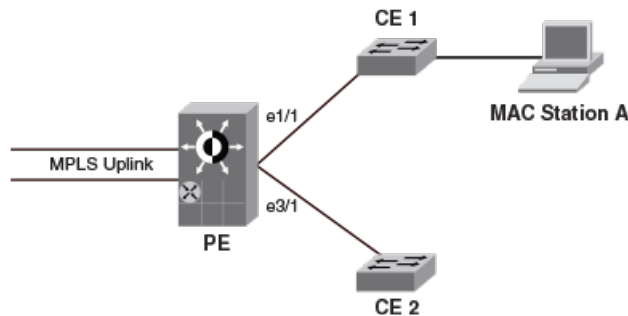The VPLS static MAC feature provides the ability to configure a static MAC address on a PE device and associate it to a VPLS endpoint.

Overview.

The VPLS static MACs captures the functionality and design for providing the ability to configure static MAC addresses at VPLS end points.

The diagram below explains how to configure PE device with static MAC and associate it to a VPLS endpoint.

VPLS static MAC enabled network

Consider MAC station **A** behind the customer edge router CE1. If **A** chooses to only receive and not to transmit any packets, then its MAC address is not learned on the provider edge PE1. When this happens, any traffic received at PE1 for **A** is flooded. If CPU protection is enabled, this flooding happens in the hardware, or else the flooding happens in the LP CPU. Also, if the link to CE1 goes down, the traffic destined to MAC Station A will be unnecessarily flooded across all the end-points.

To help in such situation, VPLS static MAC allows to configure a static MAC on a VPLS endpoint. Therefore, all packets destined to the static MAC station are hardware forwarded instead of flooding the CPU when no CPU protection is enabled. When the link to customer edge router CE1 goes down, the HW entry is reprogrammed to drop the traffic destined to MAC Station A, thereby protecting the CPU and preventing unwanted flooding in the network.

The following actions describe how static MAC is added and removed from the device.

Adding a static MAC

MAC Station **A** can be configured statically by following the configuration steps below. Once configured, the following actions are performed in the system.

1. The configured MAC address is added to the VPLS instance's MAC table in the MP.

2. The entry is also synchronized with the LPs VPLS MAC table.

   NOTE
   The maximum static MAC addresses that can be configured across all VPLS instances in the system is 1000.

Removing a static MAC

When the configuration is removed using the **no** form of the command, the following actions take place.

1. All configured hardware entries corresponding to the static MAC are deleted in the LPs.

2. The software entry is removed from the VPLS MAC table on both MP and LP.

Static MAC limit

The maximum number of static MACs that could be programmed is governed by the size of the VPLS MAC table. There is no other restriction on the number of static MACs that could be programmed.

   NOTE
   Static MACs are counted towards the total MACs learnt by the VPLS instance.

Hardware programming behavior for static MACs

Traffic destined to the statically configured MAC station are initially sent to CPU for forwarding, as there is no CAM entry in the hardware. Here the CPU forwards the packets because the software VPLS MAC table has the MAC.

This event causes the software to program the hardware so that the subsequent packets for the static MAC destination from this port are forwarded in the hardware. Therefore, the hardware is only programmed when a flow is seen for the static MAC. The programming is done only for the port on which the flow is seen to conserve the hardware resources which are used in forwarding. For MLX Series and XMR Series, once the hardware is programmed with the static MAC, it does not age out.

While creating a new hardware entry, the forwarding and dropping action depends on the state of the port on which the static MAC is configured. When the port on which static MAC is configured goes down, all programmed hardware entries are reprogrammed to drop the packets in the hardware. Once the port comes up, the programmed hardware entries are reprogrammed to forward the packets. The hardware entry also follows the STP state of the VPLS endpoint. When the port is blocked, the packets are dropped in the hardware by reprogramming the hardware entries. When the port state changes to forwarding, the hardware entries is reprogrammed to forward the packets.

Source Address learning behavior for static MACs

Learning actions for static MACs are disabled. When traffic is seen on an endpoint, whose source address (SA) matches with that of a configured static MAC, the SA learning event is not processed. At this time, the software will program a special SA CAM entry in the hardware against that port, which prevents subsequent packets from being sent from that port to the CPU for MAC learning. This helps in protecting the CPU from processing unnecessary MAC movement notifications for MACs which have already been configured as a static MAC.

SA learning behavior for MLX Series and XMR Series device

In VPLS, the CPU learns the SA and forwards the packets even if the destination address (DA) in the packet is known and programmed in the hardware. With this behavior, the user can expect packet loss when a new flow of traffic is introduced in the system destined to a static DA which may already be programmed in the hardware and this will continue until the new SA is learnt in the software and programmed in the hardware. This may cause an increment of drop counters in the TM to reflect CPU queue overflow when the rate of incoming traffic is high.

Behavior in CES 2000 Series and CER 2000 Series device

Once a static MAC is configured through the CLI, the FBD in the hardware is updated with this static MAC and will not age out. The only way to remove the hardware entry is by removing the static MAC configuration through CLI. Any traffic destined to this static MAC is always forwarded and not flooded in the hardware, unless the static MAC configuration is removed from the device. Once the static MAC configuration is removed through CLI, the FDB entry is removed and all the traffic destined to the removed static MAC is flooded in the hardware when the VPLS endpoint on which the Static MAC is configured goes down or goes to a blocking state, the FDB is reprogrammed to drop the packets destined for that MAC in the hardware.

> NOTE
> VPLS static MAC is supported only on tagged, double tagged, and untagged endpoints.

Forwarding Behavior

1. Local switching and traffic from MPLS uplink

   a. When a flow with the statically configured MAC as DA is seen for the first time on a port, the first few packets are sent to the CPU by the NP for forwarding and DA CAM entry programming.

   b. Once the CAM is programmed, subsequent traffic destined to the statically configured MAC station is forwarded in the hardware.

   c. Traffic destined to the statically configured MAC station is forwarded to the destination port if the port is UP.

   d. If the destination port is down, the flow is dropped in the ingress traffic manager.

Software aging behavior

The statically configured MAC entries in the VPLS MAC table never ages out in both MP and LP.

Hardware aging behavior

1. Entries programmed in the hardware for both VPLS endpoints and MPLS uplinks never age out.

2. The entries can only be deleted by removing the static MAC configuration from the CLI.

Supported end points

The following types of VPLS endpoints are supported:

1. Tagged

2. Untagged

3. Dual tagged

Hitless upgrade consideration

The VPLS subsystem is *not* hitless upgrade capable. There will be traffic loss during hitless upgrade.

Switchover behavior

1. If the standby MP is present while configuring the Static MAC on the active MP, the configuration is synchronized to the standby MP through existing configuration synchronization mechanism.

2. If the standby MP is inserted later after configuring Static MACs, the configuration is synchronized to the standby MP through existing configuration synchronization mechanism.

3. During switchover, the new active MP is always aware of the static MAC configurations made.

4. Switchover is hitless if the underlying protocols switchover without any hit.

Configuring static MAC address at VPLS endpoints

To configure the static MAC address at a VPLS endpoint:

VPLS must be pre-configured on the device before static MAC configuration.

1. Run the **router mpls** command to configure MPLS in the global configuration mode.

2. Run the **vpls** *vpls-id* command to define the VPLS ID.

3. Run the **vlan** command to configure a single tagged VLAN, to configure dual tagged VLAN run **vlan** *inner-vlan-id* command.

4. Run the **static-mac-address** *mac-address* **ethernet** *slot/port* command to configure static MAC on the VPLS endpoints.

Syntax
[no] **static-mac-address** *mac-address* **ethernet** *slot/port*

Description

mac-address specifies the MAC address of the system.

slot/port specifies the slot number or the port ID of the VPLS endpoints.

The following example explains how static MAC address is configured on a VPLS endpoints:

```
device(config)# router mpls
device(config-mpls)# vpls vpls-1 1
device(config-mpls-vpls-1)# vlan 900 inner-vlan 800
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.3333 ethernet 1/20
```

> **NOTE**
> The **no** form of this command will remove the static MAC configuration on a VPLS
> endpoints.

Limitations

- Static MACs can only be configured on VPLS endpoints.
- Configuring static MAC is not supported on a VPLS uplink.
- Static MACs cannot be configured if the VPLS instance has PBB or MCT configured.

VPLS static MAC error messages

Following are the error message displayed when VPLS static MAC is not supported for different scenarios.

If port not configured as part of the VPLS VLAN:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address
0000.1111.2222 ethernet 1/20

Error: port not part of this VPLS VLAN
```

If port is not part of this VPLS instance:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: port not part of this VPLS instance
```

If port is out of range, empty slot and if module type not configured in the system:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 4/1

Error: interface 4/1 is not an ETHERNET interface

device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/30

Error: invalid interface 1/30, if the interface is out of range.
```

If configuration is done on a secondary port of a LAG:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: Static MAC cannot be configured on a secondary port of a LAG.
```

If VPLS instance has PBB or MCT configured:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: Static MAC support not available for a VPLS with MCT or PBB enabled.
```

If VPLS instance has 802.1ah enabled (for bridging only):

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20

Error: Static MAC support not available for a VPLS with 802.1ah enabled
```

If MAC is a Zero MAC:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.0000.0000 ethernet 1/20

Error: Static MAC cannot be zero MAC.
```

If MAC is a Multicast MAC:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0100.1234.5678 ethernet 1/20

Error: Static MAC cannot be a multicast MAC.
```

If MAC is same as one of the local interface MACs:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0009.3400.0001 ethernet 1/20
```

Error: Static MAC cannot be same as interface MACs.

If MAC is Broadcast MAC:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address ffff.ffff.ffff ethernet 1/20
```

Error: Static MAC cannot be broadcast MAC.

If MAC is already configured on another port of the same VPLS instance:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20
```

Error: VPLS 2, MAC 0000.1111.2222 already exists on port 1/23, VLAN 900

If endpoint is double tagged:

Error: VPLS 2, MAC 0000.1111.2222 already exists on port 1/23, VLAN 900, Inner Tag: 1000

If Global VPLS MAC MAC limit reached:

Error: VPLS 1, Global VPLS MAC MAC limit (2048) reached.

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20
```

Error: VPLS 1, Global VPLS MAC MAC limit (2048) reached.

If per VPLS instance MAC MAC limit reached:

```
device(config-mpls-vpls-1-vlan-900)# static-mac-address 0000.1111.2222 ethernet 1/20
```

Error: VPLS 1, VPLS instance MAC limit (512) reached

Routing over VPLS

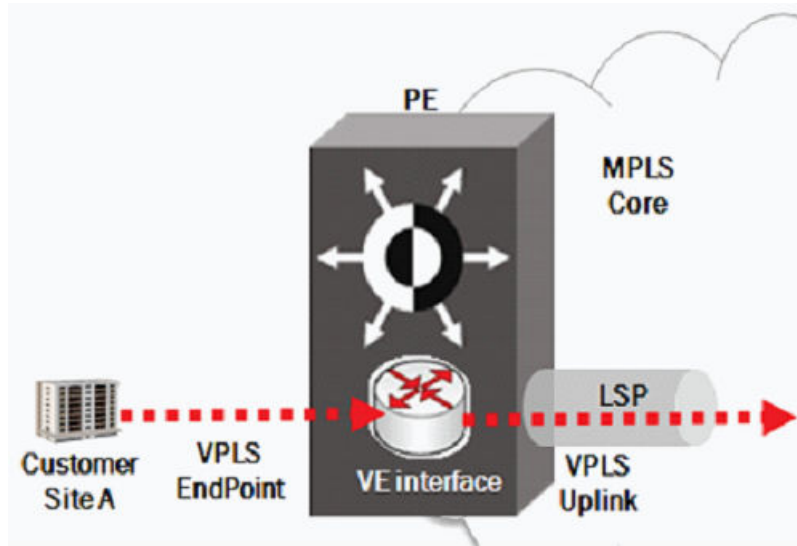## Routing over VPLS overview

Routing over VPLS provides routing functionality of a virtual Ethernet (VE) interface with Virtual Private LAN Services (VPLS) endpoints. By configuring a VE interface on a VPLS instance, VE routing packets arrive on the VPLS endpoint or uplink.

Virtual Private LAN Services (VPLS) enhances the point-to-point connectivity by specifying a method for Virtual Circuits (VCs) to provide point-to-multipoint connectivity across the Multiprotocol Label Switch (MPLS) domain, allowing traffic to flow between remotely connected sites belonging to a customer Virtual Private Network (VPN) as if the sites were connected by a Layer 2 switch. The Provider Edge (PE) devices learn the MAC addresses of locally connected customer devices, flood broadcast and unknown unicast frames to other PE devices in the VPN, and create associations between remote MAC addresses and the VC Label Switched Paths (LSPs) used to reach them. VE over VPLS routes packets between the VPLS VE interface and all other IP interfaces outside of VPLS domain which reside on the Provider Edge (PE) device including:

- Physical interfaces
- Other VLAN based VE interfaces for both tagged and untagged ports
- VE interfaces which reside on other VPLS instances

Routing over VPLS supports dynamic routing protocols such as OSPF and IS-IS on the VE over VPLS interface. Refer to Features supported for routing over VPLS on page 584 for additional protocol support information.

FIGURE 98 Routing over VPLS

In the sample topology below, you can view how various devices are configured as VPLS peers and where some VE interfaces are located.

FIGURE 99 Routing over VPLS topology
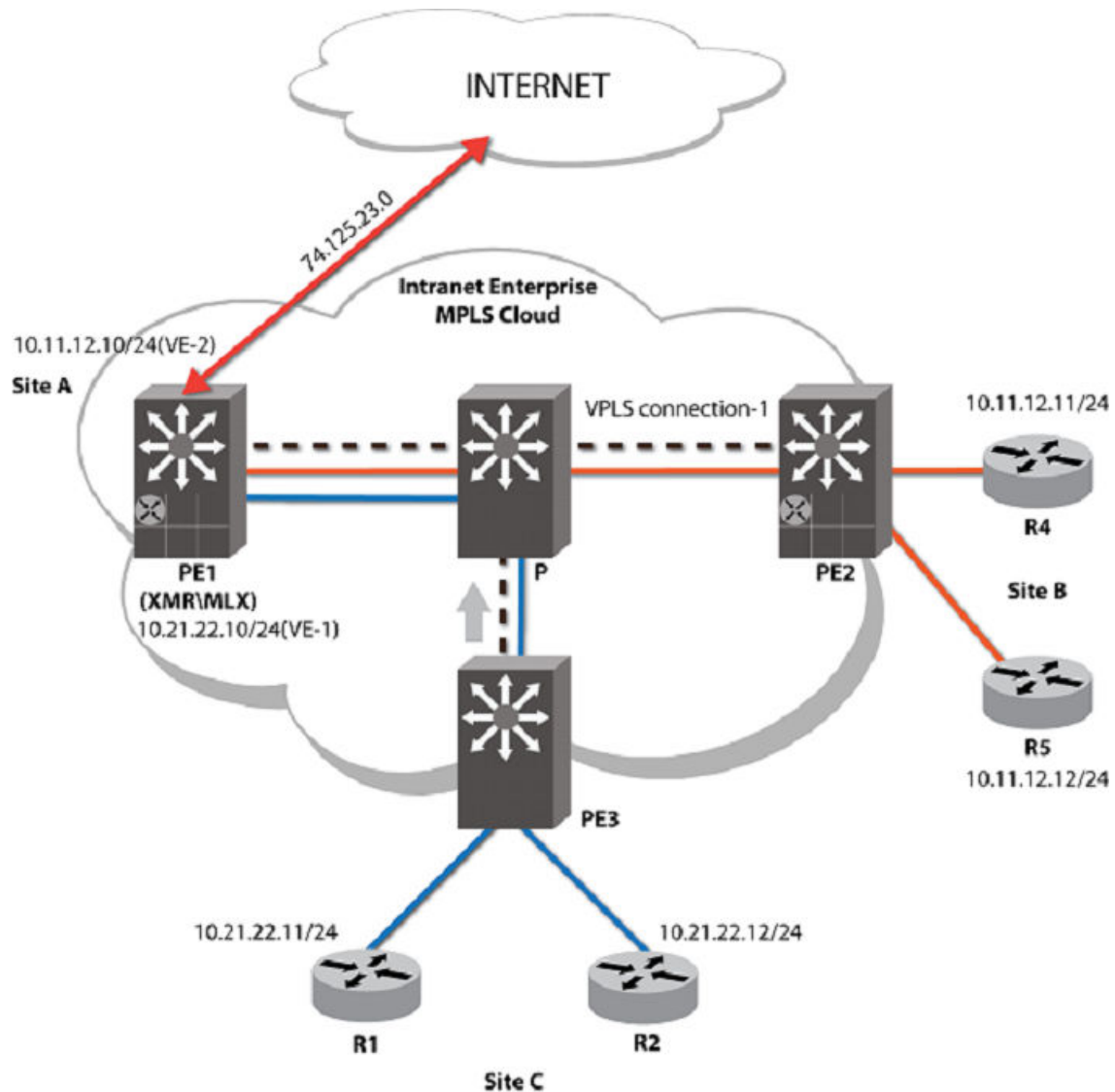
Routing over VPLS components

VE interface

VE interfaces can switch or route packets. This is decided by the destination L2 (Ethernet) MAC of the packet received at the VPLS end-point. When the Ethernet MAC is the Port MAC or Router MAC, the packet is routed.

Routing

This is the IP interface of the VE, which supports most of the existing functionalities of a legacy IP interface or the VE over VLAN interface.

- When the VE interface is disabled, all the routed packets to the interface are dropped.

- When the VE interface is not configured over a VPLS instance, all the routed traffic to the interface is L2 forwarded by VPLS. This ensures that it does not break the existing routing by way of loopback connections which may already be configured.

Switching

The VPLS L2 switching functionality remains unchanged. Refer to Features supported for routing over VPLS on page 584 for specific protocol support.

ARP

ARP maintains the relation between the L2-MAC to IP address. The VE over VPLS ARP maintains the same behavior.

- The ARP entries associated with VE over VPLS interface is resolved on one of the members of the VPLS instances, either local or remote-end point (remote VPLS peer)
- The ARP broadcast goes out through each of the end-point members of the VPLS instance
- Static ARP entries pointing to the VE over VPLS end-points is supported

ARP DAI

ARP DAI is not supported on the VE over VPLS ARP entries. The ports and VLANs associated with a VPLS instance are assumed to be always "trusted".

Re-ARP

When a resolved end-point or remote-peer for an ARP entry goes down, all the ARP entries pointing to that entity are flushed. Any further traffic to the remote host triggers a re-arp so that the ARP is resolved against the current active member from which the remote host could be reached.

- Proxy-arp is supported
- Local-proxy-arp is not supported

Unicast routing

VE over VPLS routing is only supported on the default and non-default VRF. Routing is supported between various endpoints (remote & local) of a VPLS-VE instance and other non-VPLS based IP interfaces.

- Supported unicast routing:

Packet Coming from the Local End-point of a VPLS-VE:

- to a VPLS uplink of same or another VPLS-VE instance
- to a local VPLS endpoint of same or another VPLS-VE instance
- to a VE over VLAN interface
- to a normal IP interface over a physical interface
- to an IP-over-MPLS interface
- to GRE tunnel

Packet coming from the Uplink (Remote End-point) of a VPLS-VE:

- to a VPLS uplink of another VPLS-VE instance
- to a local VPLS endpoint of same or another VPLS-VE instance

- to a VE over VLAN interface
- to a normal IP interface over a physical interface
- to an IP-over-MPLS interface

Packet coming from the non-VPLS-VE interfaces:

- Legacy, VE over VLAN, IPoMPLS, GRE tunnel interface to a VPLS-VE local endpoint
- Legacy, VE over VLAN, IPoMPLS interface to a VPLS-VE remote endpoint

Unsupported unicast routing:

- packet from a VPLS uplink to GRE tunnel
- packet from GRE tunnel to VPLS uplink

ECMP

When a VE is configured over a LSP load-balanced enabled VPLS interface, only the first active tunnel is used to complete the L3 forwarding. VPLS L2 switched traffic continues to load-balance on all tunnels.

VPLS packet routing

A VPLS instance with the VE interface enabled can participate in routing protocols exchanging routes with PEs on remote customer sites. Multiple VPLS instances can belong to the same VRF instance, for example, the user may configure different VPLS instances to different sites, while having all the sites in the same routing area.

A unicast packet that needs to be routed has the router's (PE) MAC in the MAC destination address field, and is subjected to Layer 3 processing. Routing for VPLS packets is no different than non-VPLS packets. The next hop obtained from the routing table could be a VLAN interface, a VPLS endpoint or a VPLS uplink.

> **NOTE**
> If VE is enabled on a VPLS instance then we cannot enable multicast snooping on the same VPLS instance. On the same lines, if multicast snooping is enabled on a VPLS instance, then we cannot enable VE on the same VPLS instance.

CPU protection

CPU Protection only applies to unknown unicast packets. When CPU protection is enabled, protocol packets such as ARP is always flooded by hardware.

VE over VPLS Peer FSM

When a VPLS interface is configured for VE, the peer is brought UP even when no local endpoints are configured.

> **NOTE**
> The CES 2000 Series and CER 2000 Series require at least one active endpoint for software forwarding.

VE over VPLS configuration and ICMP redirects

VE over VPLS adjacent devices can be on the same IP subnet while being part of the same VPLS segment. To prevent generation of ICMP redirects and sending of data packets to the CPU, it is recommended to turn-off "ICMP redirects". Note that ICMP redirect is ON by default.

New protocol priority classification supported

With routing over VPLS, the following protocols is now recognized at the MPLS interface and prioritized accordingly:

TABLE 89 Protocol priority classification

Protocol	Priority Group Number
VLL/VPLS Encapsulated IS-IS Hello Packets	7
VLL/VPLS Encapsulated OSPF V2 Hello	7
VLL/VPLS Encapsulated OSPF V3 Hello	7
VLL/VPLS Encapsulated OSPF	6
VLL/VPLS Encapsulated OSPFv3	6
VLL/VPLS Encapsulated RIP	6
VLL/VPLS Encapsulated RIPNG	6
VLL/VPLS Encapsulated VRRP	6
VLL/VPLS Encapsulated VRRP (IPv6)	6
VLL/VPLS Encapsulated VRRPE	6
VLL/VPLS Encapsulated VRRPE (IPv6)	6
VLL/VPLS Encapsulated BGP	5
VLL/VPLS Encapsulated BGP (IPv6)	5
VLL/VPLS Encapsulated ARP	3

Features supported for routing over VPLS

List of features supported when using routing over VPLS. Some differences in support are noted for VRF over VEoVPLS.

TABLE 90 Routing over VPLS supported features

Protocol	Support
VRF	Supported.
IP Routing	Only unicast routing is supported. Multicast routing is not supported.
ARP and Static ARP	Supported
Multi-port Static ARP	Not Supported
Local-proxy-ARP	Not Supported
Proxy ARP	Supported
ARP - DAI	Not supported
Traceroute	Supported
ICMP	Supported
ICMP Redirect Message	Supported
ICMP Unreachable Message	Not supported
OSPF	Supported
IS-IS	Supported, except for VRF VEoVPLS
RIP	Supported
BGP	Supported
Trunk Ports (LAG) Supported	Supported
L2 Multicast	Supported only for VRF VEoVPLS, otherwise not supported.
IGMP Snooping	Supported

TABLE 90 Routing over VPLS supported features (continued)

Protocol	Support
L2 ACL	Supported
L3 ACL	Supported
Rate Limiting	Supported
PBR	Not supported
Multi-netting	Supported
VRRP/VRRP-E	Not supported except for VRF VEoVPLS
	Supported (MLX Series and XMR Series only)
IP Helper Address / BootP / DHCP	Supported
ECMP	Supported
IP MTU	Not supported.
	NOTE IP MTU is ignored for L3 packets whose next hop is over the VPLS local endpoint or VPLS remote endpoint no matter what interface the packet arrived from.
VE over VPLS as MPLS interface	Not supported. The VE configured on VPLS cannot be an MPLS interface.
Dual Tag Mode	Not supported
L3 Multicast (PIM)	Not supported
BFD	Not supported
GRE	Not supported
RPF	Not supported on MPLS Uplinks on MLX Series and XMR Series routers.
	Not supported on the CES 2000 Series and CER 2000 Series devices.
IPv6	Supported, except for VRF VEoVPLS
PBB	Not supported
MCT	Supported, except for VRF VEoVPLS
IP Unnumbered	Not supported
IRDP	Not supported
Route-only	Not supported
DHCP	Supported
ACL - **ip acc redirect-deny-to-interf** *slot/port*	Supported on MLX Series and XMR Series routers.
	Not supported on the CES 2000 Series and CER 2000 Series devices.
	Not supported for VRF VEoVPLS

Modules supported

Use the table below to determine when routing over VPLS is supported on specific interface module.

TABLE 91 Modules supported when using routing over VPLS

Part Number	Support
BR-MLX-1GFX24-X	Supported
BR-MLX-1GFX24-X-ML	Supported
BR-MLX-1GCX24-X	Supported

TABLE 91 Modules supported when using routing over VPLS (continued)

| Part Number	Support
BR-MLX-1GCX24-X-ML	Supported
BR-MLX-10Gx4-X	Supported
BR-MLX-10Gx8-X	Supported
BR-MLX-10Gx8-M	Supported
BR-MLX-10Gx8-D	Supported
BR-MLX-100Gx1-X	Supported
BR-MLX-100Gx2-X	Supported
BR-MLX-10Gx24-DM	Not supported

Scalability

Routing over VPLS supports the following scalability numbers:

TABLE 92 Supported scaling

Scaling Item	XMR Series	MLX Series	CES 2000 Series	CER 2000 Series
Maximum number of VPLS Instances that can configure with VE	4K	4K	128	1024
Maximum VE interfaces allowed on the system	4K	4K	1K	4K

VE over VPLS sample topology

FIGURE 100 Routing over VPLS topology
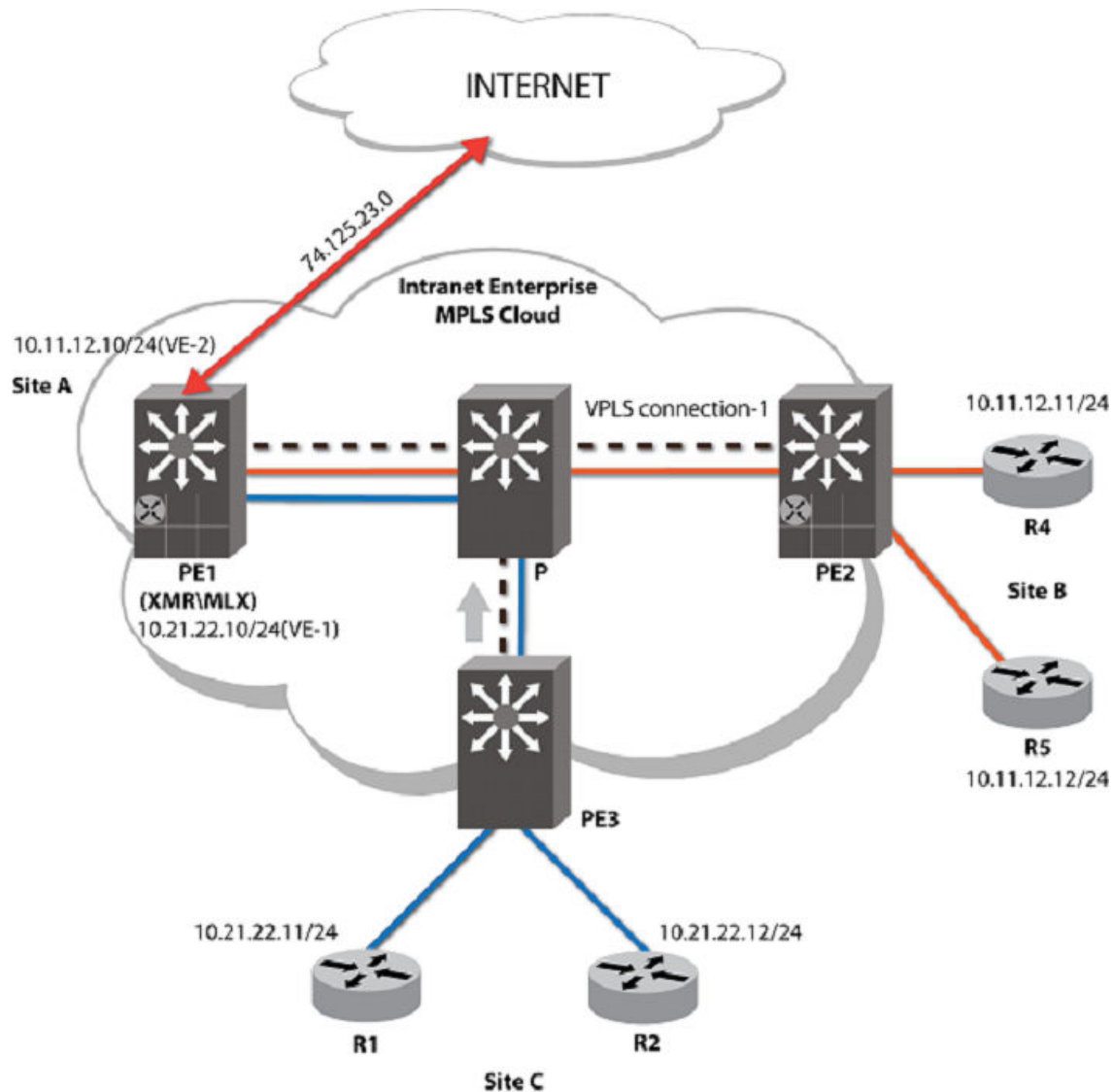
Configuration Considerations for routing over VPLS

Consider the following before configuring routing over VPLS.

- When using VE over VPLS, the user must use a different loopback interface for MPLS and GRE tunnels.
- IGP may install VE over VPLS as a preferred route once it comes up, and MPLS protocols such as LDP uses the VE over VPLS IP interface for control traffic.
- When the loopback interface used by LDP is shared with GRE as one of the tunnel interfaces, the LDP goes down forcing the VE over VPLS PW to go down.
- VE over VPLS cannot be enabled on a VPLS instance that has dual tag or ISID configured.

- VE over VPLS is not supported on 24x10G modules for any type of endpoint.
- VE over VPLS and PBB cannot be enabled on the same VPLS instance.

Migration considerations

Routing over VPLS does not affect the legacy loopback configuration already deployed. When the user chooses to migrate to the VE over VPLS solution, the following migration checks must be considered.

- When the user is using a non-default VRF VE interface, the user must continue to use the existing loopback emulation.
- When the user is using the loopback configuration, and are using the VE interface on the default-VRF, the user can move to the new VE over VPLS interface, provided all of the features the user is currently using on the VE interface are supported using the new VEoVPLS interface. Refer to Features supported for routing over VPLS on page 584 for additional information.

Configuring VE over VPLS

For information on configuring VPLS, refer to Configuring VE over VPLS
Use the **router-interface ve** command to configure the VE per VPLS instance. For additional information regarding this command, go to *NetIron Command Line Interface (CLI) Reference Guide*.

Consistency checks

The following error messages are a sample of what is seen when trying to add an invalid configuration.

1. Double tag not supported in the VPLS instance which has VE enabled.

   In this example, when a VPLS instance has VE enabled, it is blocked from configuring a double tag.

   ```
   device(config)# router mpls
   device(config-mpls)# vpls vinst 1000
   device(config-mpls-vpls-vinst)# router-interface ve 3
   device(config-mpls-vpls-vinst)# vlan 20 inner-vlan 30
   Error - Dual tag or ISID is not supported in a VPLS instace that has VE enabled
   device(config-mpls-vpls-vinst-vlan-10)#
   device(config)# router mpls
   device(config-mpls)# vpls vinst 1000
   device(config-mpls-vpls-vinst)# vlan 20 inner-vlan 30
   device(config-mpls-vpls-vinst-vlan-20-inner-vlan-30)# vpls vinst 1000
   device(config-mpls-vpls-vinst)#
   device(config-mpls-vpls-vinst)# router-interface ve 3
   Error - VE over VPLS cannot be enabled on a VPLS instance that has Dual tag or ISID configured
   ```

2. Configuration check that disallows a VE over VPLS configuration on unsupported interface modules.

   In this example, a VE over VPLS is not supported on 24x10G modules for any type of endpoint.

   ```
   device(config)# router mpls
   device(config-mpls)# vpls vinst 2000
   device(config-mpls-vpls-vinst)# router-interface ve 3
   device(config-mpls-vpls-vinst)# vlan 200
   device(config-mpls-vpls-vinst-vlan-200)# tagged ethernet 4/2
   Error - VE over VPLS cannot be enabled on 24X10G ports. One of the ports in this instance is a
   24X10G port
   ```

3. Configuration check that disallows a VE over VPLS with a PBB configuration.

    In this example, an error message is displayed when the user tries to enable both VE over VPLS and PBB in the same VPLS instance.

    ```
    device(config)# router mpls
    device(config-mpls)# vpls vinst 2000
    device(config-mpls-vpls-vinst)# router-interface ve 3
    device(config-mpls-vpls-vinst)# pbb
    Error - VE over VPLS and PBB cannot be enabled on the same VPLS instance.

    device(config)# router mpls
    device(config-mpls)#vpls vinst 2000
    device(config-mpls-vpls-vinst)# pbb
    device(config-mpls-vpls-vinst)# router-interface ve 3
    Error - VE over VPLS and PBB cannot be enabled on the same VPLS instance.
    ```

4. Configuration to enable VRRP/E to track MCT VPLS state.

    This sample set of configuration indicates the new configuration in VRRP (VE interface level) to track the MCT VPLS state.

    ```
    device(config)# interface ve 100
    device(config-vif-100)# ip vrrp vrid 1
    device(config-vif-100-vrid-1)# track-object mct-vpls-state

    device(config-vif-100-vrid-1)# show run interface ve 100
    interface ve 100
     ip ospf area 1
     ip address 12.100.0.11/24
     ip vrrp vrid 1
      backup priority 100 track-priority 90
      ip-address 12.100.0.55
      track-object mct-vpls-state
      activate
    !
    This tracking object has to be enabled per vrid to enable tracking of mct vpls state. Track-priority
    will be used to update VRRP/E current priority when MCT VPLS role changes.
    The tracking object config is applicable for ve over vpls interfaces only.
    Also this config should be there in both MCT nodes.
    ```

VRRP/VRRP-E support

VRRP/VRRP-E is supported for routing over VPLS. The following VRRP/VRRP-E functionality is expected when using it in a routing over VPLS configuration. VRRP/e functionality expects VPLS full mesh configuration in which each VPLS peer is connected to all other VPLS peers in a given VPLS instance.

VRRP/VRRP-E control message flow

VRRP/VRRP-E control messages are sent over VE over VPLS interface as regular VE interface. The VRRP/VRRP-E messages received from remote peers are forwarded to endpoints, but not to other remote peers. Once a node is selected as master, it sends out regular gratuitous ARPs with a VRRP/VRRP-E MAC address as the MAC for the gateway IP, causing all the endpoints and remote peers (of other instances) to learn the master MAC address.

VRRP backup

The backup learns the master virtual MAC as a regular Layer2 MAC. Traffic from the backup to the master is Layer2 switched.

VRRP-E master

Traffic from is routed from the VRRP-E master, based on the destination IP address.

VRRP-E backup

When **short-path-forwarding** is not enabled on the VRRP-E backup, then forwarding of the VRRP-E backup is the same as VRRP backup. When **short-path-forwarding** is enabled, the VRRP/VRRP-E backup itself can route packet instead of sending to the master.

VRRP/VRRP-E master backup state change

When the VRRP/VRRP-E master becomes the backup, it flushes all the CAM entries. When the VRRP-E backup is enabled with **short-path-forwarding**, then when the state changes from master to backup, the CAM entries are retained.

VRRP/VRRP-E configuration change

When the VPLS end point is added or deleted or a new remote peer is added or deleted, for the VRRP/VRRP-E master or VRRP-E backup with **short-path-forwarding** enabled, the entries in the CAM are added or deleted respectively enabling the reception of the VRRP related traffic.

Protocol priority classification

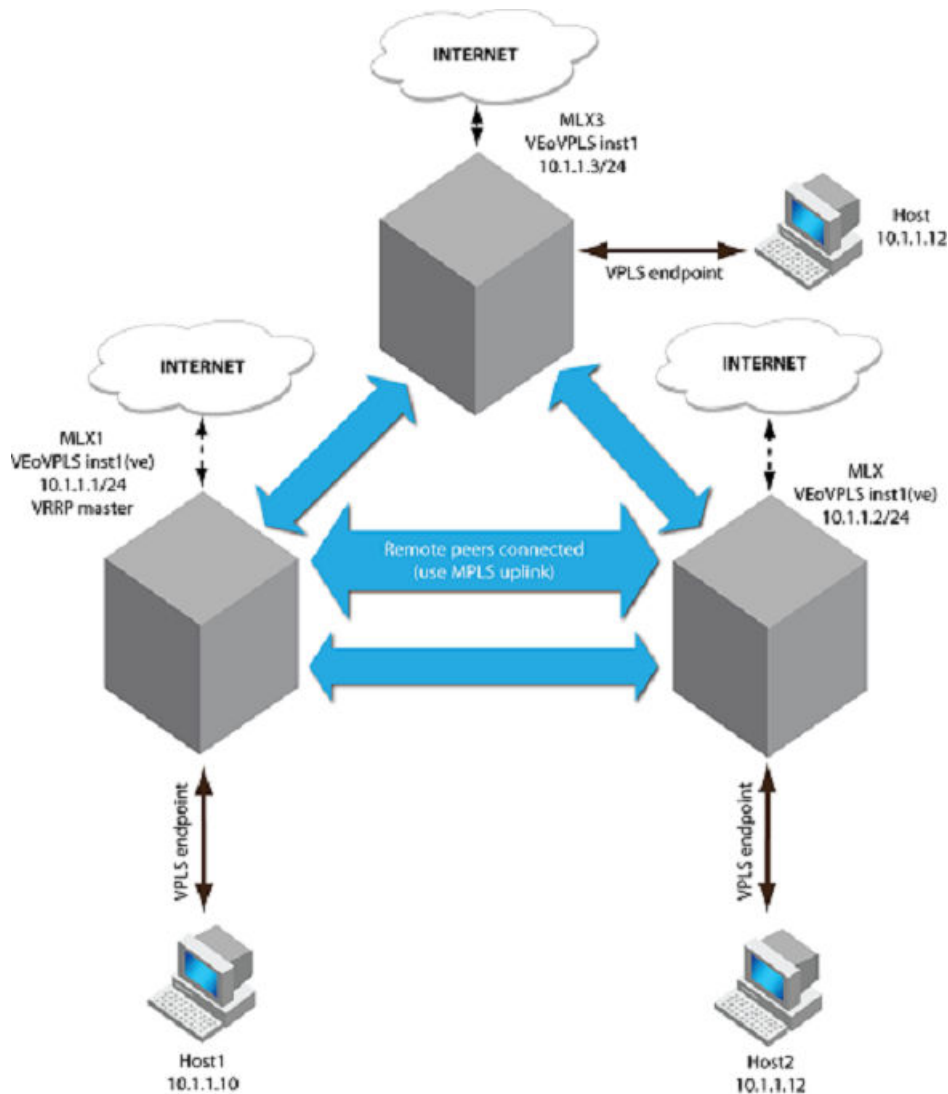VRRP/VRRP-E control packets travel with a priority of 6 within the box.

TABLE 93 Protocol priority classification

Protocol	Priority group number
VRRP/VRRP-E protocol packets	6

The following topologies are supported:

Single homing topology for VRRP/VRRP-E over VPLS

FIGURE 101 Single homing topology example

Consider the following when using VRRP/VRRP-E with routing over VPLS in a single homing topology.

- In the single homing topology, a cost is connected to a single VPLS peer. Note that the host can be connected directly or through an access switch
- The host can be directly connected to the VRRP/VRRP-E master or backup devices
- The **no ip icmp redirects** command must be configured in VRRP cases and in VRRP-E cases where server virtualization (short-path forwarding) is not enabled.

Configuration considerations

Consider the following when using VRRP/VRRP-E with routing over VPLS in a single homing topology.

- In the single homing topology, a cost is connected to a single VPLS peer. Note that the host can be connected directly or through an access switch
- The host can be directly connected to the VPPE/e master or backup
- The **no ip icmp redirects** command must be configured in VRRP cases and in VRRP-E cases where server virtualization is not enabled.

Dual homing topology for VRRP/VRRP-E over VPLS

FIGURE 102 Dual homing topology example

Extreme NetIron MPLS Configuration Guide, 06.1.00
53-1004905-02

The configuration is the same for VRRP or VRRP-E over VPLS VE.

Consider the following when using VRRP and VRRP-E with routing over VPLS in a dual homing topology.

- A single host can be used to connect to two VPLS nodes within the same VPLS instance for dual homing support. It provides redundancy in the event of VPLS node failure.
- On each node, include the VPLS endpoint VLAN in the topology group VLAN as a member VLAN, so the link from one of the nodes to the access box is blocked for Layer 2 loop avoidance. Configure the master VLAN to have a Layer 2 protocol configuration, such as Metro Ring Protocol (MRP) or Rapid Spanning Tree Protocol (RSTP).
- The recommended configuration is VRRP-E with server virtualization (short-path forwarding). Short-path forwarding helps the traffic in the backup node to be routed directly to the internet, instead of being sent to the VRRP-e master over the MPLS uplink.
- The **no ip icmp redirects** command must be configured in a VRRP instance as well as VRRP-E instances where short-path forwarding is not enabled.

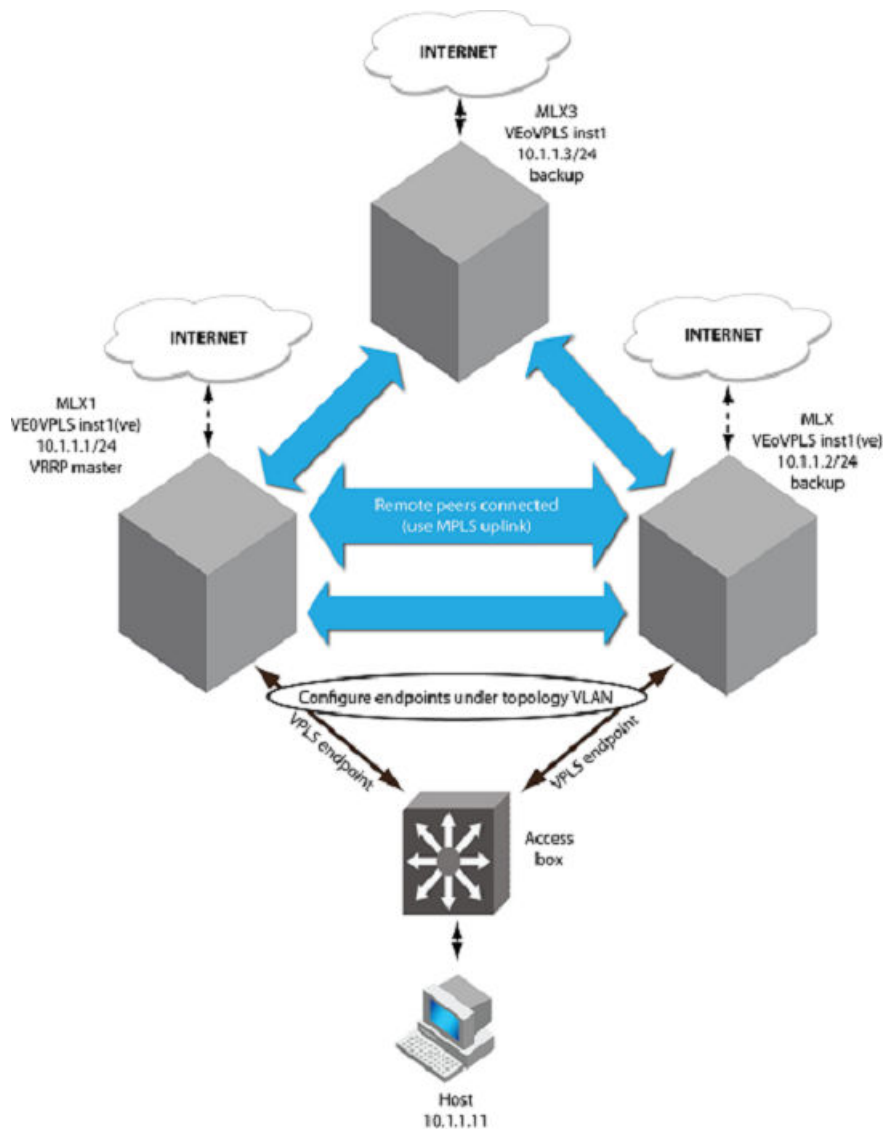## Configuration considerations

Consider the following when using VRRP and VRRP-E with routing over VPLS in a duel homing topology.

- A single host can be used to connect to two VPLS nodes within the same VPLS instance for dual homing support. It provides redundancy in the event of VPLS node failure.
- On each node, include the VPLS endpoint VLAN in the topology group VLAN as a member VLAN, so the link from one of the nodes to the access box is blocked for Layer 2 loop avoidance. Configure the master VLAN to have a Layer 2 protocol configuration, such as MRP/RSTP. Refer to the topology group VLAN configuration for setting VPLS VLAN in the topology group VLAN.
- The recommended configuration is VRRP-E with server virtualization.
- Server virtualization helps the traffic in the backup node to be routed directly to the internet, instead of being sent to the VRRP-e master over the MPLS uplink.
- The **no ip icmp redirects** command must be configured in a VRRP case as well as VRRP-E cases where server virtualization is not enabled.

MCT Support for VE over VPLS

Virtual Private LAN Services (VPLS) routes layer 3 traffic between provider edge (PE) devices in VPLS multipoint PE connections. Adding MCT support allows a VPLS Service Access Point (SAP) interface to be configured with a Layer 3 IP address making the interface routable.

The VE over VPLS feature combines routing functionality of a VE interface with VPLS endpoints. Using MCT, we achieve High Availability (HA) and dual-homing support. The MCT configuration is combined with VRRP or VRRP-E configuration to protect against link and PE router failure.

Using this feature, layer 2 and layer 3 services can be simultaneously provided over the same VPLS interfaces on an MCT-connected Extreme device. This maximizes the customer investment in terms of number of ports while also providing full redundancy to both co-located as well as geo-redundant PE placement for data center interconnect.

The MCT support for VE over VPLS is supported only on the XMR Series platforms and the MLX Series platforms.

Consider the following when configuring VE over VPLS with MCT.

- The VE interface is configured on all PE nodes.

- The router VE instance is created in the MCT VPLS instance.

- Routing protocols running on both the MCT nodes point to the same next-hop address for a specified route.

- In the absence of a configured or enabled VPLS endpoint, when the routing-interface ve command on that MCT VPLS instance is configured, the network enables the VPLS FSM. This is required to support routing for packets from remote peers.

- Packets are dropped from a non Active-Active remote VPLS peer. Here the reference is to the MCT VPLS Active/Standby preferential status.

- Hitless upgrade and MP switchover are not supported.

- Not compatible with the 24x10g module.

- Not compatible with the POS module.

FIGURE 103 MCT support for VE over VPLS
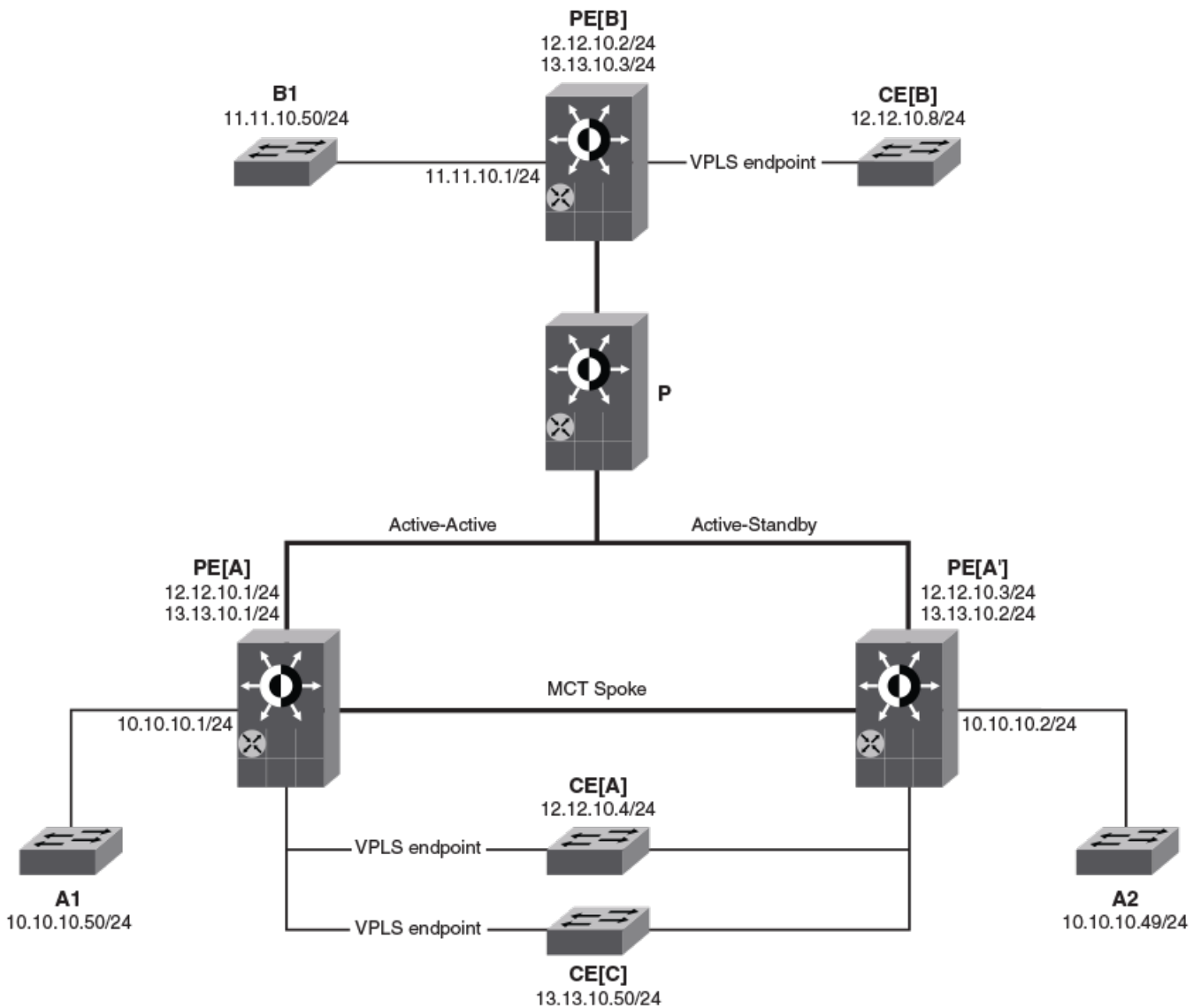
Topology Description

- PE[A] and PE[A'] are the two nodes of the MCT Cluster. PE[A] is MCT Active and PE[A'] is Standby related to MCT VPLS.
- PE[B] is the remote peer that has MPLS connectivity to PE[A] and PE[A']. The remote PW from PE[A] to PE[B] is Active-Active and remote PW from PE[A'] to PE[B] is Standby-Active.
- CE[A] and CE[C] are connected to the two MCT nodes of the cluster using LAG. The links connected to CE[A] and CE[C] are called the Cluster Client Edge Port (CCEP) end points.
- A1 and A2 are single homed to PE[A] and PE[A'] respectively. These are the Cluster Edge Port (CEP) end points.
- PE[A], PE[A'], PE[B], CE[A], and CE[B] have IP interface in same IP 12.12.10.0 subnet.
- PE[A], PE[A'], PE[B], CE[A], and CE[C] have IP interface in same IP 13.13.10.0 subnet.

Configuration Considerations

Consider the following when configuring VE over VPLS with MCT.

- The VE interface is configured on all PE nodes.
- The router VE instance is created in the MCT VPLS instance.
- Routing protocols running on both the MCT nodes point to the same next-hop address for a specified route.
- In the absence of a configured or enabled VPLS endpoint, when the routing-interface ve command on that MCT VPLS instance is configured, the network enables the VPLS FSM. This is required to support routing for packets from remote peers.
- Packets are dropped from a non Active-Active remote VPLS peer. Here the reference is to the MCT VPLS Active/Standby preferential status.

Use case scenarios

The following scenarios describe packet flow for VE over VPLS routing in non-MCT and MCT configurations.

- Packet flow from Non-VPLS end-point host to Non-VPLS end-point host on page 595
- Packet flow from non-VPLS end-point host to VPLS end-point host on page 596
- Packet flow from VPLS end-point host to VPLS end-point host on page 596

Packet flow from Non-VPLS end-point host to Non-VPLS end-point host

The following lists the packet flow in detail for packets from a non-VPLS host A2 (10.10.10.49) to the non-VPLS host B1 (11.11.10.50) referring to the Figure 104 on page 594 illustration.

1. Host A2 does a route lookup for 11.11.10.50, which points the next hop to 10.10.10.2 PE [A'] router.
2. A2 broadcasts ARP request for 10.10.10.2.
3. PE [A'] router unicasts a ARP reply to A2 thereby resolving the ARP.
4. A2 sends the packet to 10.10.10.2.
5. The PE [A'] hardware IP CAM does a lookup for 11.11.10.50, which points the next hop to 12.12.10.2 that is on VPLS uplink, which is part of VPLS-VE interface on PE [A].
6. The ARP for PE [B] on PE [A'] is resolved on the MCT spoke and it points to PE [A].
7. PE [A'] (12.12.10.3) forwards the packet to PE [B] (12.12.10.2) over the MPLS tunnel.

8. When the packet reaches PE [B] on the VPLS-VE interface, the hardware IP CAM does a lookup for the destination 11.11.10.50, which hits subnet CAM entry 11.11.10.0/24, being the 1st packet closest to 11.11.10.50. This results in the packet to go to the CPU.

9. PE [B] looks up its routing table for 11.11.10.50, which is on the directly connected subnet, on a non-VPLS domain.

10. PE [B] broadcasts a ARP request for 11.11.10.50 on the subnet.

11. Host B1 unicasts a ARP reply to 11.11.10.1, and ARP is resolved and hardware IP CAM for 11.11.10.50 is programmed. The 11.11.10.1 host forwards the packet to 11.11.10.50.

Packet flow from non-VPLS end-point host to VPLS end-point host

The following lists the packet flow in detail for packets from a non-VPLS host A2 (10.10.10.49) to the VPLS host CE [B] (12.12.10.8) referring to the Figure 104 on page 594 illustration.

1. Host A2 does a route lookup for CE [B], which points the next hop to 10.10.10.2.

2. CE [B] broadcasts a ARP request for 10.10.10.2.

3. PE [A'] unicasts a ARP reply to A2 thereby resolving the ARP.

4. A2 sends the packet to 11.11.10.2.

5. Once the packet reaches 10.10.10.2, hardware lookup is performed for the destination IP address 12.12.10.8, which hits subnet CAM entry 12.12.10.0/24 being the first packet to 12.12.10.8 that causes the packet to come to CPU.

6. PE [A'] does a route lookup in software for 12.12.10.8, which points 12.12.10.8 to be on its directly connected network over VPLS, which is part of the VPLS-VE interface.

7. ARP for 12.12.10.8 will be resolved over the MCT spoke PW.

8. The Packet gets forwarded over the MPLS tunnel from PE [A'] to PE [B] via PE [A].

9. IP CAM is programmed on PE [B] for 12.12.10.8 to facilitate hardware forwarding of packets towards 12.12.10.8 from this point onwards.

10. When the packet reaches 12.12.10.2 (PE [B]) it will be VPLS L2-switched to 12.12.10.8.

Packet flow from VPLS end-point host to VPLS end-point host

The following lists the packet flow in detail for packets from the VPLS router CE [A] to the VPLS router CE [B] referring to the Figure 104 on page 594 illustration.

1. Since router CE [A] interface 12.12.10.4 and CE [B] interface 12.12.10.8 are on same network being part of the same VPLS domain, router CE [A] does a route lookup for CE [B] and finds it to be on the directly connected interface.

2. CE [A] broadcasts a ARP request for CE [B] to resolve the MAC address.

3. The broadcasted ARP request gets replicated twice once at PE [A] and PE [B] and reaches CE [B].

4. CE [B] unicasts the ARP reply containing its MAC address to CE [A] thereby resolving the ARP.

5. CE [A] sends the packet to CE [B] which if goes to PE [A'] (MCT client is a lag) will be VPLS switched to PE [A] and then VPLS switched to CE [B]. PE [A] VE over VPLS mac which was synchronized to PE [A'] is used to forward traffic to PE [A].

ACL Support for VE over VPLS

VE over VPLS uses the same ACL commands as VE for VLANs to apply an IPv4 ACL on VE over VPLS interfaces to filter both switched and routed L3 and L4 traffic in incoming and outgoing directions.
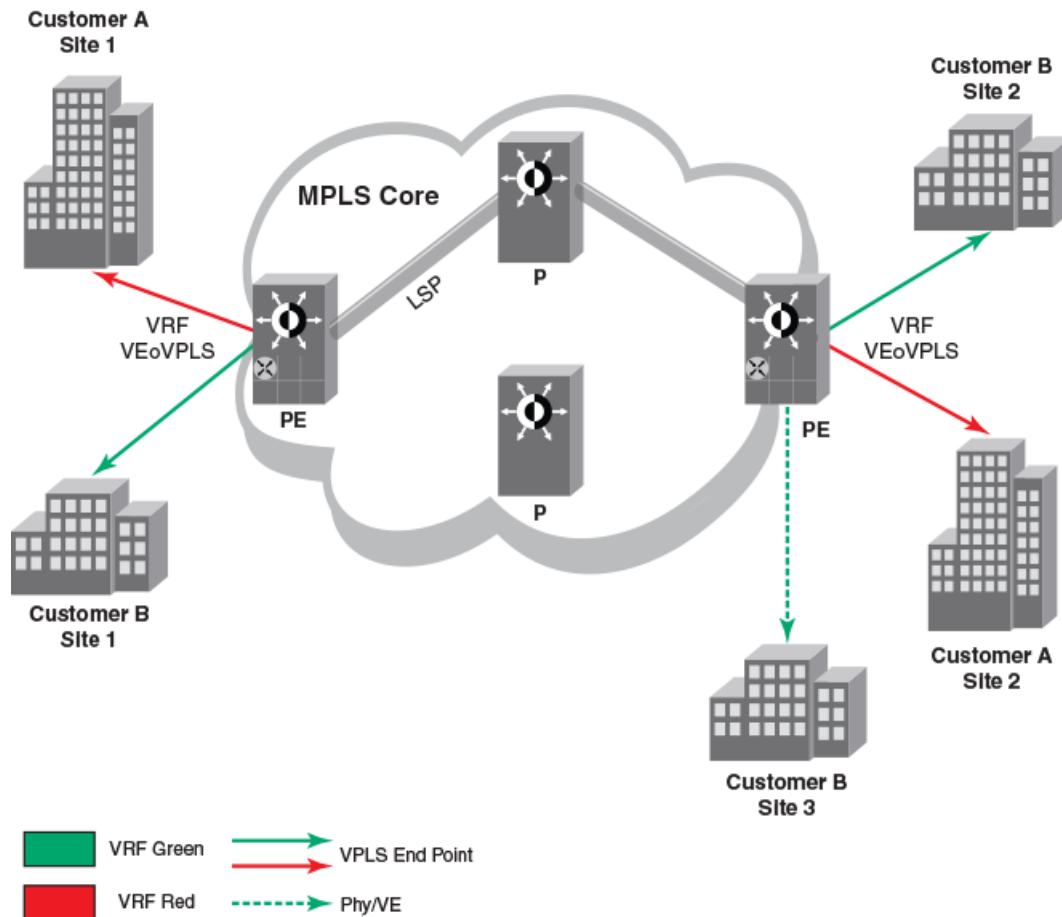
– Solid Grid: Inbound ACL to filter traffic incoming to PE1 VEoVPLS interface 10.1.1.1

– Blurred Grid: Outbound ACL to filter traffic outgoing from PE3 VEoVPLS interface 10.1.1.3

– 10.1.1.1, 10.2.2.2, 10.3.3.3, 10.11.11.11 are loopback addresses of PE1, PE2, PE3 and P nodes.

ACL support for VE over VPLS configuration considerations

- ACLs applied on the VPLS-VE interface is effective to inbound and outbound traffic received from or sent to local end-points. The MPLS uplink (VPLS Peer) inbound and outbound traffic is not filtered by the ACL.
- The ACLs having VLAN ID in their rule can not be applied to VE over VPLS interfaces.
- VPLS-VE and ACL definition modifications require explicit rebinding to take effect.

Configuration Considerations

Consider the following when configuring VE over VPLS ACLs.

- ACLs applied on the VPLS-VE interface is effective to inbound and outbound traffic received from or sent to local end-points. The MPLS uplink (VPLS Peer) inbound and outbound traffic is not filtered by the ACL.

- The ACLs having VLAN ID in their rule can not be applied to VE over VPLS interfaces.

- VPLS-VE and ACL definition modifications require explicit rebinding to take effect.

IPv6 ACL on VE over VPLS

IPv6 ACL support for VE over VPLS

IPv6 ACL support allows the user to attach an IPv6 filter to a Virtual Ethernet (VE) interface defined over a Virtual Private LAN Service (VPLS) bridge. These filters can match ingress and egress IPv6 traffic on all local endpoints of the VPLS bridge.

An IPv6 ACL applied to VE interface defined on a VPLS bridge maps to all physical ports in the VPLS bridge. It is not used for traffic coming from or going to, remote endpoints through LSPs.

IPv6 ACL on VE over VPLS considerations

- There is an existing limitation of automatic rebinding of all ACL rules whenever applying or removing an ACL on an interface (physical or VLAN-based VE) or when adding or deleting ports to and from VLAN-based VE interfaces. The same limitations extend to VE over VPLS interfaces.

- The user must issue an explicit **ipv6 rebind-acl-all** command at the global configuration level whenever the access list definition changes. Without this command, the old definition remains, and the modified command does not take effect at the CAM level. This is true for VE over VPLS interfaces.

- ACL applies only to inbound and outbound traffic received from or sent to local endpoints. The ACL does not filter the MPLS uplink (VPLS peer) inbound and outbound traffic. When a user issues a CLI command to configure the ACL on a VE over VPLS interface, a message appears on the console. If inbound ACL the following message is displayed:

      Inbound ACL will be applied to all local endpoints of VE over VPLS interface

  If outbound ACL the following message is displayed:

      Outbound ACL will be applied to all local endpoints of VE over VPLS interface

- The ACLs having a VLAN ID in their rule are not applicable to VE over VPLS interfaces.

VLAN overlapping between VLAN based VE and VE over VPLS

When creating a VPLS endpoint, if the VPLS endpoint has a VLAN ID overlapping with VLAN ID of VLAN-based VE interface having ACL configured over it, it implicitly issues a request to rebind all ACLs so that a default ACL rule to permit all traffic coming from VPLS endpoints ports with overlapping VLAN can be programmed.

In MLX, it is just for the outbound direction. The inbound direction resolves with the port mask available in inbound PRAM.

MLX: Outbound - port can have either L2 or L3 ACL.

Each ACL rule will be replicated on a per port per VLAN basis.

Following table summarizes the need for default permit ACL rule in different scenarios:

TABLE 94 MLX outbound need of default permit ACL rules

L2 ACL on Port	L3 ACL on VPLS-VE	L3 ACL on VLAN-VE	Program Implicit Permit ACL
No	No	Yes	Yes
Yes	No	X	No
No	Yes	X	No
X	X	No	No

X: MLX does not care whether ACL is programmed.

IPv6 ACL on VE over VPLS considerations

Configuration considerations

- There is an existing limitation of automatic rebinding of all ACL rules whenever applying or removing an ACL on an interface (physical or VLAN-based VE) or when adding or deleting ports to and from VLAN-based VE interfaces. The same limitations extend to VE over VPLS interfaces.
- The user must issue an explicit **ipv6 rebind-acl-all** command at the global configuration level whenever the access list definition changes. Without this command, the old definition remains, and the modified command does not take effect at the CAM level. This is true for VE over VPLS interfaces.
- ACL applies only to inbound and outbound traffic received from or sent to local endpoints. The ACL does not filter the MPLS uplink (VPLS peer) inbound and outbound traffic. When a user issues a CLI command to configure the ACL on a VE over VPLS interface, a message appears on the console. If inbound ACL the following message is displayed:

  ```
  Inbound ACL will be applied to all local endpoints of VE over VPLS interface
  ```

  If outbound ACL the following message is displayed:

  ```
  Outbound ACL will be applied to all local endpoints of VE over VPLS interface
  ```

- The ACLs having a VLAN ID in their rule are not applicable to VE over VPLS interfaces.

Upgrade and downgrade considerations

VE over VPLS does not support a hitless upgrade; IPv6 ACL on VE over VPLS also does not support it.

If there is any VLAN overlapping of VPLS instance (or VE over VPLS interface) with VLAN-based VE having ACL configured, there is packet leak.

VRF aware ACL over VEoVPLS

VRF aware ACL over VPLS introduction

VRF support for Virtual Ethernet (VE) over Virtual Private LAN Services (VPLS) uses ACLs to filter VPLS traffic.

The VE over VPLS (also known as routing over VPLS) feature combines the routing functionality of a VE interface with VPLS endpoints. VRF support applies to all the functionality that VEoVPLS supports.

Specifications

- ACL enables the user to filter traffic based on the information in the IP packet header.
- Helps the user to filter the traffic by applying ACL on VE belonging to non-default VRF over VPLS.

Configuration of VRF aware ACL over VEoVPLS

- Create a VRF
- Create required access list(s)
- Create VE over VPLS interface
- Apply the VRF and inbound/outbound ACL on the VE interface

VRF aware ACL over VEoVLPS configuration examples

User creates IN and OUT ACL on non-default VRF over VE over VPLS interface

1. Create access list(s).

```
access-list 121 permit tcp any host 100.0.0.2
access-list 121 permit tcp any host 300.0.0.2
access-list 131 permit udp any host 200.0.0.100
```

2. Create VE over VPLS interface.

```
vpls a 1
    router-interface ve 3
    vlan 10
        tagged ethernet 3/1 to 3/4
```

3. Create VRF.

```
vrf red
    rd 55:55
    route-target export 5:5
    route-target import 5:5
    address-family ipv4
        ip route 10.10.9.0/24 210.1.1.19
    exit-address-family
exit-vrf
```

4. Apply VRF and inbound/outbound ACL on VE interface.

```
interface ve 3
    vrf forwarding red
 ip access-group 121 in
 ip access-group 131 out
```

CAM usage:

A total of eight inbound and four outbound CAM entries are consumed, excluding any implicit deny clauses.

User creates IN and OUT ACL on VE over VPLS interface having LAG port.

1. Create access list(s).

```
access-list 151 permit tcp any host 100.0.0.2
access-list 161 permit tcp any host 300.0.0.2
access-list 161 permit udp any host 200.0.0.100
```

2. Create LAG.

```
lag "lag-10" static id 10
 ports ethernet 3/2 to 3/3
 primary-port 3/2
 deploy
 !
```

3. Create VE over VPLS

```
vpls vpls-10 10
     router-interface ve 10
     vlan 10
          tagged ethe 3/2 to 3/3
```

4. Create VRF

```
vrf red
    rd 55:55
    route-target export 5:5
    route-target import 5:5
    address-family ipv4
        ip route 10.10.9.0/24 210.1.1.19
    exit-address-family
exit-vrf
```

5. Apply VRF and ACL on VE.

```
interface ve 10
 vrf forwarding red
      ip access-group 151 in
 ip access-group 161 out
```

CAM usage:

A total of two inbound and four outbound CAM entries are consumed, excluding any implicit deny clauses.

VRF support for VE over VPLS

Summary of functionalities

The following table discusses supported and not supported functions for VRF VEoVPLS.

Function	Comment
IP Routing	Only unicast routing is supported. Multicast routing is not supported.
ARP/Static ARP	Supported.
Multi-port Static ARP	Not Supported.
Local-proxy-ARP	Not Supported.
Proxy ARP	Supported.
ARP - DAI	Not Supported.
Traceroute	Supported.
ICMP	Supported.
ICMP Redirect Message	Supported.
ICMP Unreachable Message	Not Supported.
OSPF	Supported.
IS-IS	Not Supported.

| Function	Comment
RIP	Supported.
BGP	Supported.
Trunk Ports (LAG)	Supported.
L2 Multicast	Supported.
IGMP Snooping	Supported.
L2 ACL	Supported.
L3 ACL	Supported.
Rate Limiting	Supported.
PBR	Not Supported.
Multi-netting	Supported.
VRRP/VRRP-E	Supported.
IP Helper Address/ BootP	Supported.
ECMP	Supported.
IP MTU*	Not supported. (Can not be enforced by way of MPLS core.)
VEoVPLS as MPLS interface	Not supported. The VE configured on VPLS cannot be made an MPLS interface.
Dual Tag Mode	Not Supported.
L3 Multicast (PIM)	Not Supported.
BFD	Not Supported.
GRE	Not Supported.
RPF	Not Supported.
IPv6	Supported.
PBB	Not Supported.
MCT	Not Supported.
IP Unnumbered	Not Supported.
IRDP	Not Supported.
Route-only	Not Supported.
DHCP	Supported.

* IP MTU: IP MTU is ignored for L3 packets whose next hop is over the VPLS local endpoint or VPLS remote endpoint no matter from which interface the packet arrives.

VRF support for VE over VPLS

The VE over VPLS (in other words "routing over VPLS") feature combines the routing functionality of a VE interface with VPLS endpoints.

Specifications

The VE over VPLS feature combines routing functionality of a VE interface with VPLS endpoints. This feature is extended to support at the VRF.
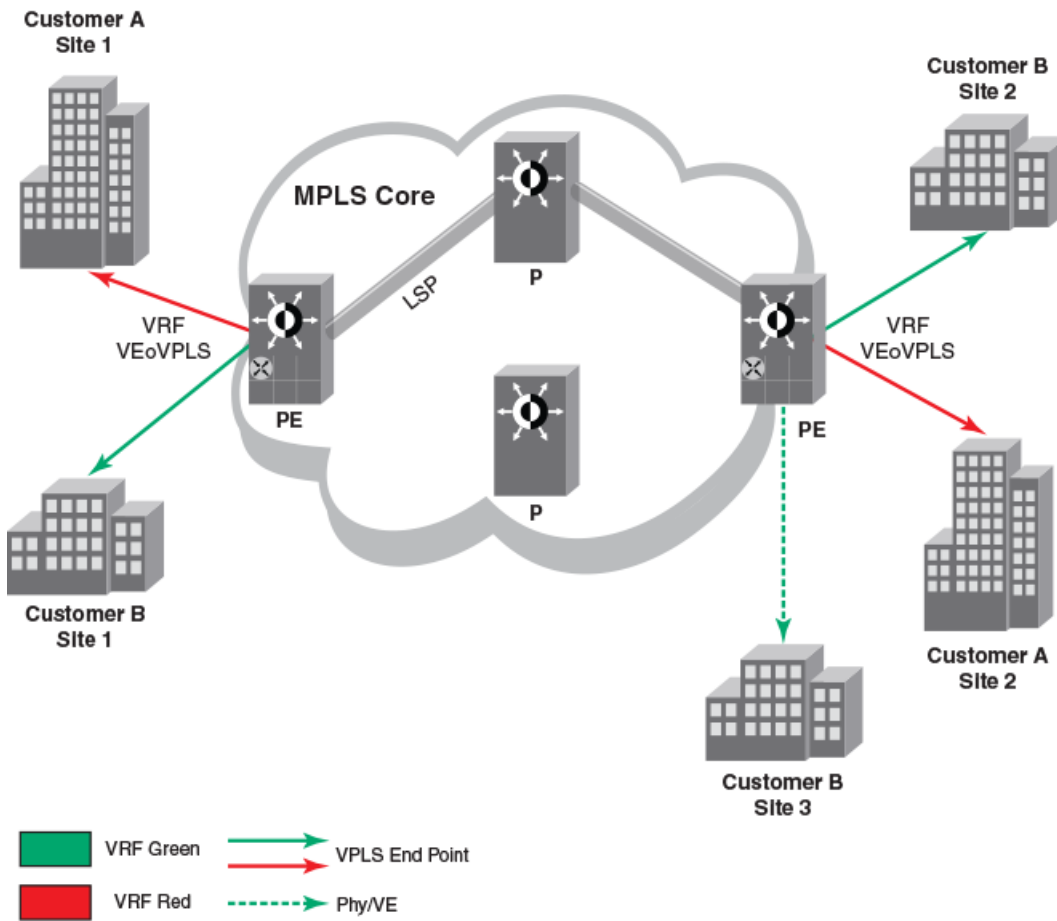
By allowing VRF configured on a VPLS VE, VRF routing can be achieved when the packets arrive on the VPLS endpoint before entering the VPLS uplink. The VRF VEoVPLS feature routes packets between the VPLS VE interface and all other IP interfaces outside of VPLS domain which resides on the same VRF.

This VRF VEoVPLS feature is supported on line cards for XMR/MLX/MLXe and supports all existing functionality of VPLS.

Routing on Generation 1 and Generation 1.1 line cards

Generation 1 and Generation 1.1 line card routing requirements foe VRF support for VE over VPLS.

Feature	Packet received from	Line cards		
		Darter	Generation 1 cards	Generation 1.1 cards
VEoVPLS	VPLS End point	Not supported. Packet will be dropped.	Supported.	Supported.
	VPLS Uplink	Not supported. Packet will be dropped.	Supported.	Supported.
VRF VEoVPLS	VPLS End point	Not supported. Packet will be dropped.	Supported.	Supported.
	VPLS Uplink	Not supported. Packet will be dropped.	Not supported. Packet will be dropped.	Not supported. Packet will be dropped.

Configuration steps

VE over VPLS uses the same ACL commands as VE for VLANs.

To configuring an ACL on VPLS-VE interface, complete the following steps.

1. Create the access-list.
2. Create the VE over VPLS interface.
3. Apply inbound and outbound ACL on VPLS-VE interface.

Sample configurations

Create an "IN" ACL on specific Ethernet port of a VE over VPLS interface.

Step 1:

```
ip access-list standard v4_acl
 permit tcp host 10.157.22.26 any eq telnet
```

Step 2:

```
vpls b 2
 router-interface ve 2
 vpls-peer 1.1.1.2
 vlan 500
  tagged ethe 4/1
 vlan 600
  tagged ethe 4/2
 vlan 700
  tagged ethe 4/2
```

Step 3:

```
interface ve 2
 ip access-group v4_acl in ethernet 4/2
```

Create an "IN" and "OUT" ACL condition on VE over VPLS interface.

Step 1:

```
access-list 121 permit tcp any host 10.0.0.2
access-list 121 permit tcp any host 10.0.0.2
access-list 131 permit udp any host 10.0.0.100
```

Step 2:

```
vpls a 1
 router-interface ve 3
 vlan 10
  tagged ethernet 3/1 to 3/4
```

Step 3:

```
interface ve 3
 ip access-group 121 in
 ip access-group 131 out
```

Error messages

The following messages are seen when an invalid configuration is attempted.

- • IN ACL – "Inbound ACL is applied to all local endpoints of VE over VPLS interface".
- • OUT ACL – "Outbound ACL is applied to all local endpoints of VE over VPLS interface".
- • This feature is not supported for 24x10G modules.
- • This feature is not supported for POS modules.

IPv4/IPv6 wildcard match

IPv4/IPv6 wildcard match

Describes the IPv4/IPv6 Wildcard Match feature.

Access Control Lists (ACLs) are configured using the keyword "access-list" and is the basic building block for many packet filtering based applications supported by MLX/XMR/CES product lines. It allows the users to define terms or rules that are matched with various fields in protocols headers of packets passing through theses boxes. ACL applications use the above ability of ACLs to implement various applications (PBR, Rate-limit , and so on) based on it. Source and Destination addresses of IPv4 and IPv6 packets are some of the most commonly matched fields in a packet. To do the address matching, the ACL expects two obvious parameters, such as value and position of bits to match. As the traditional usage of address matching is to find out whether addresses fall under a particular subnet, the corresponding CLIs are designed to take *subnet address*/*prefix length* as input. **Example: 192.168.1.0/24, ABCD::/64.**

This feature allows ACLs to match an **arbitrary** set of bits in source or a destination address of IPv4 and IPv6 packets. The user can specify the bits to be matched in the form of an explicit bit mask that replaces prefix length in the ACL configuration.

Requirements

Specific IPv6 address spaces such that specific fields are used or known, and they are requested to be masked out.

Add a new method for specifying source or destination prefixes in IPv6 ACL configuration CLIs that allows the user to specify an explicit mask. The explicit mask represents any arbitrary combination of 128 bits and system will accept. An IPv4 or IPv6 arbitrary mask is represented in the wildcard mask format. The ACL will match all the bit positions corresponding to unset (zero) bits in the mask (matching is done between source or destination addresses are given in the ACL term and the packet being processed).

TABLE 95 ABCD::/16 is represented using wildcard mask as ABCD:: ::FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

ABCD::	1010 1011 1100 1101
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
FFFF::	0000 0000 0000 0000
	1111 1111 1111 1111

TABLE 95 ABCD::/16 is represented using wildcard mask as ABCD:: ::FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF (continued)

	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111

In the example above, when the user wants to match the last 16 bits to 1234 as well, it cannot be done using the prefix length. With explicit mask, the user represents it using address mask as ABCD::1234 ::FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:0

ABCD::	1010 1011 1100 1101
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0000 0000 0000 0000
	0001 0010 0011 0100
FFFF::	0000 0000 0000 0000
	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111
	1111 1111 1111 1111
	0000 0000 0000 0000

IPv4 wildcard mask

Explains the wildcard mask format for IPv4.

For IPv4, a CLI keyword to specify mask is *already present.*

```
device(config-ext-nacl-temp)# permit ip ?
  A.B.C.D or A.B.C.D/L   IP address/Subnet mask length
  any                    Any source host
  host                   A single source host
device(config-ext-nacl-temp)# permit ip 192.168.0.1 ?
  A.B.C.D   Wildcard bits
device(config-ext-nacl-temp)# permit ip 192.168.0.1 0.0.3.0 ?
  A.B.C.D or A.B.C.D/L   IP address/Subnet mask length
  any                    Any destination host
  host                   A single destination host
device(config-ext-nacl-temp)# permit ip 192.168.0.1 0.0.3.0 192.168.4.1 ?
  A.B.C.D   Wildcard bits
```

IPv6 wildcard mask

Describes IPv6 wildcard mask configurations.

For IPv4, a CLI keyword to specify mask is **already present**. IPv6 is designed to follow the same bit format as IPv4 Wildcard Mask, that is; it matches the bits corresponding to '0's and the bits corresponding to '1' are ignored. The keyword to specify the mask is added to all the places in ACL configuration template where the IPv6 address prefix keyword was present. This feature empowers the user to configure any arbitrary combination of 128 bits as source or destination mask in the IPv6 ACLs. Masks must be specified in IPv6 address string format (X:X::X:X).

```
device(config-ipv6-access-list temp)# permit ipv6 ?
  X:X::X:X/M       IPv6 source prefix
  X:X::X:X         IPv6 source address
  any              Any source host
  host             A single source host
device(config-ipv6-access-list temp)# permit ipv6 1::1 ?
  X:X::X:X    IPv6 source mask
device(config-ipv6-access-list temp)# permit ipv6 1::1 ::ffff:0 ?
  X:X::X:X/M       IPv6 destination prefix
  X:X::X:X         IPv6 destination address
  any              Any destination host
  host             A single destination host
device(config-ipv6-access-list temp)# permit ipv6 1::1 ::ffff:0 2::2 ?
  X:X::X:X    IPv6 destination mask
```

To protect existing IPv6 deployment and scripts, follow **what you configure is what we display approach** . If prefix length is configured, the prefix length is displayed. If wildcard mask is configured, the wildcard mask is displayed. Note in an example given below.

```
device# show ipv6 access-list wildcard1
ipv6 access-list wildcard1: 6 entries
 10: permit ipv6 2::1 ::ffff:0 4::1 ::ffff:0
 20: permit ipv6 3::/64 6::/72
 30: permit ipv6 any 4::1230 ::ffff:0
 40: permit ipv6 4::1 ::fff0 any
 50: permit ipv6 host 7::7 1:2:3:4:5:6:7:8 ::fff0
 60: permit ipv6 1:2:3:4:5:6:7:8 ::fff0 host 7::7
```

IPv4/IPv6 wildcard mask customer configurations

Describes customer configurations for the IPv4 and IPv6 wildcard.

The configuration below illustrates ACL compression by IPv4 wildcard.

```
device(config-ext-nacl-expand)# show access name expand
Extended IP access list expand : 16 entries
10: permit ip host 192.168.0.1 host 192.168.4.1
20: permit ip host 192.168.0.1 host 192.168.5.1
30: permit ip host 192.168.0.1 host 192.168.6.1
40: permit ip host 192.168.0.1 host 192.168.7.1
50: permit ip host 192.168.1.1 host 192.168.4.1
60: permit ip host 192.168.1.1 host 192.168.5.1
70: permit ip host 192.168.1.1 host 192.168.6.1
80: permit ip host 192.168.1.1 host 192.168.7.1
90: permit ip host 192.168.2.1 host 192.168.4.1
100: permit ip host 192.168.2.1 host 192.168.5.1
110: permit ip host 192.168.2.1 host 192.168.6.1
120: permit ip host 192.168.2.1 host 192.168.7.1
130: permit ip host 192.168.3.1 host 192.168.4.1
140: permit ip host 192.168.3.1 host 192.168.5.1
150: permit ip host 192.168.3.1 host 192.168.6.1
160: permit ip host 192.168.3.1 host 192.168.7.1
device(config-ext-nacl-temp)# show access-list name compress
Extended IP access list compress : 1 entries
10: permit ip 192.168.0.1 0.0.3.0 192.168.4.1 0.0.3.0
```

The configuration below illustrates ACL compression by IPv6 address mask.

```
device#show ipv6 access-list expand
ipv6 access-list expand: 4 entries
 10: permit udp any host 2000::1 eq radius
 20: permit udp any host 2001::1 eq radius
 30: permit udp any host 2002::1 eq radius
 40: permit udp any host 2003::1 eq radius
device#show ipv6 access-list compress
ipv6 access-list compress: 1 entries
 10: permit udp any 2000::1 3:: eq radius
```

All IPv6 access-list statements that support IPv6 prefix also support the address mask. Review the list below of the supported statements.

```
device# show ipv6 access-list temp
ipv6 access-list temp: 14 entries
 10: permit ahp 1::1 ::f:0 2::2 ::f:0
 20: permit esp 1::1 ::f:0 2::2 ::f:0
 30: permit icmp 1::1 ::f:0 2::2 ::f:0
 40: permit ipv6 1::1 ::f:0 2::2 ::f:0
 50: permit sctp 1::1 ::f:0 2::2 ::f:0
 60: permit tcp 1::1 ::f:0 2::2 ::f:0
 70: permit udp 1::1 ::f:0 2::2 ::f:0
 80: permit vlan 1 ahp 1::1 ::f:0 2::2 ::f:0
 90: permit vlan 1 esp 1::1 ::f:0 2::2 ::f:0
 100: permit vlan 1 icmp 1::1 ::f:0 2::2 ::f:0
 110: permit vlan 1 ipv6 1::1 ::f:0 2::2 ::f:0
 120: permit vlan 1 sctp 1::1 ::f:0 2::2 ::f:0
 130: permit vlan 1 tcp 1::1 ::f:0 2::2 ::f:0
 140: permit vlan 1 udp 1::1 ::f:0 2::2 ::f:0
```