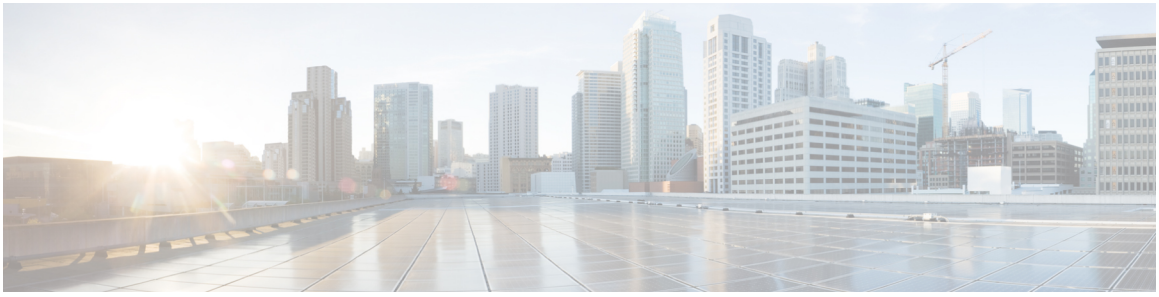
MPLS Configuration Guide, Cisco IOS XE 17.x

First Published: 2022-11-05

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

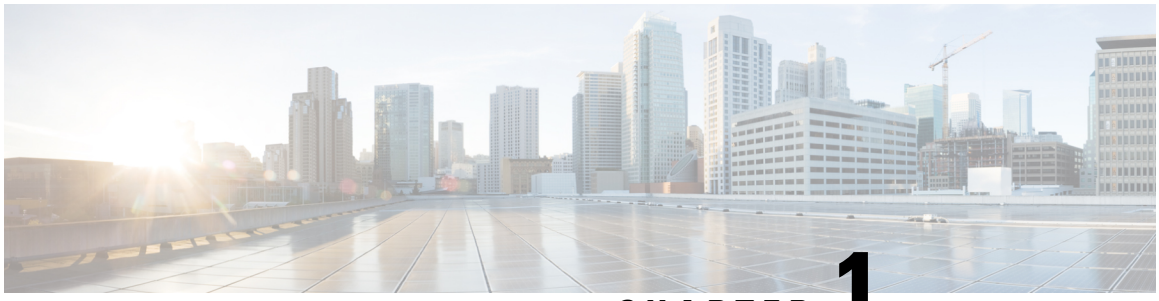
Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

PART I

MPLS Introduction



CHAPTER 1

MPLS Transport Profile

Multiprotocol Label Switching (MPLS) Transport Profile (TP) enables you to create tunnels that provide the transport network service layer over which IP and MPLS traffic traverses. MPLS-TP tunnels enable a transition from Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) time-division multiplexing (TDM) technologies to packet switching to support services with high bandwidth requirements, such as video.

Reference the Chapter Map here

Restrictions for MPLS Transport Profile

- Multiprotocol Label Switching Transport Profile (MPLS-TP) penultimate hop popping is not supported. Only ultimate hop popping is supported, because label mappings are configured at the MPLS-TP endpoints.
- Ethernet subinterfaces are not supported.
- IPv6 addressing is not supported.

L2VPN Restrictions

- Layer 2 Virtual Private Network (L2VPN) interworking is not supported.
- Local switching with Any Transport over MPLS (AToM) pseudowire as a backup is not supported.
- L2VPN pseudowire redundancy to an AToM pseudowire by one or more attachment circuits is not supported.
- Pseudowire ID Forward Equivalence Class (FEC) type 128 is supported, but generalized ID FEC type 129 is not supported.

- Static pseudowire Operations, Administration, and Maintenance (OAM) protocol and BFD VCCV attachment circuit (AC) status signaling are mutually exclusive protocols. Bidirectional Forwarding Detection (BFD) and Virtual Circuit Connectivity Verification (VCCV) in failure detection mode can be used with Static Pseudowire OAM protocol.
- BFD VCCV AC status signaling cannot be used in pseudowire redundancy configurations. You can use Static Pseudowire OAM instead.

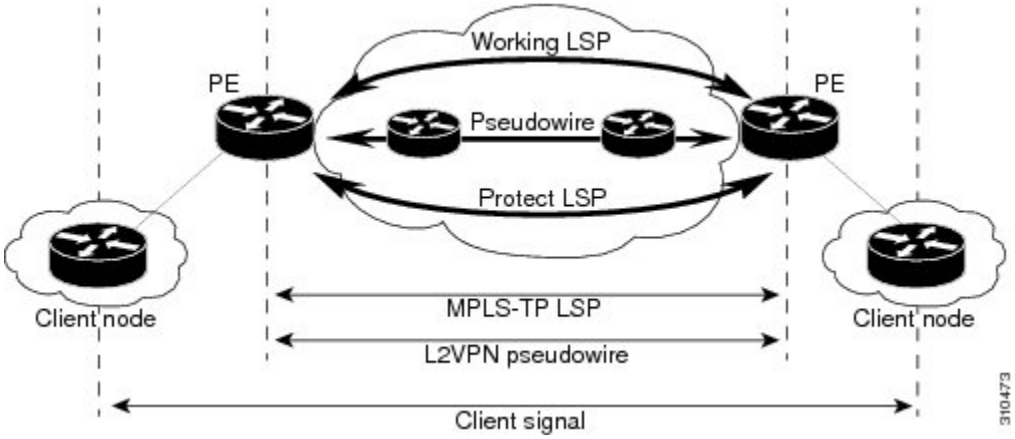
Ping and Trace Restrictions

- Ping for static pseudowires over MPLS-TP tunnels is not supported.
- Pseudowire ping and traceroute functionality for multisegment pseudowires that have one or more static pseudowire segments is not supported.
- The following packet format is supported:
  - A labeled packet with Generic Associated Channel Label (GAL) at the bottom of the label stack.
  - ACH channel is IP (0x21).
  - RFC-4379-based IP, UDP packet payload with valid source.
  - Destination IP address and UDP port 3503.
- Default reply mode for (1) is 4—Reply via application level control channel is supported. An echo reply consists of the following elements:
  - A labeled packet with a GAL label at the bottom of the label stack.
  - Associated Channel (ACh) is IP (0x21).
  - RFC-4379-based IP, UDP packet payload with valid source.
  - Destination IP address and UDP port 3503.
- The optional “do not reply” mode may be set.
- The following reply modes are not allowed and are disabled in CLI:
  - 2—Reply via an IPv4/IPv6 UDP packet
  - 3—Reply via an IPv4/IPv6 UDP packet with router alert
- Force-explicit-null is not supported with ping and trace.
- Optional Reverse Path Connectivity verification is not supported.

Information About MPLS-TP

How MPLS Transport Profile Works

Multiprotocol Label Switching Transport Profile (MPLS-TP) tunnels provide the transport network service layer over which IP and MPLS traffic traverses. MPLS-TP tunnels help transition from Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) and Time Division Multiplexing (TDM) technologies to packet switching to support services with high bandwidth utilization and lower cost. Transport networks are connection-oriented, statically provisioned, and have long-lived connections. Transport networks usually avoid control protocols that change identifiers (like labels). MPLS-TP tunnels provide this functionality through statically provisioned bidirectional label switched paths (LSPs), as shown in the figure below.



MPLS-TP Path Protection

MPLS-TP label switched paths (LSPs) support 1-to-1 path protection. There are two types of LSPs: protect LSPs and working LSPs. You can configure the both types of LSPs when configuring the MPLS-TP tunnel. The working LSP is the primary LSP used to route traffic. The protect LSP acts as a backup for a working LSP. If the working LSP fails, traffic is switched to the protect LSP until the working LSP is restored, at which time forwarding reverts back to the working LSP.

Bidirectional LSPs

Multiprotocol Label Switching Transport Profile (MPLS-TP) label switched paths (LSPs) are bidirectional and co-routed. They comprise of two unidirectional LSPs that are supported by the MPLS forwarding infrastructure. A TP tunnel consists of a pair of unidirectional tunnels that provide a bidirectional LSP. Each unidirectional tunnel can be optionally protected with a protect LSP that activates automatically upon failure conditions.

Support for MPLS Transport Profile OAM

Several Operations, Administration, and Maintenance (OAM) protocols and messages support the provisioning and maintenance of Multiprotocol Label Switching Transport Profile (MPLS-TP) tunnels and bidirectional label switched paths (LSPs).

The following OAM messages are forwarded along the specified MPLS LSP:

- OAM Fault Management—Alarm Indication Signal (AIS), Link Down Indication (LDI), and Lock Report (LKR) messages (GAL with BFD messages).
- OAM Connection Verification—Ping and traceroute messages (GAL with IP channel by default).
- OAM Continuity Check—Bidirectional Forwarding Detection (BFD) messages—non-IP BFD and IP BFD (GAL with non-IP BFD channel or IP BFD channel depending on message format).
- The following messages are forwarded along the specified pseudowire:
    - Static pseudowire OAM messages
    - Pseudowire ping and traceroute messages
    - BFD messages

- MPLS-TP OAM Fault Management (LDI, AIS, and LKR messages)—LDI messages are AIS messages whose L-flags are set. The LDI messages are generated at midpoint nodes when a failure is detected. From the midpoint, an LDI message is sent to the endpoint that is reachable with the existing failure. Similarly, LKR messages are sent from a midpoint node to the reachable endpoint when an interface is administratively shut down. By default, the reception of LDI and LKR messages on the active LSP at an endpoint will cause a path protection switchover, whereas the reception of an AIS message will not.
- MPLS-TP OAM Fault Management with Emulated Protection Switching for LSP Lockout—Cisco implements a form of Emulated Protection Switching to support LSP Lockout using customized Fault messages. When a Lockout message is sent, it does not cause the LSP to be administratively down. The Cisco Lockout message causes a path protection switchover and prevents data traffic from using the LSP. The LSP remains administratively up so that BFD and other OAM messages can continue to traverse it and so that maintenance of the LSP can take place (such as reconfiguring or replacing a midpoint LSR). After OAM verifies the LSP connectivity, the Lockout is removed and the LSP is brought back to service. Lockout of the working LSP is not allowed if a protect LSP is not configured. Conversely, the Lockout of a protect LSP is allowed if a working LSP is not configured.
- LSP ping and trace—To verify MPLS-TP connectivity, use the **ping mpls tp** and **trace mpls tp** commands. You can specify that echo requests be sent along the working LSP, the protect LSP, or the active LSP. You can also specify that echo requests be sent on a locked-out MPLS-TP tunnel LSP (either working or protected) if the working or protected LSP is explicitly specified. You can also specify ping/trace messages with or without IP.
- MPLS-TP OAM Continuity Check (CC) via BFD and Remote Defect Indication (RDI)—RDI is communicated via the BFD diagnostic field in BFD CC messages. BFD sessions run on both the working LSP and the protect LSP. To perform a path protection switchover within 60 milliseconds on an MPLS-TP endpoint, use the BFD Hardware Offload feature, which enables the router hardware to construct and send BFD messages, removing the task from the software path. The BFD Hardware Offload feature is enabled automatically on supported platforms.

MPLS-TP OAM GACH—Generic Associated Channel (G-ACh) is the control channel mechanism associated with Multiprotocol Label Switching (MPLS) LSPs in addition to MPLS pseudowire. The G-ACh Label (GAL) (Label 13) is a generic alert label to identify the presence of the G-ACh in the label packet. It is taken from the reserved MPLS label space. G-ACh/GAL supports OAMs of LSPs and in-band OAMs of pseudowires (PWs). OAM messages are used for fault management, connection verification, continuity check, and so on.

MPLS Transport Profile Static and Dynamic Multisegment Pseudowires

Multiprotocol Label Switching Transport Profile (MPLS-TP) supports the following combinations of static and dynamic multisegment pseudowires:

- Dynamic-static
- Static-dynamic
- Static-static

MPLS-TP OAM Status for Static and Dynamic Multisegment Pseudowires

With static pseudowires, status notifications can be provided by BFD over VCCV or by the static pseudowire OAM protocol. However, BFD over VCCV sends only attachment circuit status code notifications. Hop-by-hop notifications of other pseudowire status codes are not supported. Therefore, the static pseudowire OAM

protocol is preferred. You can acquire per pseudowire OAM for attachment circuit/pseudowire notification over the VCCV channel with or without the control word.

MPLS Transport Profile Links and Physical Interfaces

Multiprotocol Label Switching Transport Profile (MPLS-TP) link numbers may be assigned to physical interfaces only. Bundled interfaces and virtual interfaces are not supported for MPLS-TP link numbers.

The MPLS-TP link creates a layer of indirection between the MPLS-TP tunnel and midpoint LSP configuration and the physical interface. The **mpls tp link** command is used to associate an MPLS-TP link number with a physical interface and next-hop node. On point-to-point interfaces or Ethernet interfaces designated as point-to-point using the **medium p2p** command, the next-hop can be implicit, so the **mpls tp link** command just associates a link number to the interface.

Multiple tunnels and LSPs may then refer to the MPLS-TP link to indicate that they are traversing that interface. You can move the MPLS-TP link from one interface to another without reconfiguring all the MPLS-TP tunnels and LSPs that refer to the link.

Link numbers must be unique on the router or node.

See the section Configuring MPLS-TP Links and Physical Interfaces, on page 17, for more information.

Tunnel Midpoints

Tunnel LSPs, whether endpoint or midpoint, use the same identifying information. However, it is entered differently.

- At the midpoint, all information for the LSP is specified with the **mpls tp lsp** command for configuring forward and reverse information for forwarding.
- At the midpoint, determining which end is source and which is destination is arbitrary. That is, if you are configuring a tunnel between your device and a coworker's device, then your device is the source. However, your coworker considers his or her device to be the source. At the midpoint, either device could be considered the source. At the midpoint, the forward direction is from source to destination, and the reverse direction is from destination to source.
- At the endpoint, the local information (source) either comes from the global device ID and global ID, or from the locally configured information using the **tp source** command.
- At the endpoint, the remote information (destination) is configured using the **tp destination** command after you enter the **interface tunnel-tp** *number* command. The **tp destination** command includes the destination node ID, and optionally the global ID and the destination tunnel number. If you do not specify the destination tunnel number, the source tunnel number is used.
- At the endpoint, the LSP number is configured in working-lsp or protect-lsp submode. The default is 0 for the working LSP and 1 for the protect LSP.
- When configuring LSPs at midpoint devices, ensure that the configuration does not deflect traffic back to the originating node.

How to Configure MPLS Transport Profile

Configuring the MPLS Label Range

You must specify a static range of Multiprotocol Label Switching (MPLS) labels using the **mpls label range** command with the **static** keyword.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label range** *minimum-value maximum-value* **static** *minimum-static-value maximum-static-value*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *minimum-value maximum-value* **static** *minimum-static-value maximum-static-value* **Example:** `Device(config)# mpls label range 1001 1003 static 10000 25000`	Specifies a static range of MPLS labels.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Router ID and Global ID

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **mpls tp**
4. **router-id** *node-id*
5. **global-id** *num*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls tp** **Example:** `Device(config)# mpls tp`	Enters MPLS-TP configuration mode, from which you can configure MPLS-TP parameters for the device.
Step 4	**router-id** *node-id* **Example:** `Device(config-mpls-tp)# router-id 10.10.10.10`	Specifies the default MPLS-TP router ID, which is used as the default source node ID for all MPLS-TP tunnels configured on the device.
Step 5	**global-id** *num* **Example:** `Device(config-mpls-tp)# global-id 1`	(Optional) Specifies the default global ID used for all endpoints and midpoints. • This command makes the router ID globally unique in a multiprovider tunnel. Otherwise, the router ID is only locally meaningful. • The global ID is an autonomous system number, which is a controlled number space by which providers can identify each other. • The router ID and global ID are also included in fault messages sent by devices from the tunnel midpoints to help isolate the location of faults.
Step 6	**end** **Example:** `Device(config-mpls-tp)# end`	Exits MPLS-TP configuration mode and returns to privileged EXEC mode.

Configuring Bidirectional Forwarding Detection Templates

The **bfd-template** command allows you to create a BFD template and enter BFD configuration mode. The template can be used to specify a set of BFD interval values. You invoke the template as part of the MPLS-TP tunnel. On platforms that support the BFD Hardware Offload feature and that can provide a 60-ms cutover for MPLS-TP tunnels, it is recommended to use the higher resolution timers in the BFD template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bfd-template single-hop** *template-name*
4. **interval** [**microseconds**] {**both** *time* | **min-tx** *time* **min-rx** *time*} [**multiplier** *multiplier-value*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bfd-template single-hop** *template-name* **Example:** `Device(config)# bfd-template single-hop mpls-bfd-1`	Creates a BFD template and enter BFD configuration mode.
Step 4	**interval** [**microseconds**] {**both** *time* | **min-tx** *time* **min-rx** *time*} [**multiplier** *multiplier-value*] **Example:** `Device(config-bfd)# interval min-tx 99 min-rx 99 multiplier 3`	Specifies a set of BFD interval values.
Step 5	**end** **Example:** `Device(config-bfd)# exit`	Exits BFD configuration mode and returns to privileged EXEC mode.

Configuring Pseudowire OAM Attributes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-static-oam class** *class-name*
4. **timeout refresh send** *seconds*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-static-oam class** *class-name* **Example:** `Device(config)# pseudowire-static-oam class oam-class1`	Creates a pseudowire OAM class and enters pseudowire OAM class configuration mode.
Step 4	**timeout refresh send** *seconds* **Example:** `Device(config-st-pw-oam-class)# timeout refresh send 20`	Specifies the OAM timeout refresh interval.
Step 5	**exit** **Example:** `Device(config-st-pw-oam-class)# exit`	Exits pseudowire OAM configuration mode and returns to privileged EXEC mode.

Configuring the Pseudowire Class

When you create a pseudowire class, you specify the parameters of the pseudowire, such as the use of the control word, preferred path, OAM class, and VCCV BFD template.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **pseudowire-class** *class-name*
4. **encapsulation mpls**
5. **control-word**
6. **protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*]
7. **preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *host-name*}} [**disable-fallback**]
8. **status protocol notification static** *class-name*
9. **vccv bfd template** *name* [**udp** | **raw-bfd**]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *class-name* **Example:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Device(config-pw-class)# control-word`	Enables the use of the control word.
Step 6	**protocol** {**l2tpv2**	**l2tpv3**	**none**} [*l2tp-class-name*] **Example:** `Device(config-pw-class)# protocol none`	Specifies the type of protocol.
Step 7	**preferred-path** {**interface tunnel** *tunnel-number*	**peer** {*ip-address*	*host-name*}} [**disable-fallback**] **Example:**	Specifies the tunnel to use as the preferred path.

	Command or Action	Purpose
	`Device(config-pw-class)# preferred-path interface tunnel-tp2`	
Step 8	**status protocol notification static** *class-name* **Example:** `Device(config-pw-class)# status protocol notification static oam-class1`	Specifies the OAM class to use.
Step 9	**vccv bfd template** *name* [**udp** | **raw-bfd**] **Example:** `Device(config-pw-class)# vccv bfd template bfd-temp1 raw-bfd`	Specifies the VCCV BFD template to use.
Step 10	**end** **Example:** `Device(config-pw-class)# end`	Exits pseudowire class configuration mode and returns to privileged EXEC mode.

Configuring the Pseudowire

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **xconnect** *peer-ip-address vc-id* {**encapsulation** {**l2tpv3** [**manual**] \| **mpls** [**manual**]} \| **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** \| **receive** \| **both**}]
5. **mpls label** *local-pseudowire-label remote-pseudowire-label*
6. **mpls control-word**
7. **backup delay** {*enable-delay-period* \| **never**} {*disable-delay-period* \| **never**}
8. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 1/0`	Specifies the interface and enters interface configuration mode.
Step 4	**xconnect** *peer-ip-address vc-id* {**encapsulation** {**l2tpv3** [**manual**] | **mpls** [**manual**]} | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Device(config-if)# xconnect 10.131.191.251 100 encapsulation mpls manual pw-class mpls-tp-class1`	Binds the attachment circuit to a pseudowire VC and enters xconnect interface configuration mode.
Step 5	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-if-xconn)# mpls label 100 150`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 6	**mpls control-word** **Example:** `Device(config-if-xconn)# no mpls control-word`	Specifies the control word.
Step 7	**backup delay** {*enable-delay-period* | **never**} {*disable-delay-period* | **never**} **Example:** `Device(config-if-xconn)# backup delay 0 never`	Specifies how long a backup pseudowire virtual circuit (VC) should wait before resuming operation after the primary pseudowire VC goes down.
Step 8	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*] **Example:** `Device(config-if-xconn)# backup peer 10.0.0.2 50`	Specifies a redundant peer for a pseudowire virtual circuit (VC).
Step 9	**end** **Example:** `Device(config)# end`	Exits xconn interface connection mode and returns to privileged EXEC mode.

Configuring the MPLS-TP Tunnel

On the endpoint devices, create an MPLS TP tunnel and configure its parameters. See the **interface tunnel-tp** command for information on the parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel-tp** *number*
4. **description** *tunnel-description*
5. **tp tunnel-name** *name*
6. **tp bandwidth** *num*
7. **tp source** *node-id* [*global-id num*]
8. **tp destination** *node-id* [**tunnel-tp** *num*[**global-id** *num*]]
9. **bfd** *bfd-template*
10. **working-lsp**
11. **in-label** *num*
12. **out-label** *num* **out-link** *num*
13. **exit**
14. **protect-lsp**
15. **in-label** *num*
16. **out-label** *num* **out-link** *num*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel-tp** *number* **Example:** `Device(config)# interface tunnel-tp`	Enters tunnel interface configuration mode. Tunnel numbers from 0 to 999 are supported.
Step 4	**description** *tunnel-description* **Example:** `Device(config-if)# description headend tunnel`	(Optional) Specifies a tunnel description.

	Command or Action	Purpose
Step 5	**tp tunnel-name** *name* **Example:** `Device(config-if)# tp tunnel-name tunnel 122`	Specifies the name of the MPLS-TP tunnel.
Step 6	**tp bandwidth** *num* **Example:** `Device(config-if)# tp bandwidth 10000`	Specifies the tunnel bandwidth.
Step 7	**tp source** *node-id* [*global-id num*] **Example:** `Device(config-if)# tp source 10.11.11.11 global-id 10`	(Optional) Specifies the tunnel source and endpoint.
Step 8	**tp destination** *node-id* [**tunnel-tp** *num*[**global-id** *num*]]	Specifies the destination node of the tunnel.
	Example: `Device(config-if)# tp destination 10.10.10.10`	
Step 9	**bfd** *bfd-template* **Example:** `Device(config-if)# bfd mpls-tp-bfd-2`	Specifies the BFD template.
Step 10	**working-lsp** **Example:** `Device(config-if)# working-lsp`	Specifies a working LSP, also known as the primary LSP.
Step 11	**in-label** *num* **Example:** `Device(config-if-working)# in-label 111`	Specifies the in-label number.
Step 12	**out-label** *num* **out-link** *num* **Example:** `Device(config-if-working)# out-label 112 out-link`	Specifies the out-label number and out-link.
Step 13	**exit** **Example:** `Device(config-if-working)# exit`	Exits working LSP interface configuration mode and returns to interface configuration mode.

	Command or Action	Purpose
Step 14	**protect-lsp** **Example:** `Device(config-if)# protect-lsp`	Specifies a backup for a working LSP.
Step 15	**in-label** *num* **Example:** `Device(config-if-protect)# in-label 100`	Specifies the in label.
Step 16	**out-label** *num* **out-link** *num* **Example:** `Device(config-if-protect)# out-label 113 out-link`	Specifies the out label and out link.
Step 17	**end** **Example:** `Device(config-if-protect)# end`	Exits the interface configuration mode and returns to privileged EXEC mode.

Configuring MPLS-TP LSPs at Midpoints

Note When configuring LSPs at midpoint devices, ensure that the configuration does not deflect traffic back to the originating node.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls tp lsp source** *node-id* [**global-id** *num*] **tunnel-tp** *num* **lsp** {*lsp-num* | **protect** | **working**} **destination** *node-id* [**global-id** *num*] **tunnel-tp** *num*
4. **forward-lsp**
5. **bandwidth** *num*
6. **in-label** *num* **out-label** *num* **out-link** *num*
7. **exit**
8. **reverse-lsp**
9. **bandwidth** *num*
10. **in-label** *num* **out-label** *num* **out-link** *num*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls tp lsp source** *node-id* [**global-id** *num*] **tunnel-tp** *num* **lsp**{*lsp-num* | **protect** | **working**} **destination** *node-id* [**global-id** *num*] **tunnel-tp** *num* **Example:** `Device(config)# mpls tp lsp source 10.10.10.10 global-id 2 tunnel-tp 4 lsp protect destination 10.11.11.11 global-id 11 tunnel-tp 12`	Enables MPLS-TP midpoint connectivity and enters MPLS TP LSP configuration mode.
Step 4	**forward-lsp** **Example:** `Device(config-mpls-tp-lsp)# forward-lsp`	Enters MPLS-TP LSP forward LSP configuration mode.
Step 5	**bandwidth** *num* **Example:** `Device(config-mpls-tp-lsp-forw)# bandwidth 100`	Specifies the bandwidth.
Step 6	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Device(config-mpls-tp-lsp-forw)# in-label 53 out-label 43 out-link 41`	Specifies the in label, out label, and out link numbers.
Step 7	**exit** **Example:** `Device(config-mpls-tp-lsp-forw)# exit`	Exits MPLS-TP LSP forward LSP configuration mode.
Step 8	**reverse-lsp** **Example:** `Device(config-mpls-tp-lsp)# reverse-lsp`	Enters MPLS-TP LSP reverse LSP configuration mode.

	Command or Action	Purpose
Step 9	**bandwidth** *num* **Example:** `Device(config-mpls-tp-lsp-rev)# bandwidth 100`	Specifies the bandwidth.
Step 10	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Device(config-mpls-tp-lsp-rev)# in-label 33 out-label 23 out-link 44`	Specifies the in-label, out-label, and out-link numbers.
Step 11	**end** **Example:** `Device(config-mpls-tp-lsp-rev)# end`	Exits the MPLS TP LSP configuration mode and returns to privileged EXEC mode.

Configuring MPLS-TP Links and Physical Interfaces

MPLS-TP link numbers may be assigned to physical interfaces only. Bundled interfaces and virtual interfaces are not supported for MPLS-TP link numbers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **mpls tp link** *link-num* {**ipv4** *ip-address* | **tx-mac** *mac-address*} **rx-mac** *mac-address*
6. **ip rsvp bandwidth** [**rdm** [**bc0** *interface-bandwidth*] [[*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]]] [*interface-bandwidth* [*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]] | **mam max-reservable-bw** [*interface-bandwidth* [*single-flow-bandwidth*] [**bc0** *interface-bandwidth* [**bc1** *bandwidth*]]] | **percent** *percent-bandwidth* [*single-flow-bandwidth*]]
7. **end**
8. **show mpls tp link-numbers**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1/0`	Specifies the interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.10.10.10 255.255.255.0`	Assigns an IP address to the interface.
Step 5	**mpls tp link** *link-num* {**ipv4** *ip-address* | **tx-mac** *mac-address*} **rx-mac** *mac-address* **Example:** `Device(config-if)# mpls tp link 1 ipv4 10.0.0.2`	Associates an MPLS-TP link number with a physical interface and next-hop node. On point-to-point interfaces or Ethernet interfaces designated as point-to-point using the **medium p2p** command, the next-hop can be implicit, so the **mpls tp link** command just associates a link number to the interface. Multiple tunnels and LSPs can refer to the MPLS-TP link to indicate they are traversing that interface. You can move the MPLS-TP link from one interface to another without reconfiguring all the MPLS-TP tunnels and LSPs that refer to the link. Link numbers must be unique on the device or node.
Step 6	**ip rsvp bandwidth** [**rdm** [**bc0** *interface-bandwidth*] [[*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]]] [*interface-bandwidth* [*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]] | **mam max-reservable-bw** [*interface-bandwidth* [*single-flow-bandwidth*] [**bc0** *interface-bandwidth* [**bc1** *bandwidth*]]] | **percent** *percent-bandwidth* [*single-flow-bandwidth*]] **Example:** `Device(config-if)# ip rsvp bandwidth 1158 100`	Enables Resource Reservation Protocol (RSVP) bandwidth for IP on an interface. For the Cisco 7600 platform, if you configure non-zero bandwidth for the TP tunnel or at a midpoint LSP, make sure that the interface to which the output link is attached has enough bandwidth available. For example, if three tunnel LSPs run over link 1 and each LSP was assigned 1000 with the **tp bandwidth** command, the interface associated with link 1 needs bandwidth of 3000 with the **ip rsvp bandwidth** command.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 8	**show mpls tp link-numbers** **Example:** `Device# show mpls tp link-numbers`	Displays the configured links.

Configuring Static-to-Static Multisegment Pseudowires for MPLS-TP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **point-to-point**
4. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
5. **mpls label** *local-pseudowire-label remote-pseudowire-label*
6. **mpls control-word**
7. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
8. **mpls label** *local-pseudowire-label remote-pseudowire-label*
9. **mpls control-word**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *name* **point-to-point** **Example:** `Device(config)# l2 vfi atom point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 4	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC. Specify the IP address, the VC ID of the remote device, and the pseudowire class to use for the emulated VC. **Note** Only two **neighbor** commands are allowed for each Layer 2 VFI point-to-point command.
Step 5	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 101 201`	Configures the static pseudowire connection by defining local and remote circuit labels.

	Command or Action	Purpose
Step 6	**mpls control-word** **Example:** `Device(config-vfi)# mpls control-word`	Specifies the control word.
Step 7	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi)# neighbor 10.10.10.11 123 pw-class atom`	Sets up an emulated VC. Specify the IP address, the VC ID of the remote device, and the pseudowire class to use for the emulated VC.
Step 8	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 102 202`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 9	**mpls control-word** **Example:** **Example:** `Device(config-vfi)# mpls control-word`	Specifies the control word.
Step 10	**end** **Example:** `Device(config)# end`	Exits VFI configuration mode and returns to privileged EXEC mode.

Configuring a Template with Pseudowire Type-Length-Value Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-tlv template** *template-name*
4. **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-tlv template** *template-name* **Example:** `Device(config)# pseudowire-tlv template statictemp`	Creates a template of pseudowire type-length-value (TLV) parameters and enters pseudowire TLV template configuration mode.
Step 4	**tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value* **Example:** `Device(config-pw-tlv-template)# tlv statictemp 2 4 hexstr 1`	Specifies the TLV parameters.
Step 5	**end** **Example:** `Device(config-pw-tlv-template)# end`	Exits pseudowire TLV template configuration mode and returns to privileged EXEC mode.

Configuring Static-to-Dynamic Multisegment Pseudowires for MPLS-TP

When you configure static-to-dynamic pseudowires, you configure the static pseudowire class with the **protocol none** command, create a dynamic pseudowire class, and then invoke those pseudowire classes with the **neighbor** commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *class-name*
4. **encapsulation mpls**
5. **control-word**
6. **protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*]
7. **exit**
8. **pseudowire-class** *class-name*
9. **encapsulation mpls**
10. **exit**
11. **l2 vfi** *name* **point-to-point**
12. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
13. **neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
14. **mpls label** *local-pseudowire-label remote-pseudowire-label*

15. **mpls control-word**
16. **local interface** *pseudowire-type*
17. Do one of the following:
    - **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value*
    - **tlv template** *template-name*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *class-name* **Example:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Device(config-pw-class)# control-word`	Enables the use of the control word.
Step 6	**protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Device(config-pw-class)# protocol none`	Specifies the type of protocol. Use the **protocol none** command to specify a static pseudowire.
Step 7	**exit** **Example:** `Device(config-pw-class)# exit`	Exits pseudowire class configuration mode and returns to global configuration mode.
Step 8	**pseudowire-class** *class-name* **Example:**	Creates a pseudowire class and enters pseudowire class configuration mode.

	Command or Action	Purpose
	`Device(config)# pseudowire-class mpls-tp-class1`	
Step 9	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 10	**exit** **Example:** `Device(config-pw-class)# exit`	Exits pseudowire class configuration mode and returns to global configuration mode.
Step 11	**l2 vfi** *name* **point-to-point** **Example:** `Device(config)# l2 vfi atom point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 12	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC and enters VFI neighbor configuration mode. **Note** Note: Only two **neighbor** commands are allowed for each **l2 vfi point-to-point** command.
Step 13	**neighbor** *ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Device(config-vfi-neighbor)# neighbor 10.111.111.111 123 pw-class atom`	Sets up an emulated VC. **Note** Only two **neighbor** commands are allowed for each **l2 vfi point-to-point** command.
Step 14	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi-neighbor)# mpls label 101 201`	Configures the static pseudowire connection by defining local and remote circuit labels.
Step 15	**mpls control-word** **Example:** `Device(config-vfi-neighbor)# mpls control-word`	Specifies the control word.
Step 16	**local interface** *pseudowire-type* **Example:** `Device(config-vfi-neighbor)# local interface 4`	Specifies the pseudowire type.

	Command or Action	Purpose
Step 17	Do one of the following: • **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value* • **tlv template** *template-name* **Example:** `Device(config-vfi-neighbor)# tlv statictemp 2 4 hexstr 1`	Specifies the TLV parameters or invokes a previously configured TLV template.
Step 18	**end** **Example:** `Device(config-vfi-neighbor)# end`	Ends the session.

Verifying the MPLS-TP Configuration

Use the following commands to verify and help troubleshoot your MPLS-TP configuration:

- **debug mpls tp**—Enables the logging of MPLS-TP error messages.
- **logging (MPLS-TP)**—Displays configuration or state change logging messages.
- **show bfd neighbors mpls-tp**—Displays the BFD state, which must be up in order for the endpoint LSPs to be up.
- **show mpls l2transport static-oam l2transport static-oam**—Displays MPLS-TP messages related to pseudowires.
- **show mpls tp tunnel-tp** *number* **detail**—Displays the number and details of the tunnels that are not functioning.
- **show mpls tp tunnel-tp lsps**—Displays the status of the LSPs, and helps you ensure that both LSPs are up and working from a tunnel endpoint.
- **traceroute mpls tp** and **ping mpls tp**—Helps you identify connectivity issues along the MPLS-TP tunnel path.

Configuration Examples for MPLS Transport Profile

Example: Configuring Static-to-dynamic Multisegment Pseudowires for MPLS-TP

The following example shows how to configure static-to-dynamic multisegment pseudowires for Layer 2 VFI.

```
l2 vfi atom point-to-point (static-dynamic MSPW)
 neighbor 10.116.116.116 4294967295 pw-class dypw (dynamic)
  neighbor 10.111.111.111 123 pw-class stpw (static)
  mpls label 101 201
  mpls control-word
  local interface 4
  tlv mtu 1 4 1500
  tlv description 3 6 str abcd
  tlv descr C 4 hexstr 0505
```

Additional References for MPLS Transport Profile

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards and RFCs

Standard/RFC	Title
draft-ietf-mpls-tp-gach-gal-xx	*MPLS Generic Associated Channel*
RFC 5586	*MPLS Generic Associated Channel*
RFC 5885	*Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*
RFC 5921	*A Framework for MPLS in Transport Networks*

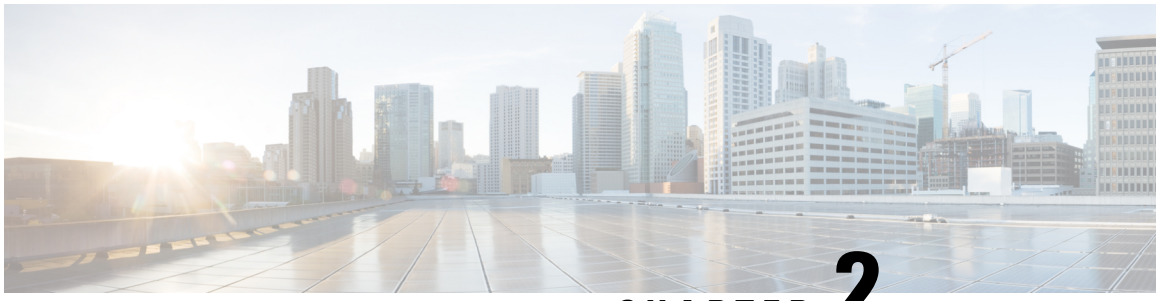
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Transport Profile

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 2

Multiprotocol Label Switching (MPLS) on Cisco Routers

This document describes commands for configuring and monitoring Multiprotocol Label Switching (MPLS) functionality on Cisco routers and switches. This document is a companion to other feature modules describing other MPLS applications.

Information About MPLS

MPLS Overview

Multiprotocol label switching (MPLS) combines the performance and capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing. MPLS enables service providers to meet the challenges of explosive growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure. The MPLS architecture is flexible and can be employed in any combination of Layer 2 technologies. MPLS support is offered for all Layer 3 protocols, and scaling is possible well beyond that typically offered in today's networks.

MPLS efficiently enables the delivery of IP services over an ATM switched network. MPLS supports the creation of different routes between a source and a destination on a purely router-based Internet backbone. By incorporating MPLS into their network architecture, service providers can save money, increase revenue and productivity, provide differentiated services, and gain competitive advantages.

Note In the Cisco IOS XE Release 16.x, the ASR 1000 routers only support fragmentation of the MPLS packets from the IP to MPLS direction.

Functional Description of MPLS

Label switching is a high-performance packet forwarding technology that integrates the performance and traffic management capabilities of data link layer (Layer 2) switching with the scalability, flexibility, and performance of network layer (Layer 3) routing.

Label Switching Functions

In conventional Layer 3 forwarding mechanisms, as a packet traverses the network, each router extracts all the information relevant to forwarding the packet from the Layer 3 header. This information is then used as an index for a routing table lookup to determine the next hop for the packet.

In the most common case, the only relevant field in the header is the destination address field, but in some cases, other header fields might also be relevant. As a result, the header analysis must be done independently at each router through which the packet passes. In addition, a complicated table lookup must also be done at each router.

In label switching, the analysis of the Layer 3 header is done only once. The Layer 3 header is then mapped into a fixed length, unstructured value called a *label* .

Many different headers can map to the same label, as long as those headers always result in the same choice of next hop. In effect, a label represents a *forwarding equivalence class* --that is, a set of packets which, however different they may be, are indistinguishable by the forwarding function.

The initial choice of a label need not be based exclusively on the contents of the Layer 3 packet header; for example, forwarding decisions at subsequent hops can also be based on routing policy.

Once a label is assigned, a short label header is added at the front of the Layer 3 packet. This header is carried across the network as part of the packet. At subsequent hops through each MPLS router in the network, labels are swapped and forwarding decisions are made by means of MPLS forwarding table lookup for the label carried in the packet header. Hence, the packet header does not need to be reevaluated during packet transit through the network. Because the label is of fixed length and unstructured, the MPLS forwarding table lookup process is both straightforward and fast.

Distribution of Label Bindings

Each label switching router (LSR) in the network makes an independent, local decision to determine a label value to represent a forwarding equivalence class. This association is known as a label binding. Each LSR informs its neighbors of the label bindings it has made.

When a labeled packet is being sent from LSR A to the neighboring LSR B, the label value carried by the IP packet is the label value that LSR B assigned to represent the forwarding equivalence class of the packet. Thus, the label value changes as the IP packet traverses the network.

The awareness of label bindings by neighbouring routers is facilitated using the following protocols:

- Label Distribution Protocol (LDP) - Enables peer LSRs in an MPLS network to exchange label binding information for supporting hop-by-hop forwarding in an MPLS network.
- Tag Distribution Protocol (TDP) - Supports MPLS forwarding along normally routed paths.
- Resource Reservation Protocol (RSVP) - Supports MPLS traffic engineering.
- Border Gateway Protocol (BGP) - Supports MPLS virtual private networks (VPNs) .

Benefits of MPLS

MPLS provides the following major benefits to service provider networks:

Scalable support for Virtual Private Networks (VPNs)--MPLS enables VPN services to be supported in service provider networks, thereby greatly accelerating Internet growth.

The use of MPLS for VPNs provides an attractive alternative to the building of VPNs by means of either ATM or Frame Relay permanent virtual circuits (PVCs) or various forms of tunneling to interconnect routers at customer sites.

Unlike the PVC VPN model, the MPLS VPN model is highly scalable and can accommodate increasing numbers of sites and customers. The MPLS VPN model also supports "any-to-any" communication among VPN sites without requiring a full mesh of PVCs or the backhauling (suboptimal routing) of traffic across the service provider network. For each MPLS VPN user, the service provider's network appears to function as a private IP backbone over which the user can reach other sites within the VPN organization, but not the sites of any other VPN organization.

From a user perspective, the MPLS VPN model enables network routing to be dramatically simplified. For example, rather than having to manage routing over a topologically complex virtual backbone composed of many PVCs, an MPLS VPN user can generally employ the service provider's backbone as the default route in communicating with all of the other VPN sites.

Explicit routing capabilities (also called constraint-based routing or traffic engineering)--Explicit routing employs "constraint-based routing," in which the path for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow.

In MPLS traffic engineering, factors such as bandwidth requirements, media requirements, and the priority of one traffic flow versus another can be taken into account. These traffic engineering capabilities enable the administrator of a service provider network to

- Control traffic flow in the network
- Reduce congestion in the network
- Make best use of network resources

Thus, the network administrator can specify the amount of traffic expected to flow between various points in the network (thereby establishing a traffic matrix), while relying on the routing system to

- Calculate the best paths for network traffic
- Set up the explicit paths to carry the traffic

Support for IP routing on ATM switches (also called IP and ATM integration)--MPLS enables an ATM switch to perform virtually all of the functions of an IP router. This capability of an ATM switch stems from the fact that the MPLS forwarding paradigm, namely, label swapping, is exactly the same as the forwarding paradigm provided by ATM switch hardware.

The key difference between a conventional ATM switch and an ATM label switch is the control software used by the latter to establish its virtual channel identifier (VCI) table entries. An ATM label switch uses IP routing protocols and the Tag Distribution Protocol (TDP) to establish VCI table entries.

An ATM label switch can function as a conventional ATM switch. In this dual mode, the ATM switch resources (such as VCI space and bandwidth) are partitioned between the MPLS control plane and the ATM control plane. The MPLS control plane provides IP-based services, while the ATM control plane supports ATM-oriented functions, such as circuit emulation or PVC services.

How to Configure MPLS

This section explains how to perform the basic configuration required to prepare a router for MPLS switching and forwarding.

Configuration tasks for other MPLS applications are described in the feature module documentation for the application.

Configuring a Router for MPLS Switching

MPLS switching on Cisco routers requires that Cisco Express Forwarding be enabled.

For more information about Cisco Express Forwarding commands, see the Cisco IOS Switching Command Reference.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Device(config)# ip cef distributed`	Enables Cisco Express Forwarding on the route processor card.

Verifying Configuration of MPLS Switching

To verify that Cisco Express Forwarding has been configured properly, issue the **show ip cef summary** command, which generates output similar to that shown below:

SUMMARY STEPS

1. **show ip cef summary**

DETAILED STEPS

show ip cef summary

Example:

```
Router# show ip cef summary
IP CEF with switching (Table Version 49), flags=0x0
  43 routes, 0 resolve, 0 unresolved (0 old, 0 new)
  43 leaves, 49 nodes, 56756 bytes, 45 inserts, 2 invalidations
  2 load sharing elements, 672 bytes, 2 references
  1 CEF resets, 4 revisions of existing leaves
  4 in-place modifications
  refcounts:  7241 leaf, 7218 node
Adjacency Table has 18 adjacencies
Router#
```

Configuring a Router for MPLS Forwarding

MPLS forwarding on Cisco routers requires that forwarding of IPv4 packets be enabled.

For more information about MPLS forwarding commands, see the *Multiprotocol Label Switching Command Reference*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot /port* [**.** *subinterface*]
4. **mpls ip**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config)# interface gigabitethernet 4/0/0`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for the Gigabit Ethernet interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

Configure either of the following:

- MPLS Label Distribution Protocol (LDP). For information about configuring MPLS LDP, see the *MPLS Label Distribution Protocol Configuration Guid*e.
- Static labels. For information about configuring static labels, see *MPLS Static Labels*.

Verifying Configuration of MPLS Forwarding

To verify that MPLS forwarding has been configured properly, issue the **show mpls interfaces detail** command, which generates output similar to that shown below:

SUMMARY STEPS

1. **show mpls interfaces detail**

DETAILED STEPS

show mpls interfaces detail

Example:

```
Device# show mpls interfaces detail

Interface GigabitEthernet1/0/0:
         IP labeling enabled (ldp)
         LSP Tunnel labeling not enabled
         MPLS operational
         MTU = 1500
Interface POS2/0/0:
         IP labeling enabled (ldp)
         LSP Tunnel labeling not enabled
         MPLS not operational
         MTU = 4470
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

Standard	Title
The supported standards applicable to the MPLS applications appear in the respective feature module for the application.	--

MIBs

MIB	MIBs Link
The supported MIBs applicable to the MPLS applications appear in the respective feature module for the application.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
The supported RFCs applicable to the MPLS applications appear in the respective feature module for the application.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	*Support & Downloads*

Feature Information for MPLS on Cisco Routers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

BGP --Border Gateway Protocol. The predominant interdomain routing protocol used in IP networks.

Border Gateway Protocol --See BGP.

FIB --Forwarding Information Base. A table that contains a copy of the forwarding information in the IP routing table.

Forwarding Information Base --See FIB.

label --A short, fixed-length identifier that tells switching nodes how the data (packets or cells) should be forwarded.

label binding --An association between a label and a set of packets, which can be advertised to neighbors so that a label switched path can be established.

Label Distribution Protocol --See LDP.

Label Forwarding Information Base --See LFIB.

label imposition --The act of putting the first label on a packet.

label switching router --See LSR.

LDP --Label Distribution Protocol. The protocol that supports MPLS hop-by-hop forwarding by distributing bindings between labels and network prefixes.

LFIB --Label Forwarding Information Base. A data structure in which destinations and incoming labels are associated with outgoing interfaces and labels.

LSR --label switching router. A Layer 3 router that forwards a packet based on the value of an identifier encapsulated in the packet.

MPLS --Multiprotocol Label Switching. An industry standard on which label switching is based.

MPLS hop-by-hop forwarding --The forwarding of packets along normally routed paths using MPLS forwarding mechanisms.

Multiprotocol Label Switching --See MPLS.

Resource Reservation Protocol --See RSVP.

RIB --Routing Information Base. A common database containing all the routing protocols running on a router.

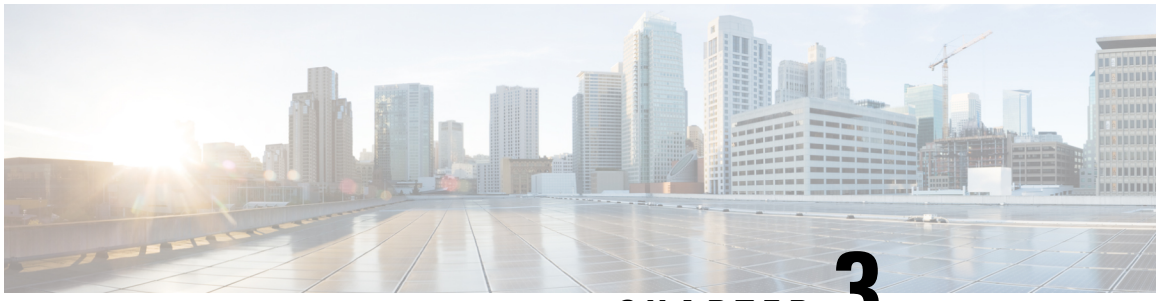
Routing Information Base --See RIB.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

traffic engineering --Techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

Virtual Private Network --See VPN.

VPN --Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.



CHAPTER 3

MPLS Infrastructure Changes Introduction of MFI and Removal of MPLS LSC and LC-ATM Features

This document explains the new MPLS Forwarding Infrastructure (MFI) and removal of support for MPLS label switch controller (LSC) and label-controlled ATM (LC-ATM) features and commands.

Information About MPLS Infrastructure Changes

Introduction of the MPLS Forwarding Infrastructure

The MPLS control plane software is enhanced to make MPLS more scalable and flexible. The MFI, which manages MPLS data structures used for forwarding, replaces the Label Forwarding Information Base (LFIB).

Note The MFI and LFIB do not coexist in the same image. For a list of supported releases, see the "Feature Information for MPLS Forwarding Infrastructure."

Introduction of IP Rewrite Manager

Cisco software introduces a module called the MPLS IP Rewrite Manager (IPRM) that manages the interactions between Cisco Express Forwarding, the IP Label Distribution Modules (LDMs), and the MFI. MPLS IPRM is enabled by default. You need not configure or customize the IPRM. These commands are related to IPRM:

- **clear mpls ip iprm counters**
- **debug mpls ip iprm**
- **debug mpls ip iprm cef**
- **debug mpls ip iprm events**
- **debug mpls ip iprm ldm**

- **debug mpls ip iprm mfi**
- **show mpls ip iprm counters**
- **show mpls ip iprm ldm**

For information about these commands, see the *Cisco IOS Debug Command Reference* and the *Cisco IOS MPLS Command Reference*.

MPLS LSC and LC-ATM Configurations

Before upgrading to Cisco IOS Release 12.4(20)T, remove all the MPLS LSC and LC-ATM configurations from the routers in your network. If your core network has ATM links, you can use packet-based MPLS. See the MPLS Label Distribution Protocol Overview for more information. If you provide ATM access to customers, you can use the Any Transport over MPLS: ATM over MPLS feature. See Any Transport over MPLS for more information.

If you have MPLS LSC or LC-ATM features configured and you upgrade to Cisco IOS Release 12.4(20)T, the configuration is not accepted. The system displays “unrecognized command” errors for any commands that are no longer supported.

Additional References

Related Documents

Related Topic	Document Title
MPLS commands	*Cisco IOS MPLS Command Reference*
MPLS Label Distribution Protocol	MPLS Label Distribution Protocol Overview
Layer 2 VPN features over MPLS	Any Transport over MPLS

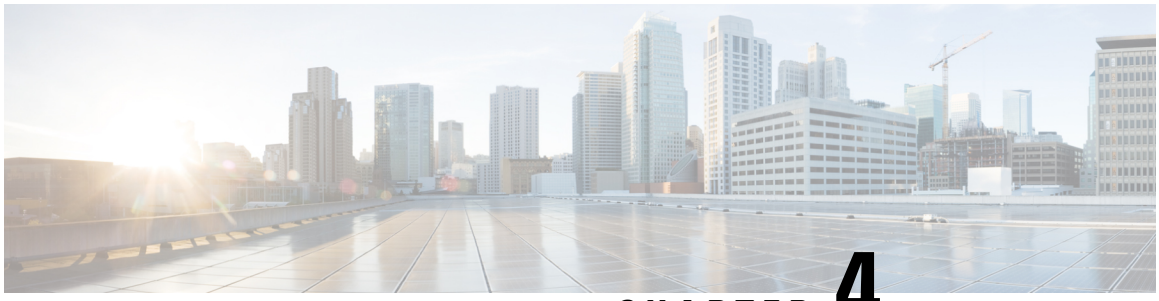
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Infrastructure Changes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 4

MPLS Static Labels

This document describes the Cisco MPLS Static Labels feature. The MPLS Static Labels feature provides the means to configure statically:

- The binding between a label and an IPv4 prefix
- The contents of an LFIB crossconnect entry

Restrictions for MPLS Static Labels

- The trouble shooting process for MPLS static labels is complex.
- On a provider edge (PE) router for MPLS VPNs, there is no mechanism for statically binding a label to a customer network prefix (VPN IPv4 prefix).
- MPLS static crossconnect labels remain in the LFIB even if the router to which the entry points goes down.
- MPLS static crossconnect mappings remain in effect even with topology changes.
- MPLS static labels are not supported for label-controlled Asynchronous Transfer Mode (lc-atm).
- MPLS static bindings are not supported for local prefixes.

Prerequisites for MPLS Static Labels

The network must support the following Cisco IOS features before you enable MPLS static labels:

- Multiprotocol Label Switching (MPLS)

- Cisco Express Forwarding

Information About MPLS Static Labels

MPLS Static Labels Overview

Generally, label switching routers (LSRs) dynamically learn the labels they should use to label-switch packets by means of label distribution protocols that include:

- Label Distribution Protocol (LDP), the Internet Engineering Task Force (IETF) standard, used to bind labels to network addresses
- Resource Reservation Protocol (RSVP) used to distribute labels for traffic engineering (TE)
- Border Gateway Protocol (BGP) used to distribute labels for Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs)

To use a learned label to label-switch packets, an LSR installs the label into its Label Forwarding Information Base (LFIB).

The MPLS Static Labels feature provides the means to configure statically:

- The binding between a label and an IPv4 prefix
- The contents of an LFIB crossconnect entry

Benefits of MPLS Static Labels

Static Bindings Between Labels and IPv4 Prefixes

Static bindings between labels and IPv4 prefixes can be configured to support MPLS hop-by-hop forwarding through neighbor routers that do not implement LDP label distribution.

Static Crossconnects

Static crossconnects can be configured to support MPLS Label Switched Path (LSP) midpoints when neighbor routers do not implement either the LDP or RSVP label distribution, but do implement an MPLS forwarding path.

How to Configure MPLS Static Labels

Configuring MPLS Static Prefix Label Bindings

To configure MPLS static prefix/label bindings, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*]
4. **mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] label

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*] **Example:** `Router(config)# mpls label range 200 100000 static 16 199`	Specifies a range of labels for use with MPLS Static Labels feature. (Default is no labels reserved for static assignment.)
Step 4	**mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] label **Example:** `Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55`	Specifies static binding of labels to IPv4 prefixes. Bindings specified are installed automatically in the MPLS forwarding table as routing demands.

Verifying MPLS Static Prefix Label Bindings

To verify the configuration for MPLS static prefix/label bindings, use this procedure:

SUMMARY STEPS

1. Enter **show mpls label range** command. The output shows that the new label ranges do not take effect until a reload occurs:
2. Enter the **show mpls static binding ipv4** command to show the configured static prefix/label bindings:
3. Use the **show mpls forwarding-table** command to determine which static prefix/label bindings are currently in use for MPLS forwarding.

DETAILED STEPS

Step 1 Enter **show mpls label range** command. The output shows that the new label ranges do not take effect until a reload occurs:

Example:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 16/100000
   [Configured range for next reload: Min/Max label: 200/100000]
Range for static labels: Min/Max/Number: 16/199
```

The following output from the **show mpls label range** command, executed after a reload, indicates that the new label ranges are in effect:

Example:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 200/100000
Range for static labels: Min/Max/Number: 16/199
```

Step 2 Enter the **show mpls static binding ipv4** command to show the configured static prefix/label bindings:

Example:

```
Router# show mpls static binding ipv4
10.17.17.17/32: Incoming label: 251 (in LIB)
  Outgoing labels:
     10.0.0.1                  18
10.18.18.18/32: Incoming label: 201 (in LIB)
  Outgoing labels:
10.0.0.1 implicit-null
```

Step 3 Use the **show mpls forwarding-table** command to determine which static prefix/label bindings are currently in use for MPLS forwarding.

Example:

```
Router# show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
201    Pop tag     10.18.18.18/32   0          PO1/1/0    point2point
       2/35        10.18.18.18/32   0          AT4/1/0.1  point2point
251    18          10.17.17.17/32   0          PO1/1/0    point2point
```

Configuring MPLS Static Crossconnects

To configure MPLS static crossconnects, use the following command beginning in global configuration mode:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*]
4. **mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] *label*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*] **Example:** `Router(config)# mpls label range 200 100000 static 16 199`	Specifies a range of labels for use with MPLS Static Labels feature. (Default is no labels reserved for static assignment.)
Step 4	**mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] *label* **Example:** `Router(config)# Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55`	Specifies static binding of labels to IPv4 prefixes. Bindings specified are installed automatically in the MPLS forwarding table as routing demands.

Verifying MPLS Static Crossconnect Configuration

To verify the configuration for MPLS static crossconnects, use this procedure:

SUMMARY STEPS

1. Use the **show mpls static crossconnect** command to display information about crossconnects that have been configured:

DETAILED STEPS

Use the **show mpls static crossconnect** command to display information about crossconnects that have been configured:

Example:

```
Router# show mpls static crossconnect
Local  Outgoing    Outgoing   Next Hop
```

```
label  label       interface
34     22          pos3/0/0   point2point  (in LFIB)
```

Monitoring and Maintaining MPLS Static Labels

To monitor and maintain MPLS static labels, use one or more of the following commands:

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table**
3. **show mpls label range**
4. **show mpls static binding ipv4**
5. **show mpls static crossconnect**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** **Example:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS LFIB.
Step 3	**show mpls label range** **Example:** `Router# show mpls label range`	Displays information about the static label range.
Step 4	**show mpls static binding ipv4** **Example:** `Router# show mpls static binding ipv4`	Displays information about the configured static prefix/label bindings.
Step 5	**show mpls static crossconnect** **Example:** `Router# show mpls static crossconnect`	Displays information about the configured crossconnects.

Configuration Examples for MPLS Static Labels

Example Configuring MPLS Static Prefixes Labels

In the following output, the **mpls label range** command reconfigures the range used for dynamically assigned labels from 16 to 100000 to 200 to 100000 and configures a static label range of 16 to 199.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls label range 200 100000 static 16 199
% Label range changes take effect at the next reload.
Router(config)# end
```

In the following output, the **show mpls label range** command indicates that the new label ranges do not take effect until a reload occurs:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 16/100000
   [Configured range for next reload: Min/Max label: 200/100000]
Range for static labels: Min/Max/Number: 16/199
```

In the following output, the **show mpls label range** command, executed after a reload, indicates that the new label ranges are in effect:

```
Router# show mpls label range

Downstream label pool: Min/Max label: 200/100000
Range for static labels: Min/Max/Number: 16/199
```

In the following output, the **mpls static binding ipv4** commands configure static prefix/label bindings. They also configure input (local) and output (remote) labels for various prefixes:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55
Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 output 10.0.0.66 2607
Router(config)# mpls static binding ipv4 10.6.0.0 255.255.0.0 input 17
Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 output 10.13.0.8 explicit-null
Router(config)# end
```

In the following output, the **show mpls static binding ipv4** command displays the configured static prefix/label bindings:

```
Router# show mpls static binding ipv4

10.0.0.0/8: Incoming label: none;
  Outgoing labels:
10.13.0.8           explicit-null
10.0.0.0/8: Incoming label: 55 (in LIB)
  Outgoing labels:
     10.0.0.66              2607
10.66.0.0/16: Incoming label: 17 (in LIB)
  Outgoing labels:  None
```

Example Configuring MPLS Static Crossconnects

In the following output, the **mpls static crossconnect** command configures a crossconnect from incoming label 34 to outgoing label 22 out interface pos3/0/0:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls static crossconnect 34 pos3/0/0 22
Router(config)# end
```

In the following output, the **show mpls static crossconnect** command displays the configured crossconnect:

```
Router# show mpls static crossconnect
Local  Outgoing    Outgoing   Next Hop
label  label       interface
34     22          pos3/0/0   point2point  (in LFIB)
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	*Multiprotocol Label Switching Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Static Labels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for MPLS Static Labels

Feature Name	Releases	Feature Information
MPLS Static Labels	Cisco IOS XE Amsterdam 17.3.2	The MPLS Static Labels feature provides the means to configure the following items statically: • The binding between a label and an IPv4 prefix • The contents of an LFIB crossconnect entry

Glossary

BGP --Border Gateway Protocol. The predominant interdomain routing protocol used in IP networks.

Border Gateway Protocol --See BGP.

FIB --Forwarding Information Base. A table that contains a copy of the forwarding information in the IP routing table.

Forwarding Information Base --See FIB.

label --A short, fixed-length identifier that tells switching nodes how the data (packets or cells) should be forwarded.

label binding --An association between a label and a set of packets, which can be advertised to neighbors so that a label switched path can be established.

Label Distribution Protocol --See LDP.

Label Forwarding Information Base --See LFIB.

label imposition --The act of putting the first label on a packet.

label switching router --See LSR.

LDP --Label Distribution Protocol. The protocol that supports MPLS hop-by-hop forwarding by distributing bindings between labels and network prefixes.

LFIB --Label Forwarding Information Base. A data structure in which destinations and incoming labels are associated with outgoing interfaces and labels.

LSR --label switching router. A Layer 3 router that forwards a packet based on the value of an identifier encapsulated in the packet.

MPLS --Multiprotocol Label Switching. An industry standard on which label switching is based.

MPLS hop-by-hop forwarding --The forwarding of packets along normally routed paths using MPLS forwarding mechanisms.

Multiprotocol Label Switching --See MPLS.

Resource Reservation Protocol --See RSVP.

RIB --Routing Information Base. A common database containing all the routing protocols running on a router.

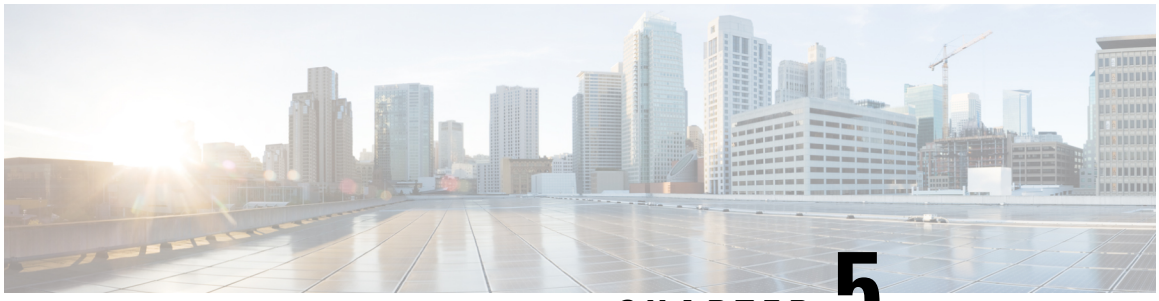
Routing Information Base --See RIB.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

traffic engineering --Techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

Virtual Private Network --See VPN.

VPN --Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.



CHAPTER 5

MPLS Multilink PPP Support

The MPLS Multilink PPP Support feature ensures that MPLS Layer 3 Virtual Private Networks (VPNs) with quality of service (QoS) can be enabled for bundled links. This feature supports Multiprotocol Label Switching (MPLS) over Multilink PPP (MLP) links in the edge (provider edge [PE]-to-customer edge [CE]) or in the MPLS core (PE-to-PE and PE-to-provider [P] device).

Service providers that use relatively low-speed links can use MLP to spread traffic across them in their MPLS networks. Link fragmentation and interleaving (LFI) should be deployed in the CE-to-PE link for efficiency, where traffic uses a lower link bandwidth (less than 768 kbps). The MPLS Multilink PPP Support feature can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate load sharing of traffic.

Prerequisites for MPLS Multilink PPP Support

- Cisco Express Forwarding must be enabled.
- Multiprotocol Label Switching (MPLS) must be enabled on provider edge (PE) and provider (P) devices.
- Cisco Express Forwarding switching must be enabled on the interface by using the **ip route-cache cef** command.

Information About MPLS Multilink PPP Support

MPLS Layer 3 Virtual Private Network Features Supported for Multilink PPP

The table below lists Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) features supported for Multilink PPP (MLP) and indicates if the feature is supported on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Table 2: MPLS Layer 3 VPN Features Supported for MLP

MPLS L3 VPN Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Static routes	Supported	Not supported	Not supported
External Border Gateway Protocol (eBGP)	Supported	Not applicable to this configuration	Supported
Intermediate System-to-Intermediate System (IS-IS)	Not supported	Supported	Not supported
Open Shortest Path First (OSPF)	Supported	Supported	Not supported
Enhanced Interior Gateway Routing Protocol (EIGRP)	Supported	Supported	Not supported
Interprovider interautonomous (Inter-AS) VPNs (with Label Distribution Protocol [LDP])	Not applicable to this configuration	Supported (MLP between Autonomous System Boundary Routers [ASBRs])	Not applicable to this configuration
Inter-AS VPNs with IPv4 Label Distribution	Not applicable to this configuration	Supported (MLP between ASBRs)	Not applicable to this configuration
CSC VPNs (with LDP)	Not supported	Not applicable to this configuration	Supported
CSC VPNs with IPv4 label distribution	Supported	Not applicable to this configuration	Supported
External and internal BGP (eiBGP) Multipath	Not supported	Not supported	Not applicable to this configuration
Internal BGP (iBGP) Multipath	Not applicable to this configuration	Not supported	Not applicable to this configuration
eBGP Multipath	Not supported	Not supported	Not supported

MPLS Quality of Service Features Supported for Multilink PPP

The table below lists the Multiprotocol Label Switching (MPLS) quality of service (QoS) features supported for Multilink PPP (MLP) and indicates if the feature is supported on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Table 3: MPLS QoS Features Supported for MLP

MPLS QoS Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Default copy of IP Precedence to EXP bits and the reverse	Supported	Not supported	Not supported

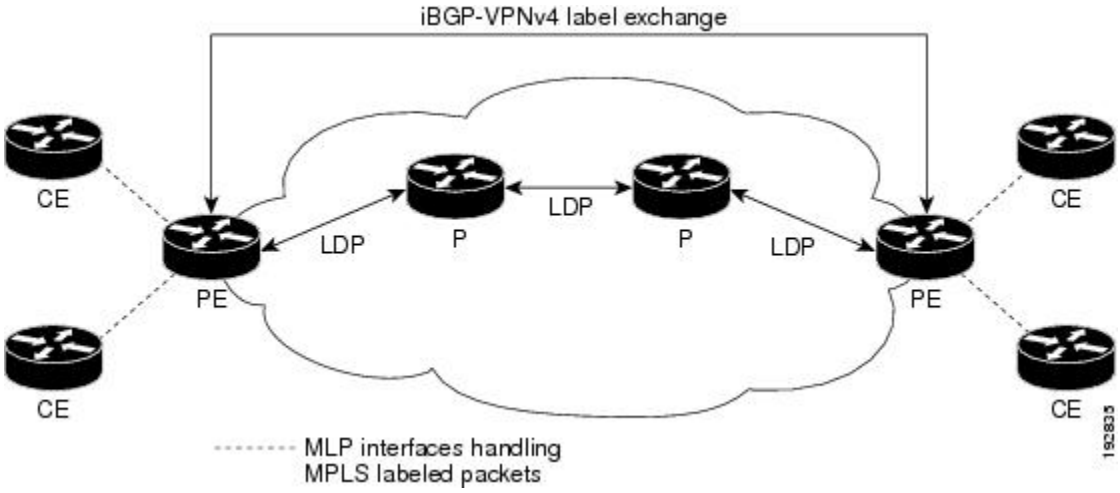
MPLS QoS Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
Set MPLS EXP bits using the modular QoS Command-Line Interface (MQC)	Supported	Supported	Supported
Matching on MPLS EXP using MQC	Supported	Supported	Supported
Low Latency Queueing (LLQ)/Class-Based Weighted Fair Queueing (CBWFQ) support	Supported	Supported	Supported
Weighted Random Early Detection (WRED) based on EXP bits using MQC	Supported	Supported	Supported
Policer with EXP bit-marking using MQC-3 action	Supported	Supported	Supported
Support for EXP bits in MPLS accounting	Supported	Supported	Supported

MPLS Multilink PPP Support and PE-to-CE Links

The figure below shows a typical Multiprotocol Label Switching (MPLS) network in which the provider edge (PE) device is responsible for label imposition (at ingress) and disposition (at egress) of the MPLS traffic.

In this topology, Multilink PPP (MLP) is deployed on the PE-to-customer edge (CE) links. The Virtual Private Network (VPN) routing and forwarding instance (VRF) interface is in a multilink bundle. There is no MPLS interaction with MLP; all packets coming into the MLP bundle are IP packets.

Figure 1: MLP and Traditional PE-to-CE Links



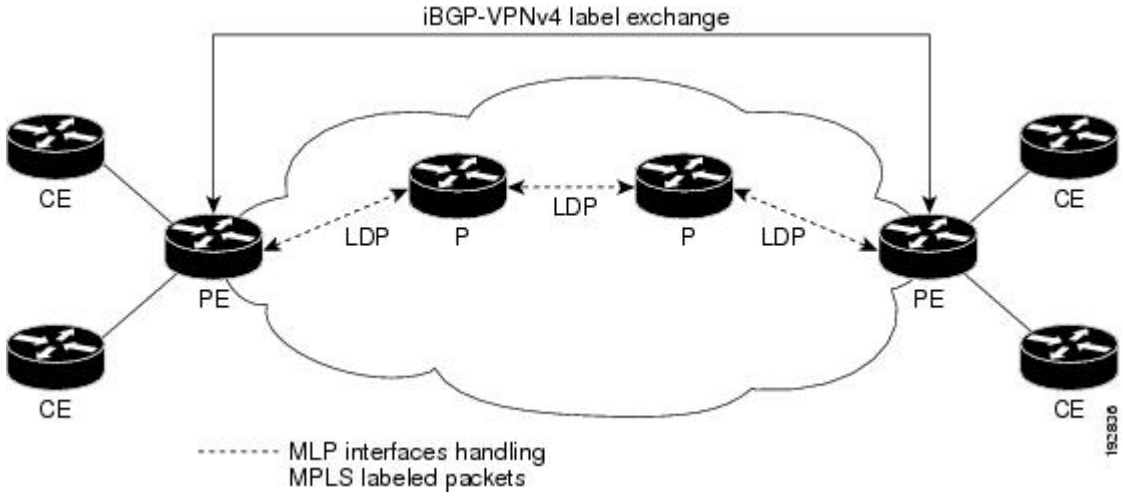
The PE-to-CE routing protocols that are supported for the MPLS Multilink PPP Support feature are external BGP (eBGP), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP). Static routes are also supported between the CE and PE device.

Quality of service (QoS) features that are supported for the MPLS Multilink PPP Support feature on CE-to-PE links are link fragmentation and interleaving (LFI), header compression, policing, marking, and classification.

MPLS Multilink PPP Support and Core Links

The figure below shows a sample topology in which Multiprotocol Label Switching (MPLS) is deployed over Multilink PPP (MLP) on provider edge-to-provider (PE-to-P) and P-to-P links. Enabling MPLS on MLP for PE-to-P links is similar to enabling MPLS on MLP for P-to-P links.

Figure 2: MLP on PE-to-P and P-to-P Links



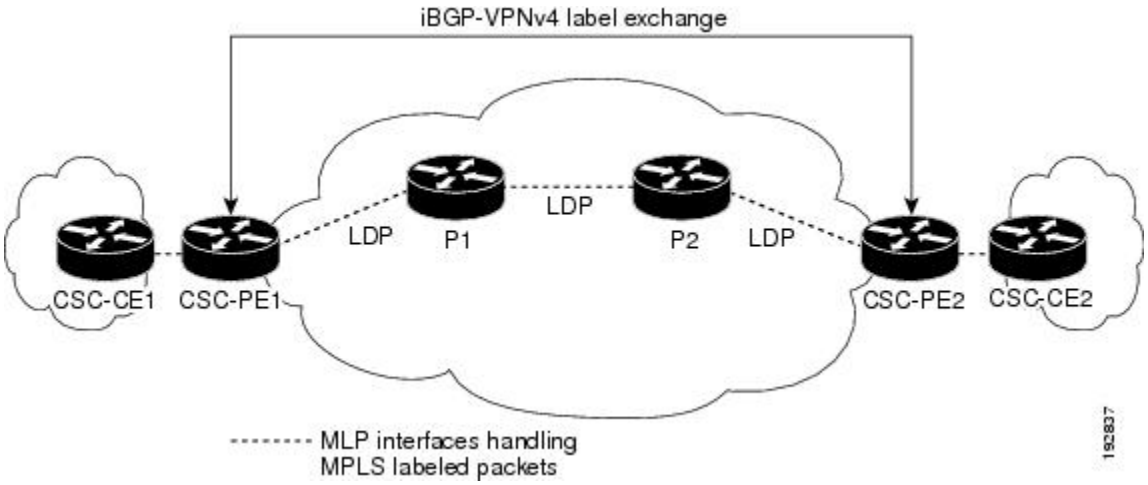
You employ MLP in the PE-to-P or P-to-P links primarily so that you can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate the load sharing of traffic.

In addition to requiring MLP on the PE-to-P links, the MPLS Multilink PPP Support feature requires the configuration of an IGP routing protocol and the Label Distribution Protocol (LDP).

MPLS Multilink PPP Support in a CSC Network

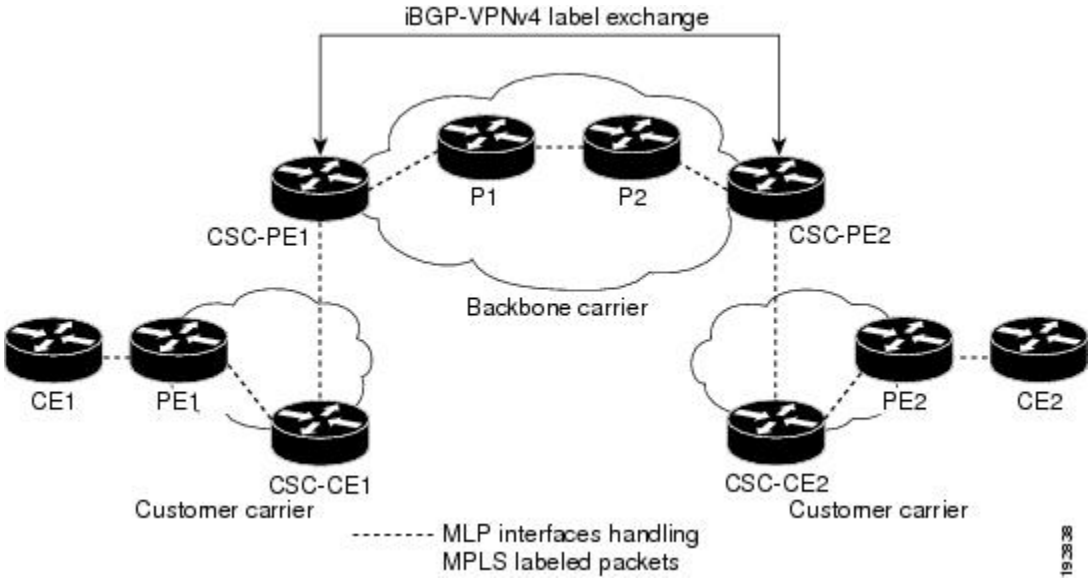
The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) network where Multilink PPP (MLP) is configured on the CSC customer edge (CE)-to-provider edge (PE) links.

Figure 3: MLP on CSC CE-to-PE Links with MPLS VPN Carrier Supporting Carrier



The MPLS Multilink PPP Support feature supports MLP between CSC-CE and CSC-PE links with the Label Distribution Protocol (LDP) or with external Border Gateway Protocol (eBGP) IPv4 label distribution. This feature also supports link fragmentation and interleaving (LFI) for an MPLS VPN CSC configuration. The figure below shows all MLP links that this feature supports for CSC configurations.

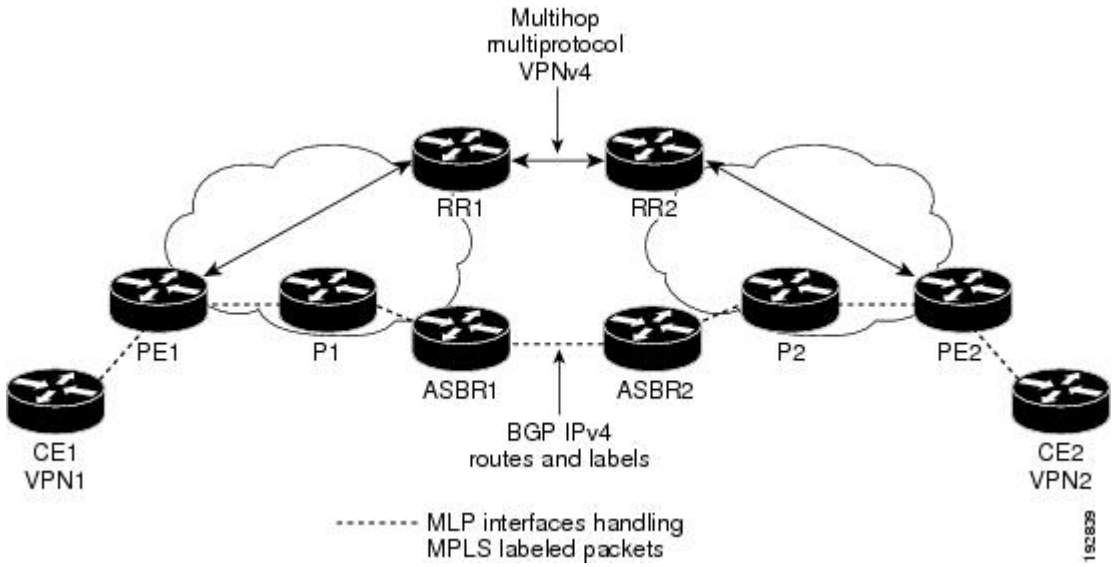
Figure 4: MLP Supported Links with MPLS VPN Carrier Supporting Carrier



MPLS Multilink PPP Support in an Interautonomous System

The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) interautonomous system (Inter-AS) network where Multilink PPP (MLP) is configured on the provider edge-to-customer edge (PE-to-CE) links.

Figure 5: MLP on ASBR-to-PE Links in an MPLS VPN Inter-AS Network



The MPLS Multilink PPP Support feature supports MLP between Autonomous System Boundary Router (ASBR) links for Inter-AS VPNs with Label Distribution Protocol (LDP) and with external Border Gateway Protocol (eBGP) IPv4 label distribution.

How to Configure MPLS Multilink PPP Support

The tasks in this section can be performed on customer edge-to-provider edge (CE-to-PE) links, PE-to-provider (P) links, P-to-P links, and Carrier Supporting Carrier (CSC) CE-to-PE links.

Enabling Cisco Express Forwarding

Perform the following task to enable Cisco Express Forwarding. Cisco Express Forwarding is required for the forwarding of MLP traffic.

Before you begin

Multilink PPP requires the configuration of Cisco Express Forwarding. To find out if Cisco Express Forwarding is enabled on your device, enter the**show ip cef** command. If Cisco Express Forwarding is enabled, you receive output that looks like the following:

```
Device# show ip cef
Prefix              Next Hop            Interface
10.2.61.8/24        192.168.100.1       FastEthernet1/0/0
                    192.168.101.1       FastEthernet6/1/0
```

If Cisco Express Forwarding is not enabled on your platform, the output for the **show ip cef** command looks like the following:

```
Device# show ip cef
%CEF not running
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Device(config)# ip cef`	Enables Cisco Express Forwarding.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Creating a Multilink Bundle

Perform this task to create a multilink bundle for the MPLS Multilink PPP Support feature. This multilink bundle can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate load sharing of traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ip address** *address mask* [**secondary**]
5. **encapsulation** *encapsulation-type*
6. **ppp multilink**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 1`	Creates a multilink bundle and enters multilink interface configuration mode. • The *group-number* argument is the number of the multilink bundle (a nonzero number).
Step 4	**ip address** *address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.0 255.255.0.0`	Sets a primary or secondary IP address for an interface. • The *address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address. This command is used to assign an IP address to the multilink interface.
Step 5	**encapsulation** *encapsulation-type* **Example:** `Device(config-if)# encapsulation ppp`	Sets the encapsulation method as PPP to be used by the interface. • The *encapsulation-type* argument specifies the encapsulation type.
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables MLP on an interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Assigning an Interface to a Multilink Bundle

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller** {**t1** | **e1**} *slot/port*
4. **channel-group** *channel-number* **timeslots** *range*
5. **exit**
6. **interface serial** *slot/subslot/port*[**.***subinterface*]
7. **ip route-cache** [**cef**]
8. **no ip address**
9. **keepalive** [*period* [*retries*]]
10. **encapsulation** *encapsulation-type*
11. **ppp multilink group** *group-number*
12. **ppp multilink**
13. **ppp authentication chap**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**controller** {**t1** | **e1**} *slot/port* **Example:** `Device# controller t1 1/3`	Configures a T1 or E1 controller and enters controller configuration mode. • The **t1** keyword indicates a T1 line card. • The **e1** keyword indicates an E1 line card. • The *slot/port* arguments are the backplane slot number and port number on the interface. Refer to your hardware installation manual for the specific slot numbers and port numbers.
Step 4	**channel-group** *channel-number* **timeslots** *range* **Example:** `Device(config-controller)# channel-group 1 timeslots 1`	Defines the time slots that belong to each T1 or E1 circuit. • The *channel-number* argument is the channel-group number. When a T1 data line is configured, channel-group numbers can be values from 0 to 23.

	Command or Action	Purpose
		When an E1 data line is configured, channel-group numbers can be values from 0 to 30. • The **timeslots** *range* keyword and argument specifies one or more time slots or ranges of time slots belonging to the channel group. The first time slot is numbered 1. For a T1 controller, the time slot range is from 1 to 24. For an E1 controller, the time slot range is from 1 to 31. You can specify a time slot range (for example, 1-29), individual time slots separated by commas (for example 1, 3, 5), or a combination of the two (for example 1-14, 15, 17-31).
Step 5	**exit** **Example:** `Device(config-controller)# exit`	Returns to global configuration mode.
Step 6	**interface serial** *slot/subslot/port*[**.***subinterface*] **Example:** `Device(config)# interface serial 1/0/0:1`	Configures a serial interface and enters interface configuration mode.
Step 7	**ip route-cache** [**cef**] **Example:** `Device(config-if)# ip route-cache cef`	Controls the use of switching methods for forwarding IP packets. • The **cef** keyword enables Cisco Express Forwarding operation on an interface after Cisco Express Forwarding operation was disabled.
Step 8	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes any specified IP address.
Step 9	**keepalive** [*period* [*retries*]] **Example:** `Device(config-if)# keepalive`	Enables keepalive packets and specifies the number of times that the Cisco software tries to send keepalive packets without a response before bringing down the interface or before bringing the tunnel protocol down for a specific interface. • The *period* argument is an integer value, in seconds, greater than 0. The default is 10. • The *retries* argument specifies the number of times that the device continues to send keepalive packets without a response before bringing the interface down. Enter an integer value greater than 1 and less than 255. If you do not enter a value, the value that was

	Command or Action	Purpose
		previously set is used; if no value was specified previously, the default of 5 is used. If you are using this command with a tunnel interface, the command specifies the number of times that the device continues to send keepalive packets without a response before bringing the tunnel interface protocol down.
Step 10	**encapsulation** *encapsulation-type* **Example:** `Device(config-if)# encapsulation ppp`	Sets the encapsulation method used by the interface. • The *encapsulation-type* argument specifies the encapsulation type. The example specifies PPP encapsulation.
Step 11	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 1`	Restricts a physical link to join only one designated multilink group interface. • The *group-number* argument is the number of the multilink bundle (a nonzero number).
Step 12	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables MLP on the interface.
Step 13	**ppp authentication chap** **Example:** `Device(config-if)# ppp authentication chap`	(Optional) Enables Challenge Handshake Authentication Protocol (CHAP) authentication on the serial interface.
Step 14	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Disabling PPP Multilink Fragmentation

Perform this task to disable PPP multilink fragmentation. PPP multilink fragmentation is enabled by default.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation might produce better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation can be outweighed by the added load on the CPU.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **ppp multilink fragmentation disable**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial 1/0/0`	Configures an interface type and enters interface configuration mode. • The *type* argument indicates the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when the interface is added to a system, and they can be displayed with the **show interfaces** command.
Step 4	**ppp multilink fragmentation disable** **Example:** `Device(config-if)# ppp multilink fragmentation disable`	Disables packet fragmentation.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the Multilink PPP Configuration

SUMMARY STEPS

1. **enable**
2. **show ip interface brief**
3. **show ppp multilink**
4. **show ppp multilink interface** *interface-bundle*

5. **show interface** *type number*
6. **show mpls forwarding-table**
7. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show ip interface brief**

Verifies logical and physical Multilink PPP (MLP) interfaces.

Example:

Step 3 **show ppp multilink**

Verifies that you have created a multilink bundle.

Example:

Step 4 **show ppp multilink interface** *interface-bundle*

Displays information about a specific MLP interface.

Example:

Step 5 **show interface** *type number*

Displays information about serial interfaces in your configuration.

Example:

```
Device#


  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open, crc 16, Data non-inverted
  Last input 00:00:01, output 00:00:01, output hang never
  Last clearing of "show interface" counters 00:47:13
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     722 packets input, 54323 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     697 packets output, 51888 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions no alarm present
  Timeslot(s) Used:1, subrate: 64Kb/s, transmit delay is 0 flags
```

```
  Transmit queue length 25

Device#


  Hardware is Multichannel T1
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open, crc 16, Data non-inverted
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters 00:47:16
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     725 packets input, 54618 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     693 packets output, 53180 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions no alarm present
  Timeslot(s) Used:2, subrate: 64Kb/s, transmit delay is 0 flags
  Transmit queue length 26
```

You can also use the **show interface** command to display information about the multilink interface:

Example:

```
Device# show interface multilink6

Multilink6 is up, line protocol is up
  Hardware is multilink group interface
  Internet address is 10.30.0.2/8
  MTU 1500 bytes, BW 128 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open, multilink Open
  Open: CDPCP, IPCP, TAGCP, loopback not set
  DTR is pulsed for 2 seconds on reset
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters 00:48:43
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     1340 packets input, 102245 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1283 packets output, 101350 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Step 6 **show mpls forwarding-table**

Displays contents of the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB). Look for information on multilink interfaces associated with a point2point next hop.

Example:

```
Device# show mpls forwarding-table
```

```
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
16     Untagged    10.30.0.1/32      0          Mu6        point2point
17     Pop tag     10.0.0.3/32       0          Mu6        point2point
18     Untagged    10.0.0.9/32[V]    0          Mu10       point2point
19     Untagged    10.0.0.11/32[V]   6890       Mu10       point2point
20     Untagged    10.32.0.0/8[V]    530        Mu10       point2point
21     Aggregate   10.34.0.0/8[V]    0
22     Untagged    10.34.0.1/32[V]   0          Mu10       point2point
```

Use the **show ip bgp vpnv4** command to display VPN address information from the Border Gateway Protocol (BGP) table.

Example:

```
Device# show ip bgp vpnv4 all summary

BGP router identifier 10.0.0.1, local AS number 100
BGP table version is 21, main routing table version 21
10 network entries using 1210 bytes of memory
10 path entries using 640 bytes of memory
2 BGP path attribute entries using 120 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1994 total bytes of memory
BGP activity 10/0 prefixes, 10/0 paths, scan interval 5 secs
10.0.0.3  4 100 MsgRc52 MsgSe52  TblV21  0  0 00:46:35 State/P5xRcd
```

Step 7 **exit**

Returns to user EXEC mode.

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS Multilink PPP Support

Example: Configuring Multilink PPP on an MPLS CSC PE Device

The following example shows how to configure for Multiprotocol Label Switching (MPLS) Carrier Supporting Carrier (CSC) provider edge (PE) device.

```
!
mpls label protocol ldp
ip cef
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
!
```

```
!

 no ip address
 encapsulation ppp

 ppp multilink
 ppp multilink group 1

interface Multilink1
 ip vrf forwarding vpn2
 ip address 10.35.0.2 255.0.0.0
 no peer neighbor-route
 load-interval 30
 ppp multilink
 ppp multilink interleave
 ppp multilink group 1

!
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Multilink1
 network 10.0.0.7 0.0.0.0 area 200
 network 10.31.0.0 0.255.255.255 area 200
!
!
 router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.0.0.11 remote-as 200
 neighbor 10.0.0.11 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.0.0.11 activate
 neighbor 10.0.0.11 send-community extended
 bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 redistribute connected
 neighbor 10.35.0.1 remote-as 300
 neighbor 10.35.0.1 activate
 neighbor 10.35.0.1 as-override
 neighbor 10.35.0.1 advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

Example: Enabling Cisco Express Forwarding

The following example shows how to enable Cisco Express Forwarding for Multilink PPP (MLP) configurations:

```
Device> enable
Device# configure terminal
Device(config)# ip cef
```

Example: Creating a Multilink Bundle

The following example shows how to create a multilink bundle for the MPLS Multilink PPP Support feature:

```
Device(config)# interface multilink 1
Device(config-if)# ip address 10.0.0.0 10.255.255.255
Device(config-if)# encapsulation ppp
Device(config-if)# ppp chap hostname group 1
Device(config-if)# ppp multilink
Device(config-if)# ppp multilink group 1
```

Example: Assigning an Interface to a Multilink Bundle

The following example shows how to create four multilink interfaces with Cisco Express Forwarding switching and Multilink PPP (MLP) enabled. Each of the newly created interfaces is added to a multilink bundle.

```
interface multilink1
 ip address 10.0.0.0 10.255.255.255
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1



 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
```

Additional References for MPLS Multilink PPP Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Basic MPLS VPNs	"MPLS Virtual Private Networks" chapter in the *MPLS Layer 3 VPNs Configuration Guide*

RFCs

RFCs	Title
RFC 1990	*The PPP Multilink Protocol (MP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Multilink PPP Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for MPLS Multilink PPP Support

Feature Name	Releases	Feature Information
MPLS Multilink PPP Support		The MPLS Multilink PPP Support feature ensures that MPLS Layer 3 Virtual Private Networks (VPNs) with quality of service (QoS) can be enabled for bundled links. This feature supports Multiprotocol Label Switching (MPLS) over Multilink PPP (MLP) links in the edge (provider edge [PE]-to-customer edge [CE]) or in the MPLS core (PE-to-PE and PE-to-provider [P]device).

Glossary

bundle—A group of interfaces connected by parallel links between two systems that have agreed to use Multilink PPP (MLP) over those links.

CBWFQ—class-based weighted fair queueing. A queueing option that extends the standard Weighted Fair Queueing (WFQ) functionality to provide support for user-defined traffic classes.

Cisco Express Forwarding—A proprietary form of switching that optimizes network performance and scalability for networks with large and dynamic traffic patterns, such as the Internet, and for networks characterized by intensive web-based applications or interactive sessions. Although you can use Cisco Express Forwarding in any part of a network, it is designed for high-performance, highly resilient Layer 3 IP backbone switching.

EIGRP—Enhanced Interior Gateway Routing Protocol. An advanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. It provides superior convergence properties and operating efficiency, and combines the advantages of link-state protocols with those of distance vector protocols.

IGP—Interior Gateway Protocol. An Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

IGRP—Interior Gateway Routing Protocol. An Interior Gateway Protocol (IGP) developed by Cisco to address the issues associated with routing in large, heterogeneous networks. Compare with Enhanced Interior Gateway Routing Protocol (EIGRP).

IS-IS—Intermediate System-to-Intermediate System. An Open Systems Interconnection (OSI) link-state hierarchical routing protocol, based on DECnet Phase V routing, in which IS-IS devices exchange routing information based on a single metric to determine network topology.

LCP—Link Control Protocol. A protocol that establishes, configures, and tests data link connections for use by PPP.

LFI—ink fragmentation and interleaving. The LFI feature reduces delay on slower-speed links by breaking up large datagrams and interleaving low-delay traffic packets with the smaller packets resulting from the fragmented datagram. LFI allows reserve queues to be set up so that Real-Time Protocol (RTP) streams can be mapped into a higher priority queue in the configured weighted fair queue set.

link—One of the interfaces in a bundle.

LLQ—low latency queueing. A quality of service QoS queueing feature that provides a strict priority queue (PQ) for voice traffic and weighted fair queues for other classes of traffic. It is also called priority queueing/class-based weighted fair queueing (PQ/CBWFQ).

MLP—Multilink PPP. A method of splitting, recombining, and sequencing datagrams across multiple logical links. The use of MLP increases throughput between two sites by grouping interfaces and then load balancing packets over the grouped interfaces (called a bundle). Splitting packets at one end, sending them over the bundled interfaces, and recombining them at the other end achieves load balancing.

MQC—Modular QoS CLI. MQC is a CLI structure that allows users to create traffic polices and attach these polices to interfaces. MQC allows users to specify a traffic class independently of QoS policies.

NCP—Network Control Protocol. A series of protocols for establishing and configuring different network layer protocols (such as for AppleTalk) over PPP.

OSPF—Open Shortest Path First. A link-state, hierarchical Interior Gateway Protocol (IGP) routing algorithm proposed as a successor to Routing Information Protocol (RIP) in the Internet community. OSPF features include least-cost routing, multipath routing, and load balancing. OSPF was derived from an early version of the IS-IS protocol.

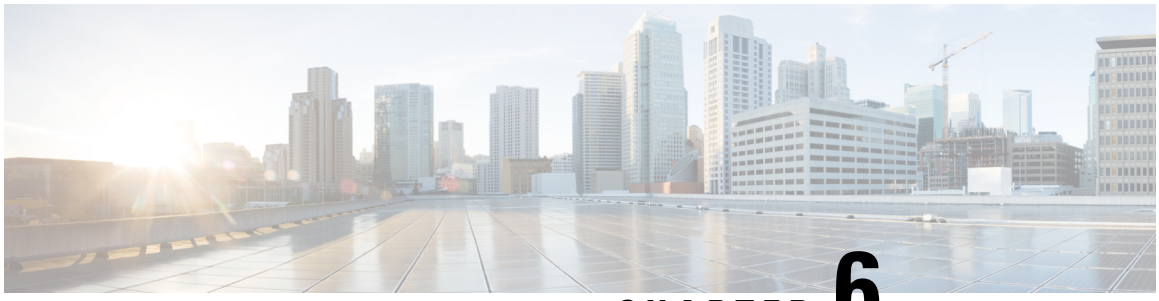
PPP—Point-to-Point Protocol. A successor to the Serial Line Interface Protocol (SLIP) that provides device-to-device and host-to-network connections over synchronous and asynchronous circuits. PPP works with several network layer protocols (such as IP, Internetwork Packet Exchange [IPX], and AppleTalk Remote Access [ARA]). PPP also has built-in security mechanisms (such as Challenge Handshake Authentication Protocol [CHAP] and Password Authentication Protocol [PAP]). PPP relies on two protocols: Link Control Protocol (LCP) and Network Control Protocol (NCP).

RIP—Routing Information Protocol. A version of Interior Gateway Protocol (IGP) that is supplied with UNIX Berkeley Standard Distribution (BSD) systems. Routing Information Protocol (RIP) is the most common IGP in the Internet. It uses hop count as a routing metric.

Virtual bundle interface—An interface that represents the master link of a bundle. It is not tied to any physical interface. Data going over the bundle is transmitted and received through the master link.

WFQ—weighted fair queueing. A congestion management algorithm that identifies conversations (in the form of traffic streams), separates packets that belong to each conversation, and ensures that capacity is shared fairly among the individual conversations. WFQ is an automatic way of stabilizing network behavior during congestion and results in improved performance and reduced retransmission.

WRED—weighted random early detection. A queueing method that ensures that high-precedence traffic has lower loss rates than other traffic during times of congestion.



CHAPTER 6

6PE Multipath

The 6PE multipath feature uses multiprotocol internal BGP (MP-iBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route

Information About 6PE Multipath

6PE Multipath

Internal and external BGP multipath for IPv6 allows the IPv6 device to load balance between several paths (for example, the same neighboring autonomous system or subautonomous system, or the same metric) to reach its destination. The 6PE multipath feature uses MP-iBGP to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route.

When MP-iBGP multipath is enabled on the 6PE device, all labeled paths are installed in the forwarding table with MPLS information (label stack) when MPLS information is available. This functionality enables 6PE to perform load balancing.

How to Configure 6PE Multipath

Configuring IBGP Multipath Load Sharing

Perform this task to configure IBGP multipath load sharing and control the maximum number of parallel IBGP routes that can be installed in a routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast**]
5. **maximum-paths ibgp** *number-of-paths*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv6** [**unicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 5	**maximum-paths ibgp** *number-of-paths* **Example:** `Device(config-router)# maximum-paths ibgp 3`	Controls the maximum number of parallel IBGP routes that can be installed in a routing table.

Configuration Examples for 6PE Multipath

Example: Configuring 6PE Multipath

```
Device# show ipv6 cef internals
IPv6 CEF is enabled and running
Slow processing intvl = 1 seconds backoff level current/max 0/0
0 unresolved prefixes, 0 requiring adjacency update
IPv6 CEF default table
14 prefixes tableid 0
table version 17
```

```
root 6283F5D0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Master Commands List, All Releases
IPv6 commands	IPv6 Command Reference
Cisco IOS IPv6 features	IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

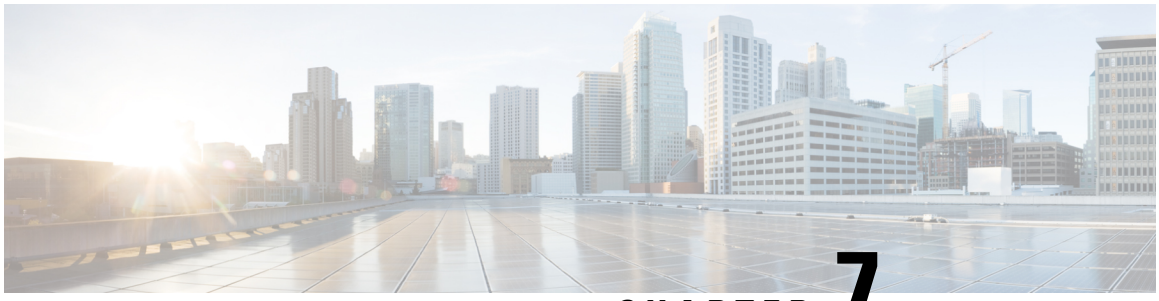
Feature Information for 6PE Multipath

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for 6PE Multipath

Feature Name	Releases	Feature Information
6PE Multipath		The 6PE multipath feature uses MP-iBGP to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route. The following commands were introduced or modified: **maximum-paths ibgp**, **router bgp**, **show ipv6 cef internals**.



CHAPTER 7

IPv6 Switching: Provider Edge Router over MPLS

Multiprotocol Label Switching (MPLS) is deployed by many service providers in their IPv4 networks. Service providers want to introduce IPv6 services to their customers, but changes to their existing IPv4 infrastructure can be expensive and the cost benefit for a small amount of IPv6 traffic does not make economic sense. Several integration scenarios have been developed to leverage an existing IPv4 MPLS infrastructure and add IPv6 services without requiring any changes to the network backbone. This document describes how to implement IPv6 over MPLS.

Prerequisites for IPv6 Switching: Provider Edge Router over MPLS

Before the IPv6 Provider Edge Router over MPLS (6PE) feature can be implemented, MPLS must be running over the core IPv4 network. If Cisco devices are used, Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled for both IPv4 and IPv6 protocols. This module assumes that you are familiar with MPLS.

Information About IPv6 Switching: Provider Edge Router over MPLS

Benefits of Deploying IPv6 over MPLS Backbones

IPv6 over MPLS backbones enables isolated IPv6 domains to communicate with each other over an MPLS IPv4 core network. This implementation requires only a few backbone infrastructure upgrades and no reconfiguration of core devices because forwarding is based on labels rather than the IP header itself, providing a very cost-effective strategy for the deployment of IPv6.

Additionally, the inherent Virtual Private Network (VPN) and MPLS traffic engineering (MPLS-TE) services available within an MPLS environment allow IPv6 networks to be combined into IPv4 VPNs or extranets over an infrastructure supporting IPv4 VPNs and MPLS-TE.

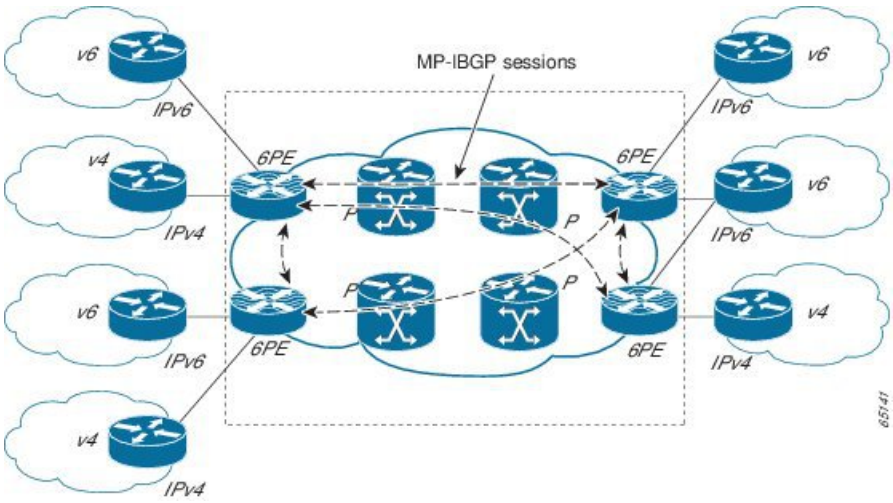
IPv6 on the Provider Edge Devices

The Cisco implementation of IPv6 Provider Edge Router over MPLS is called 6PE, and it enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS label switched paths (LSPs). This feature relies on multiprotocol Border Gateway Protocol (BGP) extensions in the IPv4 network configuration on the provider edge (PE) device to exchange IPv6 reachability information in addition to an MPLS label for each IPv6 address prefix to be advertised. Edge devices are configured to be dual stack running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange.

A hierarchy of labels is imposed on the 6PE ingress device to keep the IPv6 traffic transparent to all the core devices. The top label provides connectivity inside the IPv4 MPLS core network and the label is distributed by Label Distribution Protocol (LDP), Tag Distribution Protocol (TDP), or Resource Reservation Protocol (RSVP). TDP and LDP can both be used for label distribution, but RSVP is used only in the context of MPLS-TE label exchange. The bottom label, automatically assigned to the IPv6 prefix of the destination, is distributed by multiprotocol BGP and used at each 6PE egress device for IPv6 forwarding.

In the figure below the 6PE devices are configured as dual stack devices able to route both IPv4 and IPv6 traffic. Each 6PE device is configured to run LDP, TDP, or RSVP (if traffic engineering is configured) to bind the IPv4 labels. The 6PE devices use multiprotocol BGP to exchange reachability information with the other 6PE devices within the MPLS domain, and to distribute IPv6 labels between them. All 6PE and core devices--P devices in Figure 3--within the MPLS domain share a common IPv4 Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Integrated Intermediate System-to-Intermediate System (IS-IS).

Figure 6: 6PE Device Topology



The interfaces on the 6PE devices connecting to the CE device can be configured to forward IPv6 traffic, IPv4 traffic, or both types of traffic depending on the customer requirements. 6PE devices advertise IPv6 reachability information learned from their 6PE peers over the MPLS cloud. Service providers can delegate an IPv6 prefix from their registered IPv6 prefixes over the 6PE infrastructure; otherwise, there is no impact on the CE device.

The P devices in the core of the network are not aware that they are switching IPv6 packets. Core devices are configured to support MPLS and the same IPv4 IGP as the PE devices to establish internal reachability inside

the MPLS cloud. Core devices also use LDP, TDP, or RSVP for binding IPv4 labels. Implementing the Cisco 6PE feature does not have any impact on the MPLS core devices.

Within the MPLS network, IPv6 traffic is forwarded using label switching, making the IPv6 traffic transparent to the core of the MPLS network. No IPv6 over IPv4 tunnels or Layer 2 encapsulation methods are required.

How to Deploy IPv6 Switching: Provider Edge Router over MPLS

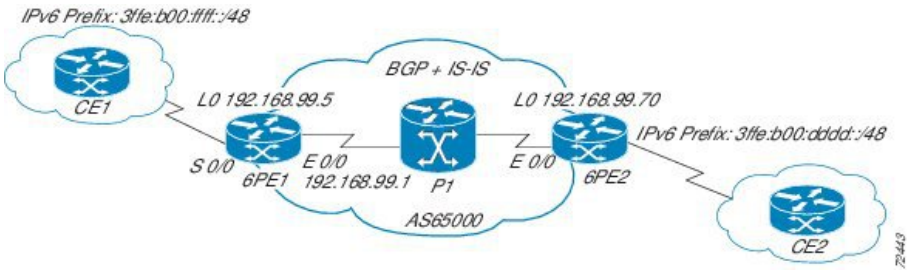
Deploying IPv6 on the Provider Edge Devices (6PE)

Specifying the Source Address Interface on a 6PE Device

Two configuration tasks using the network shown in the figure below are required at the 6PE1 device to enable the 6PE feature.

The customer edge device--CE1 in the figure below--is configured to forward its IPv6 traffic to the 6PE1 device. The P1 device in the core of the network is assumed to be running MPLS, a label distribution protocol, an IPv4 IGP, and Cisco Express Forwarding or distributed Cisco Express Forwarding, and does not require any new configuration to enable the 6PE feature.

Figure 7: 6PE Configuration Example



Before you begin

- The 6PE devices--the 6PE1 and 6PE2 devices in the figure below--must be members of the core IPv4 network. The 6PE device interfaces attached to the core network must be running MPLS, the same label distribution protocol, and the same IPv4 IGP, as in the core network.
- The 6PE devices must also be configured to be dual stack to run both IPv4 and IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **ipv6 cef**
5. **interface** *type number*
6. **ipv6 address** *ipv6-address / prefix-length | prefix-name sub-bits / prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**ipv6 cef** **Example:** `Device(config)# ipv6 cef`	Enables IPv6 Cisco Express Forwarding.
Step 5	**interface** *type number* **Example:** `Device(config)# interface`	Specifies an interface type and number and enters interface configuration mode. • In the context of this feature, the interface to be configured is the interface communicating with the CE device.
Step 6	**ipv6 address** *ipv6-address / prefix-length | prefix-name sub-bits / prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:FFFF::2/64`	Configures an IPv6 address based on an IPv6 general prefix and enable IPv6 processing on an interface.

Binding and Advertising the 6PE Label to Advertise Prefixes

Perform this task to enable the binding and advertising of labels when advertising IPv6 prefixes to a specified BGP neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*

7. **address-family ipv6** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name*| *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** command unless you configure the **no bgp default ipv4-unicast**command before configuring the **neighbor remote-as** command.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.99.70 remote-as 65000`	Adds the IP address of the neighbor in the specified autonomous system to the BGP neighbor table of the local device.
Step 6	**neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.99.70 update-source Loopback 0`	Specifies the interface whose IPv4 address is to be used as the source address for the peering. • In the context of this task, the interface must have an IPv4 address with a 32-bit mask configured. Use of a loopback interface is recommended. This address is used to determine the IPv6 next hop by the peer 6PE.
Step 7	**address-family ipv6** [**unicast**] **Example:**	Specifies the IPv6 address family and enters address family configuration mode.

	Command or Action	Purpose
	`Device(config-router)# address-family ipv6`	• The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 8	**neighbor** {*ip-address* | *peer-group-name*| *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.99.70 activate`	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device.
Step 9	**neighbor** {*ip-address* | *ipv6-address*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.99.70 send-label`	Advertises the capability of the device to send MPLS labels with BGP routes. • In IPv6 address family configuration mode this command enables binding and advertisement of labels when advertising IPv6 prefixes in BGP.

Configuring IBGP Multipath Load Sharing

Perform this task to configure IBGP multipath load sharing and control the maximum number of parallel IBGP routes that can be installed in a routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast**]
5. **maximum-paths ibgp** *number-of-paths*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv6** [**unicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 5	**maximum-paths ibgp** *number-of-paths* **Example:** `Device(config-router)# maximum-paths ibgp 3`	Controls the maximum number of parallel IBGP routes that can be installed in a routing table.

Configuration Examples for IPv6 Switching: Provider Edge Router over MPLS

Example: Provider Edge Device

The 6PE device is configured for both IPv4 and IPv6 traffic. Gigabit Ethernet interface 0/0/0 is configured with an IPv4 address and is connected to a device in the core of the network. Integrated IS-IS and TDP configurations on this device are similar to the P1 device.

Device 6PE1 exchanges IPv6 routing information with another 6PE device using internal BGP (IBGP) established over an IPv4 connection so that all the **neighbor** commands use the IPv4 address of the 6PE2 device. All the BGP peers are within autonomous system 65000, so synchronization with IGP is turned off for IPv4. In IPv6 address family configuration mode, synchronization is disabled by default.

IPv6 and Cisco Express Forwarding for IPv6 are enabled, the 6PE2 neighbor is activated, and label binding and advertisement is enabled for IPv6 prefixes using the **neighbor send-label** command. Connected and static IPV6 routes are redistributed using BGP.

Note MPLS is not supported on IPv6.

Example: Core Device

In the following example, the device in the core of the network is running MPLS, IS-IS, and IPv4 only. The Gigabit Ethernet interfaces are configured with IPv4 address and are connected to the 6PE devices. IS-IS is

the IGP for this network and the P1 and 6PE devices are in the same IS-IS area 49.0001. Tag Distribution Protocol (TDP) and tag switching are enabled on both the Gigabit Ethernet interfaces. Cisco Express Forwarding is enabled in global configuration mode.

```
ip cef
!
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 192.168.99.200 255.255.255.255
!
interface GigabitEthernet0/0/0
 description to_6PE1
 ip address 192.168.99.2 255.255.255.252
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/1/0
 description to_6PE2
 ip address 192.168.99.66 255.255.255.252
 ip router isis
 tag-switching ip
router isis
 passive-interface Loopback0
 net 49.0001.1921.6809.9200.00
```

Example: Monitoring 6PE

In the following example, output information about an IPv6 route is displayed using the **show bgp ipv6** command with an IPv6 prefix:

```
Device# show bgp ipv6 2001:DB8:DDDD::/48

BGP routing table entry for 2001:DB8:DDDD::/48, version 15
Paths: (1 available, best #1, table Global-IPv6-Table)
  Not advertised to any peer
  Local
    ::FFFF:192.168.99.70 (metric 20) from 192.168.99.70 (192.168.99.70)
      Origin IGP, localpref 100, valid, internal, best
```

In the following example, output information about a BGP peer including the IPv6 label capability is displayed using the **show bgp ipv6 neighbors** command with an IP address:

```
Device# show bgp ipv6 neighbors 192.168.99.70

BGP neighbor is 192.168.99.70, remote AS 65000, internal link
  BGP version 4, remote router ID 192.168.99.70
  BGP state = Established, up for 00:05:17
  Last read 00:00:09, hold time is 0, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received
    Address family IPv6 Unicast: advertised and received
    ipv6 MPLS Label capability: advertised and received
  Received 54 messages, 0 notifications, 0 in queue
  Sent 55 messages, 1 notifications, 0 in queue
  Default minimum time between advertisement runs is 5 seconds

 For address family: IPv6 Unicast
  BGP table version 21880, neighbor version 21880
  Index 1, Offset 0, Mask 0x2
  Route refresh request: received 0, sent 0
```

```
  77 accepted prefixes consume 4928 bytes
  Prefix advertised 4303, suppressed 0, withdrawn 1328
  Number of NLRIs in the update sent: max 1, min 0
```

In the following example, output information linking the MPLS label with prefixes is displayed using the **show mpls forwarding-table** command. If the 6PE feature is configured, the labels are aggregated because there are several prefixes for one local label, and the prefix column contains IPv6 instead of a target prefix.

```
Device# show mpls forwarding-table

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  10.1.1.1/32       0            Et0/0      10.0.0.1
18         No Label   nh-id(1)         0             Et2/0      10.0.2.2
19         No Label   nh-id(2)         0             Et1/0      10.0.1.2
20         No Label   nh-id(3)         0             Et1/0      10.0.1.2
22         No Label   nh-id(5)         0             Et1/0      10.0.1.2
24         No Label   nh-id(5)         0             Et2/0      10.0.2.2
```

In the following example, output information about the top of the stack label with label switching information is displayed using the **show bgp ipv6** command with the **labels** keyword:

```
Device# show bgp ipv6 labels

Network              Next Hop               In tag/Out tag
2001:DB8:DDDD::/64  ::FFFF:192.168.99.70   notag/20
```

In the following example, output information about labels from the Cisco Express Forwarding table is displayed using the **show ipv6 cef** command with an IPv6 prefix:

```
Device# show ipv6 cef 2001:DB8:DDDD::/64

2001:DB8:DDDD::/64
      nexthop ::FFFF:192.168.99.70
      fast tag rewrite with Se0/0, point2point, tags imposed {19 20}
```

In the following example, output information from the IPv6 routing table is displayed using the **show ipv6 route** command. The output shows the IPv6 MPLS virtual interface as the output interface of IPv6 routes forwarded across the MPLS cloud. This example shows output from the 6PE1 router.

The 6PE2 router has advertised the IPv6 prefix of 2001:DB8:dddd::/48 configured for the CE2 router and the next-hop address is the IPv4-compatible IPv6 address ::ffff:192.168.99.70, where 192.168.99.70 is the IPv4 address of the 6PE2 router.

```
Device# show ipv6 route

IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
B 2001:DB8:DDDD::/64 [200/0]
  via ::FFFF:192.168.99.70, IPv6-mpls
B 2001:DB8:DDDD::/64 [200/0]
  via ::FFFF:192.168.99.70, IPv6-mpls
L 2001:DB8:FFFF::1/128 [0/0]
  via ::, GigabitEthernet0/0/0
C 2001:DB8:FFFF::/64 [0/0]
  via ::, GigabitEthernet0/0/0
S 2001:DB8:FFFF::/48 [1/0]
  via 2001:DB8:B00:FFFF::2, GigabitEthernet0/0/0
```

Additional References for IPv6 Switching: Provider Edge Router over MPLS

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Master Commands List, All Releases
IPv6 commands	IPv6 Command Reference
Cisco IOS IPv6 features	IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Switching: Provider Edge Router over MPLS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

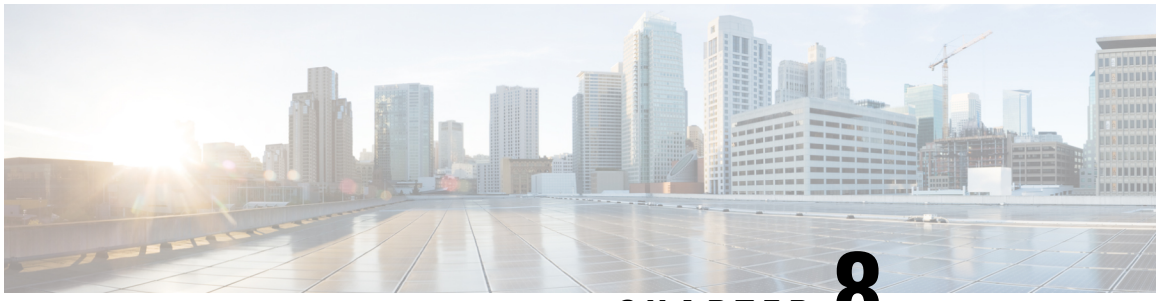
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for IPv6 Switching: Provider Edge Router over MPLS

Feature Name	Releases	Feature Information
IPv6 Switching: Provider Edge Router over MPLS	Cisco IOS XE Amsterdam 17.3.2	The Cisco implementation of IPv6 Provider Edge Router over MPLS enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS LSPs.

PART II

MPLS Embedded Management



CHAPTER 8

MPLS Enhancements to Interfaces MIB

This document describes the Multiprotocol Label Switching (MPLS) enhancements to the existing Interfaces MIB (RFC 2233) to support an MPLS layer. This layer provides counters and statistics specifically for MPLS.

Prerequisites for MPLS Enhancements to Interfaces MIB

- Simple Network Management Protocol (SNMP) must be installed and enabled on the label switching routers (LSRs)
- MPLS must be enabled on the LSRs
- MPLS IP must be enabled on an interface or an MPLS traffic engineering (TE) tunnel enabled on an interface

Restrictions for MPLS Enhancements to Interfaces MIB

- Link up and link down traps for the MPLS layer are not supported in this release.
- Write capability using the SNMP SET command is not supported for the MPLS layer in this release.
- Some counters, including discard and multicast, increment on the underlying physical layer; therefore, they equal 0 because they never reach the MPLS layer.
- The high-capacity counters for the MPLS layer interfaces of the Interfaces MIB contain 64 bits of counter data. In previous versions, the high capacity counters displayed 32 bits of counter data.

The following MIB objects are affected:

- • ifHCInOctets
  • ifHCOutOctets
  • ifHCInUcastPkts
  • ifHCOutUcastPkts

When the 64-bit values are less than the value of 232, the 32-bit and 64-bit values are identical.

After the counter increases to more than 232, the counters are different; the 64-bit value is computed by the following formula:

X * (232) + Y

where:

- • X is the number of times the 32-bit counter has rolled.
  • Y is the residual value of the counter after the roll occurred. The Y value equals the 32-bit value.

When the high-capacity counter values are compared to their 32-bit values, there is a period of time that the counter values are not equal. The 64-bit values lag the 32-bit values when the counters poll the 32-bit hardware counters and computing the correct counter value. During the polling and computation interval, the following high-capacity counter values counters might be inconsistent:

- • ifInOctets
  • ifOutOctets
  • ifInUcastPkts
  • ifOutUcastPkts

The inconsistent values can occur if traffic is constantly flowing over an interface and a MIB walk is performed. The 32-bit value is correct at that moment. The 64-bit value lags slightly, because of the polling computations needed to generate it. Once traffic stops flowing over the interface, and a polling period has passed, the two counters are identical and correct.

The lag time depends on the following factors:

- • The polling interval used by the Interfaces MIB. The less time the polling interval takes, the more accurate the value is.
  • The size of the Interfaces MIB. A large MIB takes a long time to walk and might affect the values found at that instant.
  • The number of computations needed to generate the 64-bit value. The number of MPLS-enabled interfaces increases the number of 64-bit counter values that need to be computed.

Information About MPLS Enhancements to Interfaces MIB

Feature Design of the MPLS Enhancements to Interfaces MIB

The Interfaces MIB (IF MIB) provides an SNMP-based method for managing interfaces. Each entry in the IF MIB establishes indexing, statistics, and stacking relationships among underlying physical interfaces, subinterfaces, and Layer 2 protocols that exist within Cisco software.

The enhancements add an MPLS layer to the IF MIB as a Layer 2 protocol to provide statistics for traffic encapsulated as MPLS on an interface. In this structure, MPLS-specific data such as MPLS-encapsulated

traffic counters and the MPLS maximum transmission unit (MTU) resides on top of the underlying physical or virtual interface to allow separation from non-MPLS data.

The enhancements also allow you to display indexing, statistics, and stacking relationships using the ifStackTable. MPLS layer interfaces are stacked above the underlying physical or virtual interface that is actually forwarding the MPLS traffic. MPLS traffic engineering tunnels are then stacked above those MPLS layers.

The IF MIB supports several types of interfaces. A virtual interface that provides protocol statistics for MPLS-encapsulated traffic has been added. This interface is stacked above real Cisco interfaces or subinterfaces, such as Fast Ethernet (fe0/1/0) or ATM (at1/1.1).

Cisco software creates a corresponding MPLS layer above each interface capable of supporting MPLS when the MPLS encapsulation is enabled by issuing the **mpls ip** command in interface configuration mode.

You can also create the interface layer if you enable MPLS TE by using the **mpls traffic-eng tunnels** command in interface configuration mode.

Note You must also issue these commands in global configuration mode for MPLS IP or MPLS TE to be enabled.

An IF MIB entry is created when you enable either MPLS IP or MPLS TE tunnels on an interface; the entry is removed when you disable both MPLS IP and MPLS TE.

ifStackTable Objects

The table below defines the ifStackTable objects.

Table 7: ifStackTable Objects and Definitions

Object	Definition
ifStackHigherLayer	The value of ifIndex corresponding to the higher sublayer of the relationship; that is, the sublayer that runs on top of the sublayer identified by the corresponding instance of the ifStackLowerLayer. **Note** Index objects are not accessible in a MIB walk. This value is part of the object identifier (OID) for every object in the ifStackTable.
ifStackLowerLayer	The value of ifIndex corresponding to the lower sublayer of the relationship; that is, the sublayer that runs below the sublayer identified by the corresponding instance of the ifStackHigherLayer. **Note** Index objects are not accessible in a MIB walk. This value is part of the OID for every object in the ifStackTable.
ifStackStatus	Used to create and delete rows in the ifStackTable; status is always active(1) for MPLS.

ifRcvAddressTable Objects

The table below defines the ifRcvAddressTable objects.

Note Entries for the MPLS layer do not appear in the ifRcvAddressTable.

Table 8: ifRcvAddressTable Objects and Descriptions

Object	Definition
ifRcvAddressAddress	An address for which the system accepts packets and frames on this entry's interface. **Note** Index objects are not accessible in a MIB walk. This value is part of the OID for every object in the ifRcvAddressTable.
ifRcvAddressStatus	Used to create and delete rows in the ifRcvAddressTable.
ifRcvAddressType	Type of storage used for each entry in the ifRcvAddressTable.

Interfaces MIB Scalar Objects

The IF MIB supports the following scalar objects:

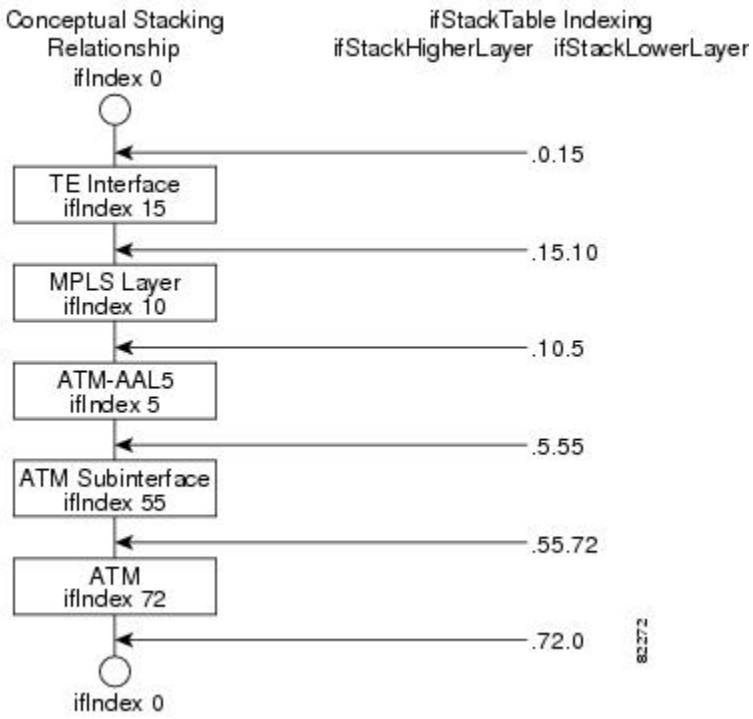
- ifStackLastChange--The value of sysUpTime at the time of the last change of the entire interface stack. A change of the interface stack is defined to be any creation, deletion, or change in value of any instance of ifStackStatus. If the interface stack has been unchanged since the last reinitialization of the local network management subsystem, then this object contains a zero value.
- ifTableLastChange--The value of sysUpTime at the time of the last creation or deletion of an entry in the ifTable. If the number of entries has been unchanged since the last reinitialization of the local network management subsystem, then this object contains a zero value.

Stacking Relationships for MPLS Layer Interfaces

The ifStackTable within the IF MIB provides a conceptual stacking relationship between the interfaces and subinterfaces represented as entries in the ifTable.

The ifStackTable is indexed like a linked list. Each entry shows a relationship between two interfaces providing the ifIndexes of the upper and the lower interface. The entries chain together to show the entire stacking relationship. Each entry links with one another until the stack terminates with an ifIndex of 0 at the highest and lowest ends of the stack. For example, in the figure below, the indexes .10.5 show that ifIndex 10 is stacked upon ifIndex 5. There are 0 entries at the highest and lowest ends of the stack; in the figure, the indexes .0.15 and .72.0 are the highest and lowest ends of the stack, respectively.

Figure 8: Sample ATM Stacking Relationship in the ifStackTable



The table below describes the indexing of the ifStackTable for the layer relationships shown in the figure above.

Note The order of the entries in the table may not be the same as that seen in the MIB walk, which has to follow SNMP ordering rules.

Table 9: Layer Relationships

Layer Relationship (in Descending Order)	ifStackHigherLayer/ifStackLowerLayer
TE interface as top layer	.0.15
TE interface stacked upon MPLS layer	.15.10
MPLS layer stacked upon ATM-AAL5	.10.5
ATM-AAL5 layer stacked upon ATM subinterface	.5.55
ATM subinterface stacked upon ATM	.55.72
ATM as bottom layer	.72.0

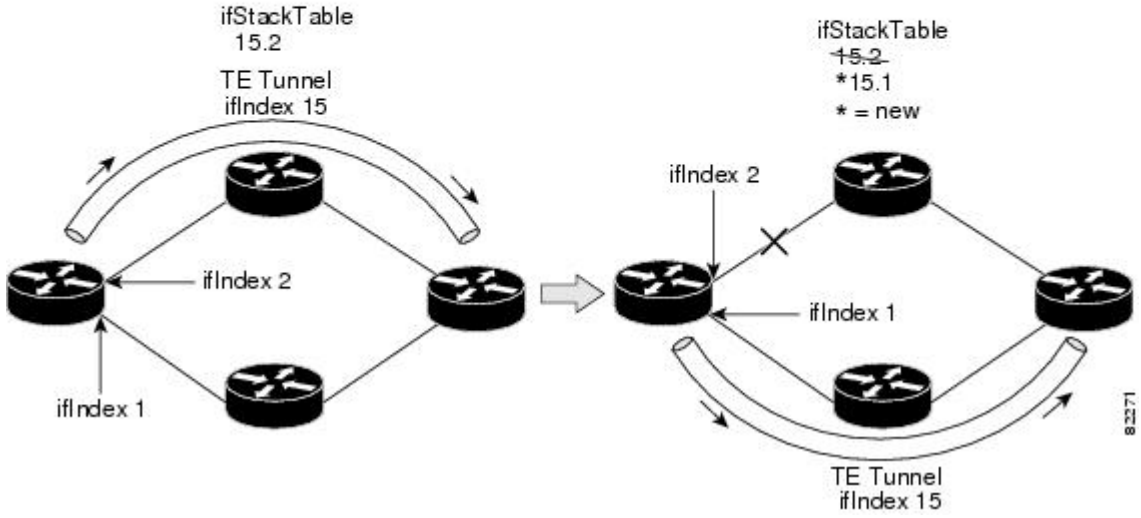
Stacking Relationships for Traffic Engineering Tunnels

MPLS TE tunnels are represented in Cisco software and the IF MIB as virtual interfaces. When properly signaled, TE tunnels pass traffic through MPLS over a physical interface. This process dictates that a TE tunnel is to be stacked on an MPLS layer that is stacked on an underlying interface.

TE tunnels can also change paths in response to different error or network conditions. These changes are instigated by using the RSVP-TE signaling protocol. When a change occurs, a tunnel can switch to a different MPLS interface. If no signaling path exists, no paths will be chosen and thus no MPLS interface will be used.

Because a TE tunnel is represented as an IF MIB ifTable entry, the ifStackTable also contains an entry corresponding to the TE tunnel. If the TE tunnel is successfully signaled, the ifStackTable also contains a link between the tunnel interface and one MPLS interface. Note that because it is possible for a TE tunnel to not have a corresponding signaled path, it is thus possible for a TE tunnel's ifStackTable entry to not have a corresponding lower layer. In this case, the lower layer variable contains the value of 0.

The figure below shows a TE tunnel before (left) and after (right) being rerouted and the effect on the ifStackTable. When ifIndex 2 fails, the TE tunnel is rerouted through ifIndex1, the 15.2 entry is removed from the ifStackTable, and the 15.1 entry is added.



MPLS Label Switching Router MIB Enhancements

All of the ifIndex references in the MPLS-LSR-MIB tables have changed from the ifIndex of the underlying physical or virtual interface to the ifIndex of the MPLS layer.

The table below shows the specific changes.

Table 10: MPLS-LSR-MIB ifIndex Objects Enhanced

Table	ifIndex
MPLS interface configuration table (mplsInterfaceConfTable)	mplsInterfaceConfIndex
MPLS in-segment table (mplsInSegmentTable)	mplsInSegmentIfIndex
MPLS cross-connect table (mplsXCTable)	mplsInSegmentIfIndex
MPLS out-segment table (mplsOutSegmentTable)	mplsOutSegmentIfIndex

The following objects from the mplsInterfaceConfTable are affected:

- mplsInterfaceOutPackets--Count only MPLS-encapsulated out packets
- mplsInterfaceInPackets--Count only MPLS-encapsulated in packets

Benefits of the MPLS Enhancements to Interfaces MIB

Improved Accounting Capability

By viewing the MPLS layer, you get MPLS-encapsulated traffic counters that do not include non-MPLS encapsulated traffic (for example, IP packets). Therefore, the counters are more useful for MPLS-related statistics.

TE Tunnel Interfaces

For TE tunnel interfaces, the stacking relationship reflects the current underlying MPLS interface that is in use and dynamically changes as TE tunnels reoptimize and reroute.

MPLS-Specific Information

The MPLS layer shows MPLS-specific information including the following:

- If MPLS is enabled
- MPLS counters
- MPLS MTU
- MPLS operational status

How to Configure MPLS Enhancements to Interfaces MIB

Enabling the SNMP Agent

SUMMARY STEPS

1. **enable**

2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** *number*]
5. **end**
6. **write memory**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** *number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) community strings for the MPLS Label Distribution Protocol (LDP) MIB. • The *string* argument functions like a password, permitting access to SNMP functionality on label switch routers (LSRs) in an MPLS network. • The optional **ro** keyword configures read-only (ro) access to the objects in the MPLS LDP MIB.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.

	Command or Action	Purpose
Step 7	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If you see any snmp-server statements, SNMP has been enabled on the router. If any SNMP information is displayed, you can modify the information or change it as desired.

Configuration Examples for the MPLS Enhancements to Interfaces MIB

MPLS Enhancements to Interfaces MIB: Examples

The following example shows how to enable an SNMP agent:

```
Router# configure terminal
Router(config)# snmp-server community
```

In the following example, SNMPv1 and SNMPv2C are enabled. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*.

```
Router(config)# snmp-server community public
```

In the following example, read-only access is allowed for all objects to members of access list 4 that specify the comaccess community string. No other SNMP managers have access to any objects.

```
Router(config)# snmp-server community comaccess ro 4
```

Additional References

Related Documents

Related Topic	Document Title
SNMP commands	*Cisco IOS Network Management Command Reference*
SNMP configuration	"Configuring SNMP Support" in the *Network Management Configuration Guide.*
A description of SNMP agent support for the MPLS Traffic Engineering MIB (MPLS TE MIB)	MPLS Traffic Engineering (TE) MIB

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
Interfaces Group MIB (IF MIB)	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1156	*Management Information Base for Network Management of TCP/IP-based internets*
RFC 1157	*A Simple Network Management Protocol (SNMP)*
RFC 1213	*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*
RFC 1229	*Extensions to the Generic-Interface MIB*
RFC 2233	*Interfaces MIB*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Enhancements to Interfaces MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for MPLS Enhancements to Interfaces MIB

Feature Name	Releases	Feature Information
MPLS Enhancements to Interfaces MIB	12.0(23)S 12.3(8)T 12.2(33)SRA 12.2(33)SXH 12.2(33)SB Cisco IOS XE Release 2.1	This document describes the Multiprotocol Label Switching (MPLS) enhancements to the existing Interfaces MIB (RFC 2233) to support an MPLS layer. This layer provides counters and statistics specifically for MPLS. In Cisco IOS Release 12.0(23)S, this feature was introduced. This feature was integrated into Cisco IOS Release 12.3(8)T. This feature was integrated into Cisco IOS Release 12.2(33)SRA. This feature was integrated into Cisco IOS Release 12.2(33)SXH. This feature was integrated into Cisco IOS Release 12.2(33)SB. In Cisco IOS XE Release 2.1, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced or modified: **snmp-server community**.

Glossary

ATM -- Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3.

ATM-AAL5 --ATM adaptation layer 5. One of four AALs recommended by the ITU-T. AAL5 supports connection-oriented variable bit rate (VBR) services and is used predominantly for the transfer of classical IP over ATM and LAN emulation (LANE) traffic. AAL5 uses simple and efficient AAL (SEAL) and is the least complex of the current AAL recommendations. It offers low bandwidth overhead and simpler processing requirements in exchange for reduced bandwidth capacity and error-recovery capability.

encapsulation -- Wrapping of data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when bridging dissimilar networks, the entire frame from one network is simply placed in the header used by the data link layer protocol of the other network.

IETF --Internet Engineering Task Force. A task force (consisting of more than 80 working groups) that is developing standards for the Internet and the IP suite of protocols.

interface --The boundary between adjacent layers of the ISO model.

label --A short, fixed-length identifier that is used to determine the forwarding of a packet.

label switching--A term used to describe the forwarding of IP (or other network layer) packets using a label swapping algorithm based on network layer routing algorithms. The forwarding of these packets uses the exact match algorithm and rewrites the label.

LSR --label switching router. A device that forwards Multiprotocol Label Switching (MPLS) packets based on the value of a fixed-length label encapsulated in each packet.

MIB --Management Information Base. A database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP). The value of a MIB object can be changed or retrieved by means of SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

MPLS interface--An interface on which Multiprotocol Label Switching (MPLS) traffic is enabled.

MTU --maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

NMS --network management system. System responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. NMSs communicate with agents to help keep track of network statistics and resources.

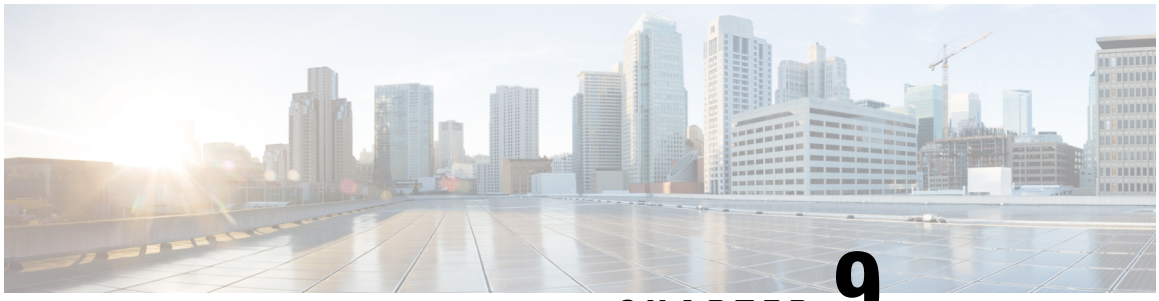
OID --object identifier. Values are defined in specific MIB modules. The Event MIB allows you or an NMS to watch over specified objects and to set event triggers based on existence, threshold, and Boolean tests. An event occurs when a trigger is fired; this means that a specified test on an object returns a value of true. To create a trigger, you or a network management system (NMS) configures a trigger entry in the mteTriggerTable of the Event MIB. This trigger entry specifies the OID of the object to be watched. For each trigger entry type, corresponding tables (existence, threshold, and Boolean tables) are populated with the information required for carrying out the test. The MIB can be configured so that when triggers are activated (fired) either a Simple Network Management Protocol (SNMP) Set is performed, a notification is sent out to the interested host, or both.

SNMP --Simple Network Management Protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices, and for managing configurations, statistics collection, performance, and security.

traffic engineering tunnel--A label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.

trap --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps are less reliable than notification requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received.

tunnel --A secure communication path between two peers, such as routers.



CHAPTER 9

MPLS Label Switching Router MIB

The MPLS Label Switching Router MIB (MPLS-LSR-MIB) allows you to use the Simple Network Management Protocol (SNMP) to remotely monitor a label switch router (LSR) that is using the Multiprotocol Label Switching (MPLS) technology.

Scalability enhancements provided in the Cisco IOS 12.0(28)S release reduce the size of any MIB walk and improve the usability of the MPLS-LSR-MIB.

Note In Cisco IOS Release 12.2(33)SRB and Cisco IOS Release 12.2(33)SB, this MIB has been deprecated and replaced by MPLS-LSR-STD-MIB (RFC 3813). In those two releases and in later images, the entire MIB can be referenced by the name mplsLsrMIB for purposes of the SNMP server excluded/included command. If other MIB object names need to be referenced on the router, they must be referenced by MPLS-LSR-MIB::<table_entry_name>.

Information About MPLS Label Switching Router MIB

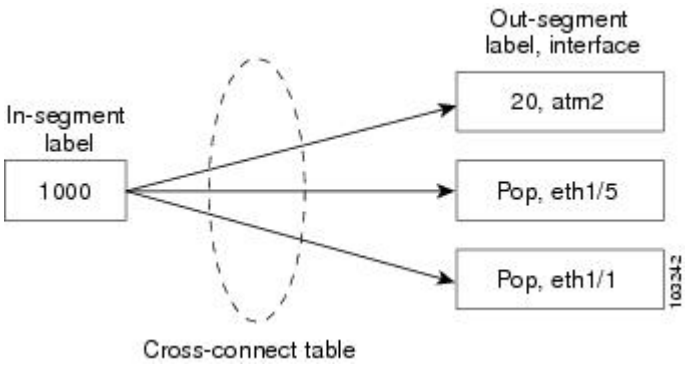
The MPLS-LSR-MIB contains managed objects that support the retrieval of label switching information from a router. The MIB is based on Revision 05 of the IETF MPLS-LSR-MIB. The MPLS-LSR-MIB mirrors a portion of the Cisco MPLS subsystem; specifically, it mirrors the Label Forwarding Information Base (LFIB). This implementation enables a network administrator to get information on the status, character, and performance of the following:

- MPLS-capable interfaces on the LSR
- Incoming MPLS segments (labels) at an LSR and their associated parameters
- Outgoing segments (labels) at an LSR and their associated parameters

In addition, the network administrator can retrieve the status of cross-connect table entries that associate MPLS segments with each other.

The figure below shows the association of the cross-connect table with incoming and outgoing segments (labels).

Figure 9: Label Forwarding with the Cross-Connect Table



Note The out-segment table does not display “no label” entries. Labels that are displayed as “POP” are the special MPLS label 3.

The notation used in the MPLS-LSR-MIB follows the conventions defined in Abstract Syntax Notation One (ASN.1). ASN.1 defines an Open System Interconnection (OSI) language used to describe data types apart from particular computer structures and presentation techniques. Each object in the MIB incorporates a DESCRIPTION field that includes an explanation of the object’s meaning and usage, which, together with the other characteristics of the object (SYNTAX, MAX-ACCESS, and INDEX) provides sufficient information for management application development, as well as for documentation and testing.

The MPLS-LSR-MIB represents an ASN.1 notation reflecting an idealized MPLS LSR.

A network administrator can access the entries (objects) in the MPLS-LSR-MIB by means of any SNMP-based network management system (NMS). The network administrator can retrieve information in the MPLS-LSR-MIB using standard SNMP **get** and **getnext** operations.

Typically, SNMP runs as a low-priority process. The response time for the MPLS-LSR-MIB is expected to be similar to that for other MIBs. The size and structure of the MIB and other MIBs in the system influence response time when you retrieve information from the management database. Traffic through the LSR also affects SNMP performance. The busier the switch is with forwarding activities, the greater the possibility of lower SNMP performance.

MPLS-LSR-MIB Elements

The top-level components of the MPLS-LSR-MIB consist of

- Tables and scalars (mplsLsrObjects)
- Traps (mplsLsrNotifications and mplsLsrNotifyPrefix)
- Conformance (mplsLsrConformance)

This Cisco implementation does not support the notifications defined in the MIB, nor does it support the labelStackTable or the trafficParamTable.

MPLS-LSR-MIB Tables

The Cisco implementation of the MPLS-LSR-MIB supports four main tables:

- Interface configuration
- In-segment
- Out-segment
- Cross-connect

The MIB contains three supplementary tables to supply performance information. This implementation does not support the label stack and traffic parameter tables.

The following sections list the MPLS-LSR-MIB tables (main and supplementary), their functions, table objects that are supported, and table objects that are *not* supported.

MPLS interface configuration table (mplsInterfaceConfTable)

Provides information for each MPLS-capable interface on an LSR.

Supports:

- A unique interface index or zero
- Minimum and maximum values for an MPLS label received on the interface
- Minimum and maximum values for an MPLS label sent from the interface
- A value for an MPLS label sent from the interface
- Per platform (0) or per interface (1) setting
- The storage type

Does not support:

- The total usable bandwidth on the interface
- The difference between the total usable bandwidth and the bandwidth in use

MPLS interface performance table (mplsInterfacePerfTable)

Augments the MPLS interface configuration table.

Supports:

- The number of labels in the incoming direction in use
- The number of top-most labels in outgoing label stacks in use

Does not support:

- The number of top-most labels in outgoing label stacks in use
- The number of labeled packets discarded because no cross-connect entries exist

- The number of outgoing MPLS packets requiring fragmentation for transmission

MPLS in-segment table (mplsInSegmentTable)

Contains a description of incoming segments (labels) at an LSR and their associated parameters.

Administrative and operational status objects for this table control packet transmission. If administrative and operational status objects are down, the LSR does not forward packets. If these status objects are up, the LSR forwards packets.

Supports:

- A unique index identifier
- The incoming label
- The number of labels to pop from the incoming segment
- An address family number from the Internet Assigned Number Authority (IANA)
- A segment cross-connect entry association
- The segment owner
- The storage type
- The administrative status
- The operational status

Note The administrative status and operational status are always up for inSegments in the Cisco implementation. Otherwise, these entries do not appear in the table.

Does not support:

- A pointer to a traffic parameter table entry (set to the default 0.0)

MPLS in-segment performance table (mplsInSegmentPerfTable)

Augments the MPLS in-segment table, providing performance information and counters for incoming segments on an LSR.

Supports:

- The number of 32-bit octets received
- The number of 64-bit octets received
- The time of the last system failure that corresponded to one or more incoming segment discontinuities

Note The lastFailure parameter is set to zero because it has no meaning in the Cisco implementation.

Does not support:

- The total number of packets received
- The number of packets with errors
- The number of labeled packets discarded with no errors

MPLS out-segment table (mplsOutSegmentTable)

Contains a description of outgoing segments from an LSR and their associated parameters.

Administrative and operational status objects for this table control packet transmission. If administrative and operational status objects are down, the LSR does not forward packets. If these values are up, the LSR forwards packets.

Supports:

- A unique index identifier
- An interface index of the outgoing interface
- An indication of whether or not a top label is pushed onto the outgoing packet's label stack
- The label to push onto the outgoing packet's label stack (if the previous value is true)
- The next hop address type
- The IPv4 address of the next hop
- The segment cross-connect entry association
- The segment owner
- The storage type
- The administrative status
- The operational status

Note The administrative and operational status entries are always up in the Cisco implementation. Otherwise, the administrative and operational status entries do not appear in the table.

Does not support:

- An IPv6 address of the next hop
- A pointer to a traffic parameter table entry (set to the default 0.0)

MPLS out-segment performance table (mplsOutSegmentPerfTable)

Augments the MPLS out-segment table, providing performance information and counters for outgoing segments on an LSR.

Supports:

- The number of 32-bit octets sent
- The number of 64-bit octets sent

- The time of the last system failure that corresponded to one or more outgoing segment discontinuities

Does not support:

- The number of packets sent
- The number of packets that could not be sent because of errors
- The number of packets discarded with no errors

MPLS cross-connect table (mplsXCTable)

Associates inSegments (labels) with outSegments (labels) to show the manager how the LSR is currently swapping these labels.

A row in this table consists of one cross-connect entry that is indexed by the cross-connect index, the interface index of the incoming segment, the incoming label, and the out-segment index.

The administrative and operational objects for this table control packet forwarding to and from a cross-connect entry (XCEntry). The administrative status and operational status are always up in the Cisco implementation. Otherwise, the LSR would not forward packets.

Supports:

- A unique index identifier for a group of cross-connect segments
- A label switched path (LSP) to which the cross-connect entry belongs
- An index to the MPLS label stack table that identifies the stack of labels to be pushed under the top label
- An indication whether or not to restore the cross-connect entry after a failure (the default value is false)
- The cross-connect owner
- The storage type
- The administrative status (if up)
- The operational status (if up)

Note The administrative status and operational status are always up in the Cisco implementation. Otherwise, these status entries do not appear in the table.

Does not support:

- Tunnel IDs as label switched path (LSP) ID objects

Information from Scalar Objects

The MPLS-LSR-MIB supports several scalar objects. In the Cisco implementation of the MIB, the following scalar objects are hard-coded to the value indicated and are read-only objects:

- mplsOutSegmentIndexNext (0)--The value for the out-segment index when an LSR creates a new entry in the MPLS out-segment table. The 0 indicates that this is not implemented because modifications to this table are not allowed.

- mplsXCTIndexNext (0)--The value for the cross-connect index when an LSR creates an entry in the MPLS cross-connect table. The 0 indicates that no unassigned values are available.
- mplsMaxLabelDepth(2)--The value for the maximum stack depth.
- mplsLabelStackIndexNext (0)--The value for the label stack index when an LSR creates entries in the MPLS label stack table. The 0 indicates that no unassigned values are available.
- mplsTrafficParamIndexNext (0)--The value for the traffic parameter index when an LSR creates entries in the MPLS traffic parameter table. The 0 indicates that no unassigned values are available.

The following scalar objects do not contain information for the MPLS-LSR-MIB and are coded as false:

- mplsInSegmentTrapEnable (false)--In-segment traps are not sent when this value is false.
- mplsOutSegmentTrapEnable (false)--Out-segment traps are not sent when this value is false.
- mplsXCTrapEnable (false)--Cross-connect traps are not sent when this value is false.

No trap information exists to support the MIB. Therefore, the following traps are not supported:

- mplsInSegmentUp
- mplsInSegmentDown
- mplsOutSegmentUp
- mplsOutSegmentDown
- mplsXCUp
- mplsXCDown

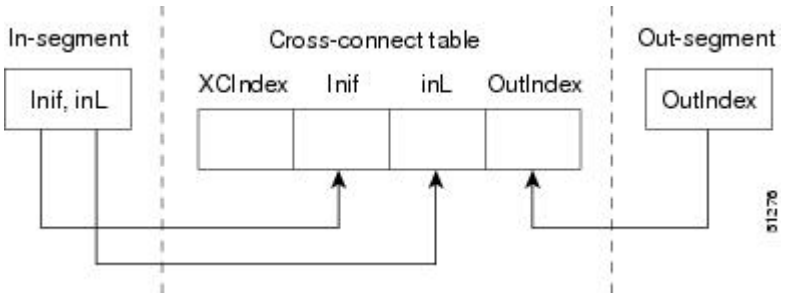
Linking Table Elements

In the cross-connect table, cross-connect entries associate incoming segments and interfaces with outgoing segments and interfaces. The following objects index the cross-connect entry:

- Cross-connect index--A unique identifier for a group of cross-connect entries in the cross-connect table. In the Cisco implementation, this value is always the same as that for the outSegmentIndex, unless there is no label or if the label has been popped.
- Interface index of the in-segment--A unique index for an entry in the in-segment table that represents an incoming MPLS interface. The value 0 means platform wide, for any entries that apply to all interfaces.
- Incoming label--An entry in the in-segment table that represents the label on the incoming packet.
- Out-segment index--A unique identifier for an entry in the out-segment table that contains a top label for the outgoing packet's label stack and an interface index of the outgoing interface.

The figure below shows the links between the in-segment and the out-segment in the cross-connect table.

Figure 10: Cross-Connect Table Links



The table below shows the cross-connect table links you might see in the output from SNMP **get** operations on the MPLS-LSR-MIB objects that index a cross-connect entry. These objects include

- In-Segment Values--mplsInSegmentIfIndex and mplsInSegmentLabel
- Cross-Connect Entry--mplsXCIndex
- Out-Segment Values--mplsOutSegmentIndex

Table 12: MPLS LSR Output Showing Cross-Connect Table Links

In-Segment Values	Cross-Connect Entry	Out-Segment Values
0[1], 1000	500[2], 0, 1000, 0	--
	501, 0, 1000, 501	501 = Pop (topLabel), Eth 1/5
	502, 0, 1000, 502	502 = Pop (topLabel), Eth, 1/1

1 All MPLS-enabled interfaces can receive incoming labels.
2 For this implementation of the MPLS-LSR-MIB, the cross-connect index and the out-segment index are the same. If there is no outsegment, the value will be zero.

Note The OutSegmentIndex object is not the label. The label can be retrieved from the mplsOutSegmentTopLabel object.

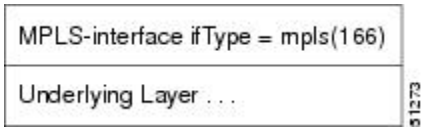
Interface Configuration Table and Interface MIB Links

The MPLS interface configuration table lists interfaces that support MPLS technology. An LSR creates an entry dynamically in this table for each MPLS-capable interface. An interface becomes MPLS-capable when MPLS is enabled on that interface. A non-zero index for an entry in this table points to the ifIndex for the corresponding interface entry in the MPLS-layer in the ifTable of the Interfaces Group MIB.

The ifTable contains information on each interface in the network. Its definition of an interface includes any sublayers of the internetwork layer of the interface. MPLS interfaces fit into this definition of an interface. Therefore, each MPLS-enabled interface is represented by an entry in the ifTable.

The interrelation of entries in the ifTable is defined by the interfaces stack group of the Interfaces Group MIB. The figure below shows how the stack table might appear for MPLS interfaces. The underlying layer refers to any interface that is defined for MPLS internetworking, for example, ATM, Frame Relay, or Ethernet.

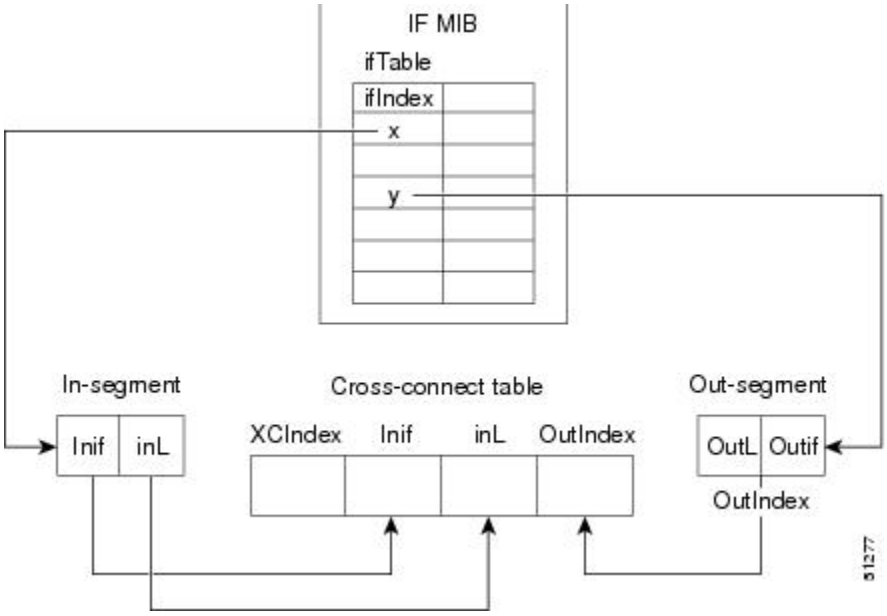
Figure 11: Interface Group MIB Stack Table for MPLS Interfaces



Note Tunnel interfaces are included in the MPLS list for the current implementation.

The incoming and outgoing packets include a reference to the interface index for the ifTable of the Interfaces Group MIB. The figure below shows the links between MPLS-LSR-MIB objects and the Interfaces Group MIB.

Figure 12: MPLS-LSR-MIB and Interfaces Group MIB Links



- For the Interfaces Group MIB (IF MIB):
  - ifTable represents the MPLS interface table.
  - ifIndex represents the index to an entry in the MPLS interface table.
- For the In-segment:
  - Inif represents the interface on the incoming segment (references an index entry in the ifTable).
  - inL represents the label on the incoming segment.
- For the Out-segment:
  - OutL represents the label on the outgoing segment.
  - Outif represents the interface on the outgoing segment (references an index entry in the ifTable).

- For the Cross-connect table:
  - XCIndex represents the index to an entry in the MPLS cross-connect table.
  - Inif represents the interface on the incoming segment.
  - inL represents the MPLS label on the incoming segment.
  - OutIndex represents an index to an entry in the MPLS out-segment table.

Using the MPLS-LSR-MIB

The MPLS-LSR-MIB enables you to display the contents of the MPLS Label Forwarding Information Base (LFIB). It gives you the same information that you can obtain using the CLI command **show mpls forwarding-table**.

However, the MPLS-LSR-MIB approach offers these advantages over the CLI command approach:

- A more efficient use of network bandwidth
- Greater interoperability among vendors
- Greater security (SMNP Version 3)

The following paragraphs describe the MPLS-LSR-MIB structure and show, through the use of an example, how the two approaches to the information display compare.

MPLS-LSR-MIB Structure

MIB structure is represented by a tree hierarchy. Branches along the tree have short text strings and integers to identify them. Text strings describe object names, and integers allow computer software to encode compact representations of the names.

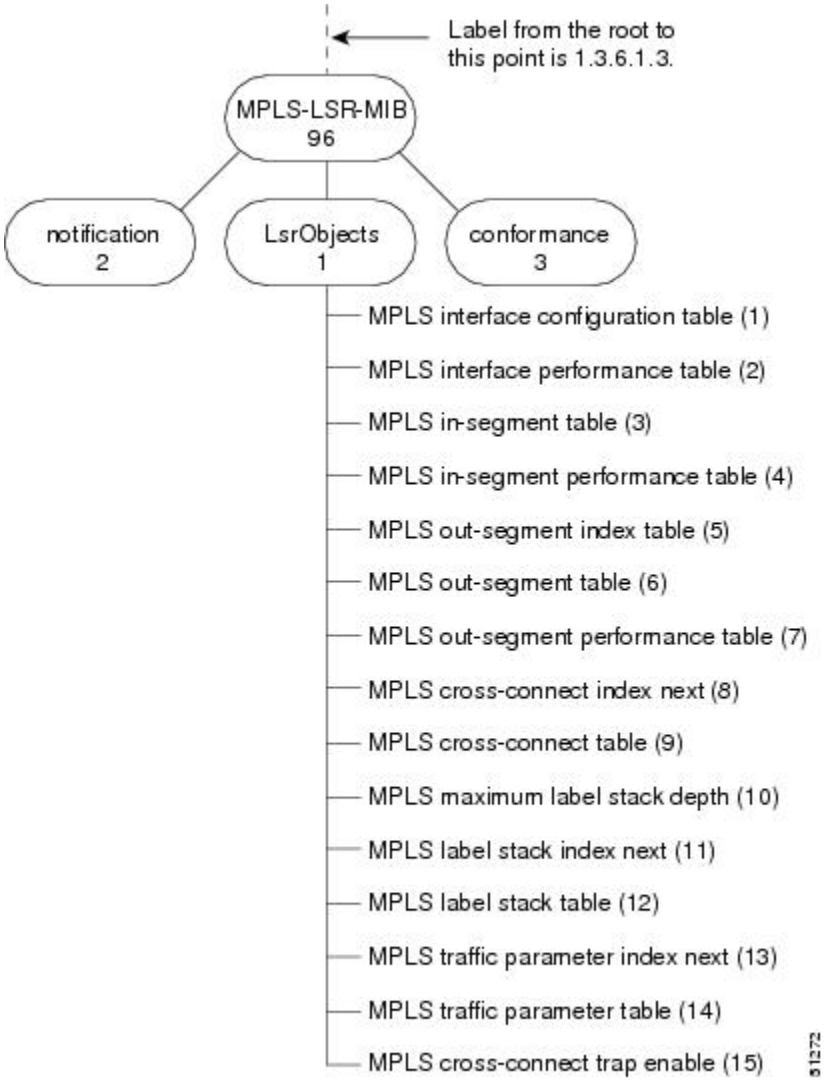
The MPLS-LSR-MIB falls on the experimental branch of the Internet MIB hierarchy. The experimental branch of the Internet MIB hierarchy is represented by the object identifier 1.3.6.1.3. This branch can also be represented by its object name *iso.org.dod.internet.experimental* . The MPLS-LSR-MIB is identified by the object name *mplsLsrMIB* , which is denoted by the number 96. Therefore, objects in the MPLS-LSR-MIB can be identified in either of the following ways:

- The object identifier--1.3.6.1.3.96.[MIB-variable]
- The object name--*iso.org.dod.internet.experimental.mplsLsrMIB.[MIB-variable]*

To display a *MIB-variable* , you enter an SNMP **get** command with an object identifier. Object identifiers are defined by the MPLS-LSR-MIB.

The figure below shows the position of the MPLS-LSR-MIB in the Internet MIB hierarchy.

Figure 13: MPLS-LSR-MIB in the Internet MIB Hierarchy



CLI Commands and the MPLS-LSR-MIB

The MPLS LFIB is the component of the Cisco MPLS subsystem that contains management information for LSRs. You can access this management information by means of either of the following:

- Using the **show mpls forwarding-table** CLI command
- Entering SNMP **get** commands on a network manager

The following examples show how you can gather LSR management information using both methods.

CLI Command Output

A **show mpls forwarding-table** CLI command allows you to look at label forwarding information for a packet on a specific MPLS LSR.

```
Router# show mpls forwarding-table
Local  Outgoing      Prefix            Bytes Tag     Outgoing   Next Hop
Tag    Tag or VC     or Tunnel Id      Switched      interface
19     Pop Tag       10.3.4.0/24       0             Et1/4      10.22.23.23
22     23            14.14.14.14/32    0             AT2/0.1    point2point
       1/36          14.14.14.14/32    0             AT2/0.2    point2point
```

MPLS-LSR-MIB Output

SNMP commands on MIB objects also allow you to look at the label forwarding information for a specific MPLS LSR.

You can do a walk-through of the MIB by running a command such as **getmany -v2c public mplsLsrMIB** on a network manager where **getmany** does repeated SNMP **getnext** operations to retrieve the contents of the MPLS-LSR-MIB.

```
mplsXCOperStatus.9729.0.19.9729 = up(1)
mplsXCOperStatus.11265.0.22.11265 = up(1)
mplsXCOperStatus.11266.0.22.11266 = up(1)
```

You can continue to scan the output of the **getmany** command for the following (from the MPLS out-segment table):

- Out-segment's top label objects (mplsOutSegmentTopLabel)

```
mplsOutSegmentTopLabel.9729 = 3
mplsOutSegmentTopLabel.11265 = 23
mplsOutSegmentTopLabel.11266 = 65572
```

Note 65572 is 1/36 in label form (1 is the high-order 16 bits. 36 is the low-order 16 bits.)

- Out-segment's interface index (mplsOutSegmentIfIndex)

```
mplsOutSegmentIfIndex.9729 = 7
mplsOutSegmentIfIndex.11265 = 28
mplsOutSegmentIfIndex.11266 = 31
```

Benefits

The benefits described in the following paragraphs are available to you with the MPLS-LSR-MIB.

Troubleshooting LSR Problems

By monitoring the cross-connect entries and the associated incoming and outgoing segments, you can see which labels are installed and how they are being swapped. Use the MPLS-LSR-MIB in place of the **show mpls forwarding** CLI command.

Monitoring of LSR Traffic Loads

By monitoring interface and packet operations on an MPLS LSR, you can identify high- and low-traffic patterns, as well as traffic distributions.

Improvement of Network Performance

By identifying potentially high-traffic areas, you can set up load sharing to improve network performance.

Verification of LSR Configuration

By comparing results from SNMP **get** commands and the **show mpls forwarding** CLI command, you can verify your LSR configuration.

Displaying of Active Label Switched Paths

By monitoring the cross-connect entries and the associated incoming segments and outgoing segments, you can determine the active LSPs.

How to Configure the MPLS LSR MIB

Prerequisites

The MPLS-LSR-MIB requires the following:

- SNMP installed and enabled on the LSR
- MPLS enabled on the LSR
- 60K of memory

Note Additional capacity is not required for runtime dynamic random-access memory (DRAM).

Enabling the SNMP Agent

The SNMP agent for the MPLS-LSR-MIB is disabled by default. To enable the SNMP agent, perform the following steps:

SUMMARY STEPS

1. **enable**
2. show running-config
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*]
5. **end**
6. copy running-config startup-config

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show running-config **Example:** `Router# show running-config`	Displays the running configuration of the router to determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) SNMP community strings. This command enables the SNMP agent and permits any SNMP manager to access all objects with read-only permission using the community string public.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	copy running-config startup-config **Example:** `Router# copy running-config startup-config`	Copies the modified SNMP configuration into router NVRAM, permanently saving the SNMP settings. When you are working with Cisco IOS Release 10.3 or earlier, use the **write memory** command.

Verifying That the SNMP Agent Has Been Enabled

To verify that the SNMP agent has been enabled, perform the following steps:

SUMMARY STEPS

1. Access the router through a Telnet session:
2. Enter privileged mode:
3. Display the running configuration and look for SNMP information:

DETAILED STEPS

Step 1 Access the router through a Telnet session:

Example:

```
Prompt# telnet xxx.xxx.xxx.xxx
```

where *xxx.xxx.xxx.xxx* represents the IP address of the target device.

Step 2 Enter privileged mode:

Example:

```
Router# enable
```

Step 3 Display the running configuration and look for SNMP information:

Example:

```
Router# show running-configuration
...
...
snmp-server community public RO
```

If you see any "snmp-server" statements, SNMP has been enabled on the router.

Configuration Examples for the MPLS LSR MIB

The following example shows how to enable an SNMP agent.

```
configure terminal
snmp-server community
```

In the following example, SNMPv1 and SNMPv2C are enabled. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public* .

```
configure terminal
snmp-server community public
```

In the following example, read-only access is allowed for all objects to members of access list 4 that specify the *comaccess* community string. No other SNMP managers have access to any objects.

```
configure terminal
nmp-server community comaccess ro 4
```

Additional References

Related Documents

Related Topic	Document Title
Configuring SNMP using Cisco IOS software	• *Network Management Configuration Guide* . Configuring SNMP Support • Network Management Command Reference, SNMP Commands

Standards

Standard	Title
draft-ietf-mpls-lsr-mib-05.txt	MPLS Label Switch Router Management Information Base Using SMIv2
draft-ietf-mpls-arch-07.txt	Multiprocol Label Switching Architecture

MIBs

MIBs	MIBs Link
• MPLS Label Switching Router MIB (MPLS-LSR-MIB)	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
The LSR implementation supporting the MPLS-LSR-MIB is in full compliance with all provisions of Section 10 of RFC 2026.	*The Internet Standards Process*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content. The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Label Switching Router MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13: Feature Information for MPLS Label Switching Router MIB

Feature Name	Releases	Feature Information
MPLS Label Switching Router MIB	12.0(14)ST 12.2(2)T 12.0(22)S 12.2(14)S 12.2(25)S 12.0(28)S 12.2(33)SRB 12.2(33)SB	The MPLS Label Switching Router MIB (MPLS-LSR-MIB) allows you to use the Simple Network Management Protocol (SNMP) to remotely monitor a label switch router (LSR) that is using the Multiprotocol Label Switching (MPLS) technology. This feature was introduced on Cisco IOS Release 12.0(14)ST. This feature was integrated into Cisco IOS Release 12.2(2)T. This feature was implemented on the Cisco 12000 series routers and integrated into Cisco IOS Release 12.0(22)S. This feature was integrated into Cisco IOS Release 12.2(14)S and implemented on Cisco 7200 and Cisco 7500 series routers. This feature was updated to include scalability enhancements in Cisco IOS Release 12.0(28)S. In Cisco IOS Release 12.2(25)S, this feature was updated to work in the MPLS High Availability environment with the Cisco 7500 series routers. In Cisco IOS Release 12.2(33)SRB, this MIB has been deprecated and replaced by MPLS-LSR-STD-MIB (RFC 3813). In Cisco IOS Release 12.2(33)SB, this MIB has been deprecated and replaced by MPLS-LSR-STD-MIB (RFC 3813). No commands were introduced or modified.

Glossary

cross-connect (XC) --An association of in-segments and incoming Multiprotocol Label Switching (MPLS) interfaces to out-segments and outgoing MPLS interfaces.

IETF --Internet Engineering Task Force. A task force (consisting of more that 80 working groups) that is developing standards for the Internet and the IP suite of protocols.

inSegment --A label on an incoming packet that is used to determine the forwarding of the packet.

Internet Engineering Task Force --See IETF.

label --A short, fixed length identifier that is used to determine the forwarding of a packet.

Label Distribution Protocol --See LDP.

label switched path --See LSP.

label switching --Describes the forwarding of IP (or other network layer) packets by a label swapping algorithm based on network layer routing algorithms. The forwarding of these packets uses the exact match algorithm and rewrites the label.

label switch router --See LSR.

LDP --Label Distribution Protocol. A standard protocol that operates between Multiprotocol Label Switching (MPLS)-enabled routers to negotiate the labels (addresses) used to forward packets. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LSP --label switched path. A sequence of hops in which a packet travels from one router to another router by means of label switching mechanisms. A label switched path can be established dynamically, based on normal routing mechanisms, or through configuration.

LSR --label switch router. A device that forwards Multiprotocol Label Switching (MPLS) packets based on the value of a fixed-length label encapsulated in each packet.

Management Information Base --See MIB.

MIB --Management Information Base. A database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP). The value of a MIB object can be changed or retrieved by means of SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS --Multiprotocol Label Switching. A switching method that forwards IP traffic through use of a label. This label instructs the routers and the switches in the network where to forward the packets. The forwarding of MPLS packets is based on preestablished IP routing information.

MPLS interface --An interface on which Multiprotocol Label Switching (MPLS) traffic is enabled.

Multiprotocol Label Switching --See MPLS.

notification request --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal, indicating that a significant event occurred. SNMP notification requests are more reliable than traps, because a notification request from an SNMP agent requires that the SNMP manager acknowledge receipt of the notification request. The manager replies with an SNMP response protocol data unit (PDU). If the manager does not receive a notification message from an SNMP agent, it does not send a response. If the sender (SNMP agent) never receives a response, the notification request can be sent again.

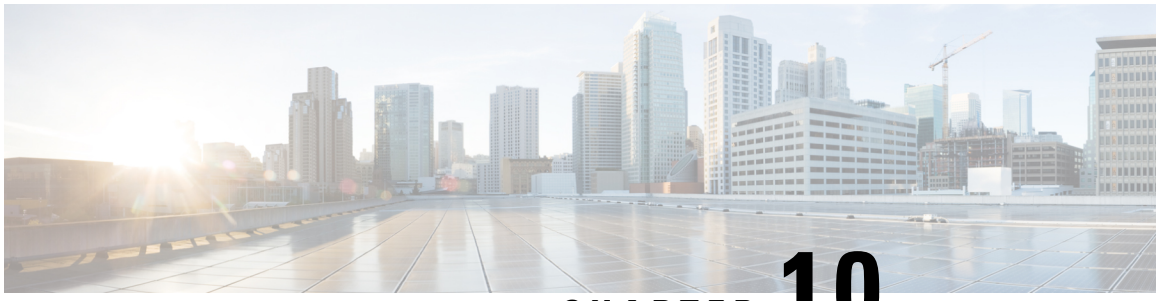
outSegmen t--A label on an outgoing packet.

Simple Network Management Protocol --See SNMP.

SNMP --Simple Network Management Protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices, and for managing configurations, statistics collection, performance, and security.

trap --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps are less reliable than notification requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received.

Note Refer to the Cisco Dictionary of Internetworking Terms and Acronyms for terms not included in this glossary.



CHAPTER 10

MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV

The MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV feature helps service providers monitor label switched paths (LSPs) and quickly isolate Multiprotocol Label Switching (MPLS) forwarding problems.

The feature provides the following capabilities:

- MPLS LSP ping to test LSP connectivity for IPv4 Label Distribution Protocol (LDP) prefixes, Resource Reservation Protocol (RSVP) traffic engineering (TE), and Any Transport over MPLS (AToM) forwarding equivalence classes (FECs).
- MPLS LSP traceroute to trace the LSPs for IPv4 LDP prefixes and RSVP TE prefixes.

Prerequisites for MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV

Before you use the MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV feature, you should:

- Determine the baseline behavior of your MPLS network. For example:
  - Expected MPLS experimental (EXP) treatment.
  - Expected maximum size packet or maximum transmission unit (MTU) of the LSP.
  - The topology, expected label switched path, and number of links in the LSP. Trace the paths of the label switched packets including the paths for load balancing.
- Understand how to use MPLS and MPLS applications. You need to:

- Know how LDP is configured.
- Understand AToM concepts.

- Understand label switching, forwarding, and load balancing.

Before using the **ping mpls** or **trace mpls** command, you must ensure that the router is configured to encode and decode MPLS echo packets in a format that all receiving routers in the network can understand.

Restrictions for MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV

- You cannot use MPLS LSP traceroute to trace the path taken by AToM packets. MPLS LSP traceroute is not supported for AToM. (MPLS LSP ping is supported for AToM.) However, you can use MPLS LSP traceroute to troubleshoot the Interior Gateway Protocol (IGP) LSP that is used by AToM.
- You cannot use MPLS LSP ping to validate or trace MPLS Virtual Private Networks (VPNs).
- You cannot use MPLS LSP traceroute to troubleshoot LSPs that employ time-to-live (TTL) hiding.
- MPLS supports per-destination and per-packet (round robin) load balancing. If per-packet load balancing is in effect, you should not use MPLS LSP traceroute because LSP traceroute at a transit router consistency checks the information supplied in the previous echo response from the directly connected upstream router. When round robin is employed, the path that an echo request packet takes cannot be controlled in a way that allows a packet to be directed to TTL expire at a given router. Without that ability, the consistency checking may fail during an LSP traceroute. A consistency check failure return code may be returned.
- A platform must support LSP ping and traceroute in order to respond to an MPLS echo request packet.
- Unless the MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV feature is enabled along the entire path, you cannot get a reply if the request fails along the path at any node.
- There are certain limitations when a mixture of draft versions are implemented within a network. The version of the draft must be compatible with Cisco's implementation. Due to the way the LSP Ping draft was written, earlier versions may not be compatible with later versions because of changes to type, length, values (TLVs) formats without sufficient versioning information. Cisco attempts to compensate for this in its implementations by allowing the sending and responding routers to be configured to encode and decode echo packets assuming a certain version.
- If you want to use MPLS LSP traceroute, the network should not use TTL hiding.

Information About MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV

MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV Functionality

Internet Control Message Protocol (ICMP) ping and traceroute are often used to help diagnose the root cause when a forwarding failure occurs. However, they are not well suited for identifying LSP failures because an ICMP packet can be forwarded via IP to the destination when an LSP breakage occurs.

The MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV feature is well suited for identifying LSP breakages for the following reasons:

- An MPLS echo request packet cannot be forwarded via IP because IP TTL is set to 1 and the IP destination address field is set to a 127/8 address.
- The FEC being checked is not stored in the IP destination address field (as is the case of ICMP).

MPLS echo request and reply packets test LSPs. There are two methods by which a downstream router can receive packets:

- The Cisco implementation of MPLS echo request and echo reply that was previously based on the Internet Engineering Task Force (IETF) Internet Draft Detecting MPLS Data Plane Failures (draft-ietf-mpls-lsp-ping-03.txt).
- Features described in this document that are based on the IETF RFC 4379 Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures :
  - Echo request output interface control
  - Echo request traffic pacing
  - Echo request end-of-stack explicit-null label shimming
  - Echo request request-dsmap capability
  - Request-fec checking
  - Depth limit reporting

MPLS LSP Ping Operation

MPLS LSP ping uses MPLS echo request and reply packets to validate an LSP. You can use MPLS LSP ping to validate IPv4 LDP, AToM, and IPv4 RSVP FECs by using appropriate keywords and arguments with the **ping mpls**command.

The MPLS echo request packet is sent to a target router through the use of the appropriate label stack associated with the LSP to be validated. Use of the label stack causes the packet to be forwarded over the LSP itself.

The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.*x* .*y* .*z* /8 address. The 127.*x* .*y* .*z* /8 address prevents the IP packet from being IP switched to its destination if the LSP is broken.

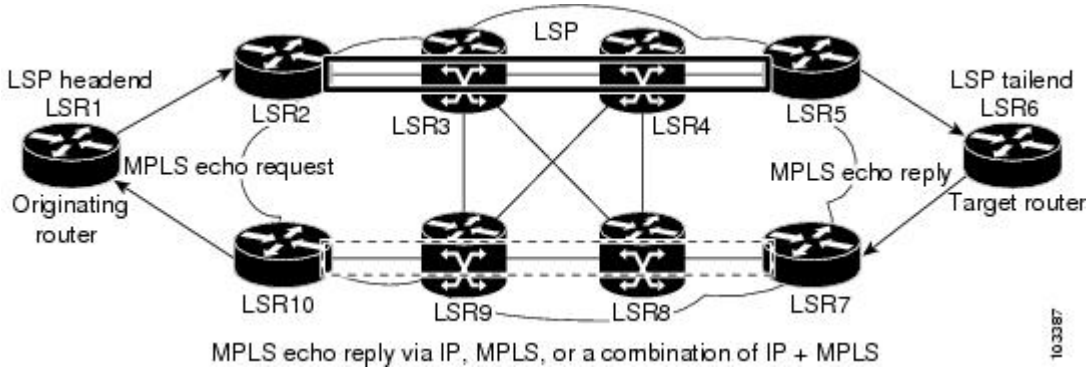
An MPLS echo reply is sent in response to an MPLS echo request. The reply is sent as an IP packet and it is forwarded using IP, MPLS, or a combination of both types of switching. The source address of the MPLS

echo reply packet is an address obtained from the router generating the echo reply. The destination address is the source address of the router that originated the MPLS echo request packet.

The MPLS echo reply destination port is set to the echo request source port.

The figure below shows MPLS LSP ping echo request and echo reply paths.

Figure 14: MPLS LSP Ping Echo Request and Echo Reply Paths



If you initiate an MPLS LSP ping request at LSR1 to a FEC at LSR6, you get the results shown in the table below.

Table 14: MPLS LSP Ping Example

Step	Router	Action
1.	LSR1	Initiates an MPLS LSP ping request for an FEC at the target router LSR6 and sends an MPLS echo request to LSR2.
2.	LSR2	Receives the MPLS echo request packet and forwards it through transit routers LSR3 and LSR4 to the penultimate router LSR5.
3.	LSR5	Receives the MPLS echo request, pops the MPLS label, and forwards the packet to LSR6 as an IP packet.
4.	LSR6	Receives the IP packet, processes the MPLS echo request, and sends an MPLS echo reply to LSR1 through an alternate route.
5.	LSR7 to LSR10	Receives the MPLS echo reply and forwards it back toward LSR1, the originating router.
6.	LSR1	Receives the MPLS echo reply in response to its MPLS echo request.

MPLS LSP Traceroute Operation

MPLS LSP traceroute uses MPLS echo request and reply packets to validate an LSP. You can use MPLS LSP traceroute to validate IPv4 LDP and IPv4 RSVP FECs by using appropriate keywords and arguments with the **trace mpls** command.

The MPLS LSP Traceroute feature uses TTL settings to force expiration of the TTL along an LSP. MPLS LSP Traceroute incrementally increases the TTL value in its MPLS echo requests (TTL = 1, 2, 3, 4) to discover the downstream mapping of each successive hop. The success of the LSP traceroute depends on the transit

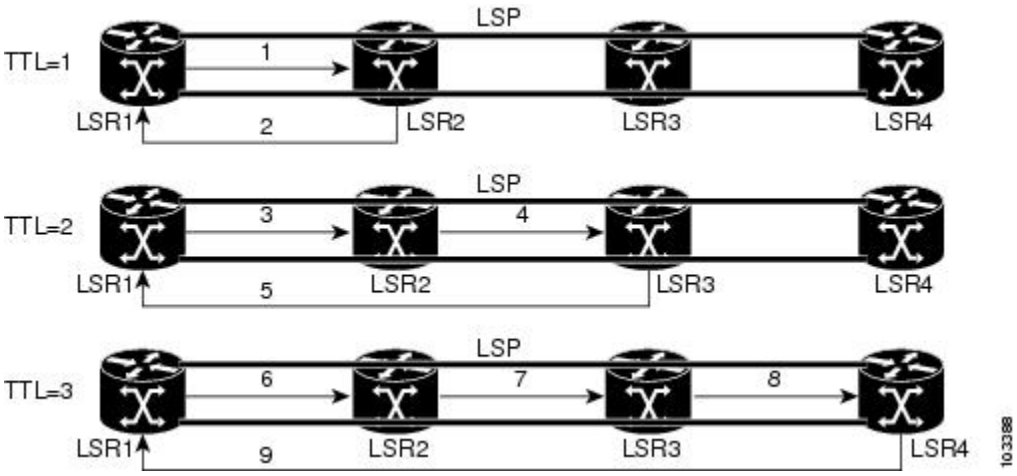
router processing the MPLS echo request when it receives a labeled packet with a TTL = 1. On Cisco routers, when the TTL expires, the packet is sent to the Route Processor (RP) for processing. The transit router returns an MPLS echo reply containing information about the transit hop in response to the TTL-expired MPLS packet.

The MPLS echo reply destination port is set to the echo request source port.

Note When a router traces an IPV4 FEC that goes over a traffic engineering tunnel, intermediate routers may return U (unreachable) if LDP is not running in those intermediate routers.

The figure below shows an MPLS LSP traceroute example with an LSP from LSR1 to LSR4.

Figure 15: MPLS LSP Traceroute Example



If you enter an LSP traceroute to an FEC at LSR4 from LSR1, you get the results shown in the table below.

Table 15: MPLS LSP Traceroute Example

Step	Router	MPLS Packet Type and Description	Router Action (Receive or Send)
1.	LSR1	MPLS echo request--With a target FEC pointing to LSR4 and to a downstream mapping	• Sets the TTL of the label stack to 1 • Sends the request to LSR2
2.	LSR2	MPLS echo reply	• Receives the packet with a TTL = 1 • Processes the User Datagram Protocol (UDP) packet as an MPLS echo request • Finds a downstream mapping and replies to LSR1 with its own downstream mapping, based on the incoming label
3.	LSR1	MPLS echo request--With the same target FEC and the downstream mapping received in the echo reply from LSR2	• Sets the TTL of the label stack to 2 • Sends the request to LSR2

Step	Router	MPLS Packet Type and Description	Router Action (Receive or Send)
4.	LSR2	MPLS echo request	• Receives the packet with a TTL = 2 • Decrements the TTL • Forwards the echo request to LSR3
5.	LSR3	MPLS reply packet	• Receives the packet with a TTL = 1 • Processes the UDP packet as an MPLS echo request • Finds a downstream mapping and replies to LSR1 with its own downstream mapping based on the incoming label
6.	LSR1	MPLS echo request--With the same target FEC and the downstream mapping received in the echo reply from LSR3	• Sets the TTL of the packet to 3 • Sends the request to LSR2
7.	LSR2	MPLS echo request	• Receives the packet with a TTL = 3 • Decrements the TTL • Forwards the echo request to LSR3
8.	LSR3	MPLS echo request	• Receives the packet with a TTL = 2 • Decrements the TTL • Forwards the echo request to LSR4
9.	LSR4	MPLS echo reply	• Receives the packet with a TTL = 1 • Processes the UDP packet as an MPLS echo request • Finds a downstream mapping and also finds that the router is the egress router for the target FEC • Replies to LSR1

MPLS Network Management with MPLS LSP Ping and MPLS LSP Traceroute

To manage an MPLS network, you must have the ability to monitor LSPs and quickly isolate MPLS forwarding problems. You need ways to characterize the liveliness of an LSP and reliably detect when an LSP fails to deliver user traffic.

You can use MPLS LSP ping to verify the LSP that is used to transport packets destined for IPv4 LDP prefixes, and AToM PW FECs. You can use MPLS LSP traceroute to trace LSPs that are used to carry packets destined for IPv4 LDP prefixes.

An MPLS echo request is sent through an LSP to validate it. A TTL expiration or LSP breakage causes the transit router to process the echo request before it gets to the intended destination. The router returns an MPLS echo reply that contains an explanatory reply code to the originator of the echo request.

The successful echo request is processed at the egress of the LSP. The echo reply is sent via an IP path, an MPLS path, or a combination of both back to the originator of the echo request.

Any Transport over MPLS Virtual Circuit Connection

AToM Virtual Circuit Connection Verification (VCCV) allows you to send control packets inband of an AToM PW from the originating provider edge (PE) router. The transmission is intercepted at the destination PE router, instead of being forwarded to the customer edge (CE) router. This capability allows you to use MPLS LSP ping to test the PW section of AToM virtual circuits (VCs).

LSP ping allows verification of AToM VC setup by FEC 128 or FEC 129. FEC 128-based AToM VCs can be set up by using LDP for signaling or by using a static pseudowire configuration without using any signaling component on the two endpoints. Cisco software does not distinguish between FEC 128 and FEC 129 static pseudowires while issuing MPLS ping; the same commands are used.

AToM VCCV consists of the following:

- A signaled component in which the AToM VCCV capabilities are advertised during VC label signaling
- A switching component that causes the AToM VC payload to be treated as a control packet

AToM VCCV Signaling

One of the steps involved in AToM VC setup is the signaling or communication of VC labels and AToM VCCV capabilities between AToM VC endpoints. To communicate the AToM VCCV disposition capabilities of each endpoint, the router uses an optional parameter, defined in the IETF Internet Draft *Pseudo Wire (PW) Virtual Circuit Connection Verification (VCCV)* (draft-ieft-pwe3-vccv-01).

The AToM VCCV disposition capabilities are categorized as follows:

- Applications--MPLS LSP ping and ICMP ping are applications that AToM VCCV supports to send packets inband of an AToM PW for control purposes.
- Switching modes--Type 1 and Type 2 are switching modes that AToM VCCV uses for differentiating between control and data traffic.

The table below describes AToM VCCV Type 1 and Type 2 switching modes.

Table 16: Type 1 and Type 2 AToM VCCV Switching Modes

Switching Mode	Description
Type 1	Uses a Protocol ID (PID) field in the AToM control word to identify an AToM VCCV packet
Type 2	Uses an MPLS Router Alert Label above the VC label to identify an AToM VCCV packet

Selection of AToM VCCV Switching Types

Cisco routers always use Type 1 switching, if available, when they send MPLS LSP ping packets over an AToM VC control channel. Type 2 switching accommodates those VC types and implementations that do not support or interpret the AToM control word.

The table below shows the AToM VCCV switching mode advertised and the switching mode selected by the AToM VC.

Table 17: AToM VCCV Switching Mode Advertised and Selected by AToM VC

Type Advertised	Type Selected
AToM VCCV not supported	--
Type 1 AToM VCCV switching	Type 1 AToM VCCV switching
Type 2 AToM VCCV switching	Type 2 AToM VCCV switching
Type 1 and Type 2 AToM VCCV switching	Type 1 AToM VCCV switching

An AToM VC advertises its AToM VCCV disposition capabilities in both directions: that is, from the originating router (PE1) to the destination router (PE2), and from PE2 to PE1.

In some instances, AToM VCs might use different switching types if the two endpoints have different AToM VCCV capabilities. If PE1 supports Type 1 and Type 2 AToM VCCV switching and PE2 supports only Type 2 AToM VCCV switching, there are two consequences:

- LSP ping packets sent from PE1 to PE2 are encapsulated with Type 2 switching.
- LSP ping packets sent from PE2 to PE1 use Type 1 switching.

You can determine the AToM VCCV capabilities advertised to and received from the peer by entering the **show mpls l2transport binding** command at the PE router.

Information Provided by the Router Processing LSP Ping or LSP Traceroute

The table below describes the characters that the router processing an LSP ping or LSP traceroute packet returns to the sender about the failure or success of the request.

You can also display the return code for an MPLS LSP Ping operation if you enter the **ping mpls verbose** command.

Table 18: Echo Reply Return Codes

Output Code	Echo Return Code	Meaning
x	0	No return code.
M	1	Malformed echo request.
m	2	Unsupported TLVs.
!	3	Success.
F	4	No FEC mapping.
D	5	DS Map mismatch.
I	6	Unknown Upstream Interface index.
U	7	Reserved.

Output Code	Echo Return Code	Meaning
L	8	Labeled output interface.
B	9	Unlabeled output interface.
f	10	FEC mismatch.
N	11	No label entry.
P	12	No receive interface label protocol.
p	13	Premature termination of the LSP.
X	unknown	Undefined return code.

Note Echo return codes 6 and 7 are accepted only for Version 3 (draft-ieft-mpls-ping-03).

IP Does Not Forward MPLS Echo Request Packets

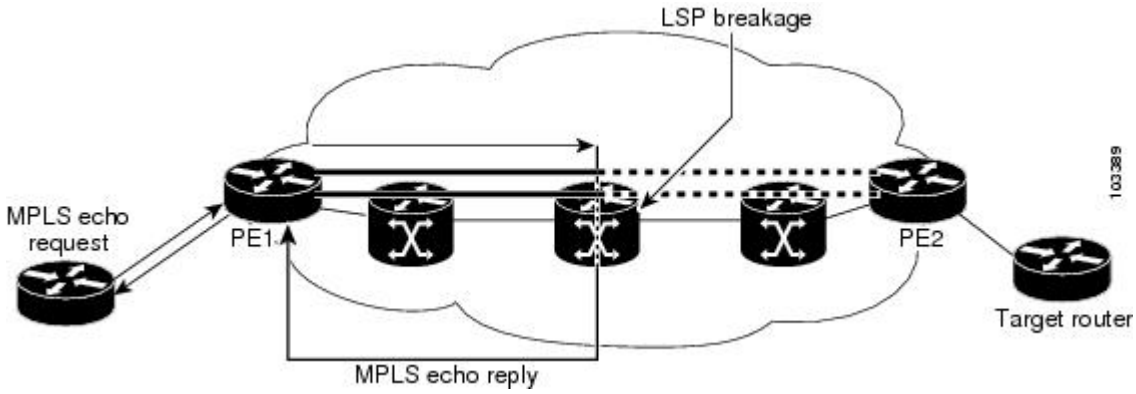
MPLS echo request packets sent during an LSP ping are never forwarded by IP. The IP header destination address field in an MPLS echo request packet is a 127.*x.y.z* /8 address. Routers should not forward packets using a 127.*x.y.z* /8 address. The 127.*x.y.z* /8 address corresponds to an address for the local host.

Use of a 127.*x .y .z* address as the destination address of the UDP packet is significant because the MPLS echo request packet fails to make it to the target router if a transit router does not label switch the LSP. The use of the 127.*x .y .z* address allows for the detection of LSP breakages. The following occurs at the transit router:

- If an LSP breakage occurs at a transit router, the MPLS echo packet is not forwarded; it is consumed by the router.
- If the LSP is intact, the MPLS echo packet reaches the target router and is processed by the terminal point of the LSP.

The figure below shows the path of the MPLS echo request and reply when a transit router fails to label switch a packet in an LSP.

Figure 16: Path when Transit Router Fails to Label Switch a Packet



Note An AToM payload does not contain usable forwarding information at a transit router because the payload may not be an IP packet. An MPLS VPN packet, although an IP packet, does not contain usable forwarding information at a transit router because the destination IP address is significant only to the virtual routing and forwarding (VRF) instances at the endpoints of the MPLS network.

Compatibility Between the MPLS LSP and Ping or Traceroute Implementations

LSP ping drafts after Version 3 (draft-ietf-mpls-ping-03) have undergone numerous TLV format changes, but the versions of the draft do not always interoperate.

To allow later Cisco implementations to interoperate with draft Version 3 Cisco and non-Cisco implementations, use a global configuration mode to decode echo packets in formats understood by draft Version 3 implementations.

Unless configured otherwise, a Cisco implementation encodes and decodes echo requests assuming the version on which the IETF implementations is based.

To prevent failures reported by the replying router due to TLV version issues, you should configure all routers in the core. Encode and decode MPLS echo packets in the same draft version. For example, if the network is running RFC 4379 (Cisco Version 4) implementations but one router is capable of only Version 3 (Cisco Revision 3), configure all routers in the network to operate in Revision 3 mode.

The Cisco implementation of MPLS echo request and echo reply is based on the IETF RFC 4379. IEFT drafts subsequent to this RFC (drafts 3, 4, 5, 6, and 7) introduced TLV format differences. These differences could not be identified because the echo packet had no way to differentiate between one TLV format and another TLV format. To allow interoperability, a **revision** keyword was added for the **ping mpls** and **trace mpls** commands. The **revision** keyword enables Cisco IOS XE releases to support the existing draft changes and any changes from future versions of the IETF LSP Ping draft.

Note We recommend that you use the **mpls oam** global configuration command instead of the revision option.

Note No images are available on cisco.com to support Revision 2. It is recommended that you use only images supporting Version 3 and later when configuring TLV encode and decode modes. MPLS Multipath LSP traceroute requires Cisco Revision 4 or later.

CiscoVendorExtensions

In Cisco's Version 3 (draft-ietf-mpls-ping-03.txt) implementations, Cisco defined a vendor extension TLV in the ignore-if-not-understood TLV space. It is used for the following purposes:

- Provide an ability to track TLV versions.
- Provide an experimental Reply TOS capability.

The first capability was defined before the existence of the global configuration command for setting the echo packet encode and decode behavior. TLV version information in an echo packet overrides the configured decoding behavior. Using this TLV for TLV versions is no longer required since the introduction of the global configuration capability.

The second capability controls the reply DSCP. Draft Version 8 defines a Reply TOS TLV, so the use of the reply DSCP is no longer required.

You enable compatibility between the MPLS LSP and ping or traceroute implementation by customizing the default behavior of echo packets.

DSCP Option to Request a Specific Class of Service in an Echo Reply

Cisco software includes a reply differentiated services code point (DSCP) option that lets you request a specific class of service (CoS) in an echo reply.

The reply DSCP option is supported in the experimental mode for IETF draft-ietf-mpls-lsp-ping-03.txt. Cisco implemented a vendor-specific extension for the reply DSCP option rather than using a Reply TOS TLV. A Reply TOS TLV serves the same purpose as the **reply dscp** command in RFC 4379. This draft provides a standardized method of controlling the reply DSCP.

Note Before draft Version 8, Cisco implemented the Reply DSCP option as an experimental capability using a Cisco vendor extension TLV. If a router is configured to encode MPLS echo packets for draft Version 3 implementations, a Cisco vendor extension TLV is used instead of the Reply TOS TLV that was defined in draft Version 8.

Reply Modes for an MPLS LSP Ping and LSP Traceroute Echo Request Response

The reply mode controls how a responding router replies to an MPLS echo request sent by a **ping mpls** or **trace mpls** command. There are two reply modes for an echo request packet:

- ipv4--Reply with an IPv4 UDP packet (default)
- router-alert--Reply with an IPv4 UDP packet with router alert

Note It is useful to use ipv4 and router-alert reply modes in conjunction with each other to prevent false negatives. If you do not receive a reply via the ipv4 mode, it is useful to send a test with the router-alert reply mode. If both fail, something is wrong in the return path. The problem may be only that the Reply TOS is not set correctly.

IPv4 Reply Mode

IPv4 packet is the most common reply mode used with a **ping mpls** or **trace mpls** command when you want to periodically poll the integrity of an LSP. With this option, you do not have explicit control over whether the packet traverses IP or MPLS hops to reach the originator of the MPLS echo request. If the originating (headend) router fails to receive a reply to an MPLS echo request when you use the **reply mode ipv4** keywords, use the **reply mode router-alert** keywords.

Router-Alert Reply Mode

The router-alert reply mode adds the router alert option to the IP header. When an IP packet that contains an IP router alert option in its IP header or an MPLS packet with a router alert label as its outermost label arrives at a router, the router punts (redirects) the packet to the Route Processor (RP) level for handling. This forces the Cisco router to handle the packet at each intermediate hop as it moves back to the destination. Hardware and line-card forwarding inconsistencies are bypassed. Router-alert reply mode is more expensive than IPv4 mode because the reply goes hop-by-hop. It also is slower, so the sender receives a reply in a relatively longer period of time.

The table below describes how IP and MPLS packets with an IP router alert option are handled by the router switching path processes.

Table 19: Path Process Handling of IP and MPLS Router Alert Packets

Incoming Packet	Normal Switching Action	Process Switching Action	Outgoing Packet
IP packet--Router alert option in IP header	Router alert option in IP header causes the packet to be punted to the process switching path.	Forwards the packet as is	IP packet--Router alert option in IP header
		Forwards the packet as is	MPLS packet
MPLS packet--Outermost label contains a router alert	If the router alert label is the outermost label, it causes the packet to be punted to the process switching path.	Removes the outermost router alert label and forwards the packet as an IP packet	IP packet--Router alert option in IP header
		Preserves the outermost router alert label and forwards the MPLS packet	MPLS packet-- Outermost label contains a router alert.

LSP Breaks

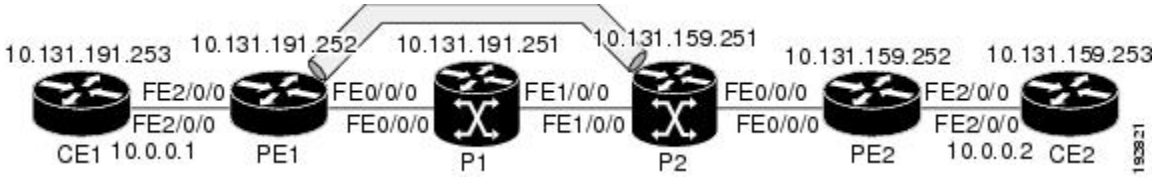
If there is a problem forwarding MPLS packets in your network, you can determine where there are LSP breaks. This section describes MTU discovery in an LSP.

Untagged output interfaces at a penultimate hop do not impact the forwarding of IP packets through an LSP because the forwarding decision is made at the penultimate hop through use of the incoming label. However, untagged output interfaces cause AToM and MPLS VPN traffic to be dropped at the penultimate hop.

During an MPLS LSP ping, MPLS echo request packets are sent with the IP packet attribute set to "do not fragment." That is, the Don't Fragment (DF) bit is set in the IP header of the packet. This allows you to use the MPLS echo request to test for the MTU that can be supported for the packet through the LSP without fragmentation.

The figure below shows a sample network with a single LSP from PE1 to PE2 formed with labels advertised by the LDP.

Figure 17: Sample Network with LSP--Labels Advertised by LDP



You can determine the maximum receive unit (MRU) at each hop by using the MPLS Traceroute feature to trace the LSP. The MRU is the maximum size of a labeled packet that can be forwarded through an LSP.

How to Configure MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV

Enabling Compatibility Between the MPLS LSP and Ping or Traceroute Implementation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mplsoam**
4. **echo revision** {**3** | **4**}
5. **echo vendor-extension**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mplsoam** **Example:** `Router(config)# mpls oam`	Enters MPLS OAM configuration mode for customizing the default behavior of echo packets.
Step 4	**echo revision** {**3** | **4**} **Example:** `Router(config-mpls)# echo revision 4`	Specifies the revision number of the echo packet's default values. • 3--draft-ietf-mpls-ping-03 (Revision 2). • 4--RFC 4379 compliant (default).
Step 5	**echo vendor-extension** **Example:** `Router(config-mpls)# echo vendor-extension`	Sends the Cisco-specific extension of TLVs with echo packets.
Step 6	**exit** **Example:** `Router(config-mpls)# exit`	Returns to global configuration mode.

Validating an LDP IPv4 FEC by Using MPLS LSP Ping and MPLS LSP Traceroute

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **ping mpls ipv4** *destination-address /destination-mask-length* [**repeat** *count*] [**exp** *exp-bits*] [**verbose**]
   - **trace mpls ipv4** *destination-address /destination-mask-length*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose

Step 2	Do one of the following:
	• **ping mpls ipv4** *destination-address /destination-mask-length* [**repeat** *count*] [**exp** *exp-bits*] [**verbose**]
	• **trace mpls ipv4** *destination-address /destination-mask-length*
	Example:
	`Router# ping mpls ipv4 10.131.191.252/32 exp 5 repeat 5 verbose`
	Example:
	`Router# trace mpls ipv4 10.131.191.252/32`
		Selects an LDP IPv4 prefix FEC for validation.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Validating a Layer 2 FEC by Using MPLS LSP Ping and MPLS LSP Traceroute

SUMMARY STEPS

1. **enable**
2. **ping mpls pseudowire** *ipv4-address* **vc-id** *vc-id*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping mpls pseudowire** *ipv4-address* **vc-id** *vc-id* **Example:** `Router# ping mpls pseudowire 10.131.191.252 vc-id 333`	Selects a Layer 2 FEC for validation.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Using DSCP to Request a Specific Class of Service in an Echo Reply

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **ping mpls** {**ipv4** *destination-address/destination-mask-length* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} [**reply dscp** *dscp-value*]
   - **trace mpls ipv4** *destination-address/destination-mask-length* [**reply dscp** *dscp-value*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	Do one of the following:
	• **ping mpls** {**ipv4** *destination-address/destination-mask-length*
		pseudowire *ipv4-address* **vc-id** *vc-id*} [**reply dscp** *dscp-value*]
		• **trace mpls ipv4** *destination-address/destination-mask-length* [**reply dscp** *dscp-value*]
		Example:
		`Router# ping mpls ipv4 10.131.191.252/32 reply dscp 50`
		Example:
		`Router# trace mpls ipv4 10.131.191.252/32 reply dscp 50`
			Controls the DSCP value of an echo reply.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Controlling How a Responding Router Replies to an MPLS Echo Request

SUMMARY STEPS

1. **enable**

2. Do one of the following:
   - **ping mpls** {**ipv4***destination-address*/*destination-mask-length* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} **reply mode** {**ipv4** | **router-alert**}
   - **trace mpls ipv4** *destination-address*/*destination-mask* **reply mode** {**ipv4** | **router-alert**}
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	Do one of the following:
	• **ping mpls** {**ipv4***destination-address*/*destination-mask-length*
		pseudowire *ipv4-address* **vc-id** *vc-id*} **reply mode** {**ipv4**	**router-alert**}
			• **trace mpls ipv4** *destination-address*/*destination-mask* **reply mode** {**ipv4**
				router-alert}
				Example:
				`Router# ping mpls ipv4 10.131.191.252/32 reply mode ipv4`
				Example:
				`Router# trace mpls ipv4 10.131.191.252/32 reply mode router-alert`
					Checks MPLS LSP connectivity.
					or
					Discovers MPLS LSP routes that packets actually take when traveling to their destinations.
					Note To specify the reply mode, you must enter the **reply mode** keyword with the **ipv4** or the **router-alert** keyword.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Using MPLS LSP Ping to Discover Possible Loops

With the MPLS LSP Ping feature, loops can occur if you use the UDP destination address range, repeat option, or sweep option.

To use MPLS LSP ping to discover possible loops, perform the following steps.

SUMMARY STEPS

1. **enable**

2. **ping mpls** {**ipv4** *destination-address/destination-mask* [**destination** *address-start address-end increment* | [**pseudowire** *ipv4-address* **vc-id** *vc-id address-end increment*]} [**repeat** *count*] [**sweep** *minimum maximum size-increment*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping mpls** {**ipv4** *destination-address/destination-mask* [**destination** *address-start address-end increment*	[**pseudowire** *ipv4-address* **vc-id** *vc-id address-end increment*]} [**repeat** *count*] [**sweep** *minimum maximum size-increment*] **Example:** `Router# ping mpls ipv4 10.131.159.251/32 destination 127.0.0.1 127.0.0.2 1 repeat 2 sweep 1450 1475 25`	Checks MPLS LSP connectivity.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Using MPLS LSP Traceroute to Discover Possible Loops

With the MPLS LSP Traceroute feature, loops can occur if you use the UDP destination address range option and the time-to-live option.

By default, the maximum TTL is set to 30. Therefore, the traceroute output may contain 30 lines if the target of the traceroute is not reached, which can happen when an LSP problem exists. If an LSP problem occurs, there may be duplicate entries. The router address of the last point that the trace reaches is repeated until the output is 30 lines. You can ignore the duplicate entries.

SUMMARY STEPS

1. **enable**
2. **trace mpls ipv4** *destination-address* /*destination-mask* [**destination** *address-start address-end address increment*] [**ttl** *maximum-time-to-live*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls ipv4** *destination-address* /*destination-mask* [**destination** *address-start address-end address increment*] [**ttl** *maximum-time-to-live*] **Example:** `Router# trace mpls ipv4 10.131.159.251/32 destination 127.0.0.1 127.0.0.3 1 ttl 5`	Discovers MPLS LSP routes that packets take when traveling to their destinations. The example shows how a loop can occur.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

What to do next

Tracking Packets Tagged as Implicit Null

SUMMARY STEPS

1. **enable**
2. **trace mpls ipv4** *destination-address*/*destination-mask*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls ipv4** *destination-address*/*destination-mask* **Example:** `Router# trace mpls ipv4 10.131.159.252/32`	Discovers MPLS LSP routes that packets actually take when traveling to their destinations.
Step 3	**exit** **Example:**	Returns to user EXEC mode.

	Command or Action	Purpose
	`Router# exit`	

Tracking Untagged Packets

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** *destination-address/destination-mask*
3. **show mpls ldp discovery**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** *destination-address/destination-mask* **Example:** `Router# show mpls forwarding-table 10.131.159.252/32`	Displays the content of the MPLS Label Forwarding Information Base (LFIB) and displays whether the LDP is properly configured.
Step 3	**show mpls ldp discovery** **Example:** `Router# show mpls ldp discovery`	Displays the status of the LDP discovery process and displays whether the LDP is properly configured.
Step 4	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Determining Why a Packet Could Not Be Sent

The Q return code means that the packet could not be sent. The problem can be caused by insufficient processing memory, but it probably results because an LSP could not be found that matches the FEC information that was entered on the command line.

You need to determine the reason why the packet was not forwarded so that you can fix the problem in the path of the LSP. To do so, look at the Routing Information Base (RIB), the Forwarding Information Base

(FIB), the Label Information Base (LIB), and the MPLS LFIB. If there is no entry for the FEC in any of these routing or forwarding bases, there is a Q return code.

To determine why a packet could not be transmitted, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip route** [*ip-address* [**mask**]]
3. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label*[-*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route** [*ip-address* [**mask**]] **Example:** `Router# show ip route 10.0.0.1`	Displays the current state of the routing table. When the MPLS echo reply returns a Q, troubleshooting occurs on the routing information database.
Step 3	**show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label*[-*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] **Example:** `Router# show mpls forwarding-table 10.0.0.1/32`	Displays the content of the MPLS LFIB. When the MPLS echo reply returns a Q, troubleshooting occurs on a label information database and an MPLS forwarding information database.
Step 4	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Detecting LSP Breaks when Load Balancing Is Enabled for IPv4 LDP LSPs

An ICMP ping or trace follows one path from the originating router to the target router. Round robin load balancing of IP packets from a source router discovers the various output paths to the target IP address.

For MPLS ping and traceroute, Cisco routers use the source and destination addresses in the IP header for load balancing when multiple paths exist through the network to a target router. The Cisco implementation of MPLS may check the destination address of an IP payload to accomplish load balancing (the type of checking depends on the platform).

To detect LSP breaks when load balancing is enabled for IPv4 LDP LSPs, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **ping mpls ipv4** *destination-address/destination-mask-length* [**destination** *address-start address-end increment*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping mpls ipv4** *destination-address/destination-mask-length* [**destination** *address-start address-end increment*] **Example:** `Router# ping mpls ipv4 10.131.159.251/32 destination 127.0.0.1/8`	Checks for load balancing paths. Enter the 127.*z.y.x* /8 destination address.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Specifying the Interface Through Which Echo Packets Leave a Router

You can control the interface through which packets leave a router. Path output information is used as input to LSP ping and traceroute.

The echo request output interface control feature allows you to force echo packets through the paths that perform detailed debugging or characterizing of the LSP. This feature is useful if a PE router connects to an MPLS cloud and there are broken links. You can direct traffic through a certain link. The feature also is helpful for troubleshooting network problems.

To specify the output interface for echo requests, perform the following steps.

SUMMARY STEPS

1. **enable**
2. Enter one of the following commands:
   - **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} [**output interface** *tx-interface*]
   - **trace mpls ipv4** *destination-address/destination-mask*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	Enter one of the following commands:
	• **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} [**output interface** *tx-interface*]
	• **trace mpls ipv4** *destination-address/destination-mask*
	Example:
	`Router# ping mpls ipv4 10.131.159.251/32 output interface fastethernet0/0/0`
	Example:
	`Router# trace mpls ipv4 10.131.159.251/32 output interface fastethernet0/0/0`
		Checks MPLS LSP connectivity.
		or
		Discovers MPLS LSP routes that packets actually take when traveling to their destinations.
		Note For this task, you must specify the **output interface** keyword.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Pacing the Transmission of Packets

Echo request traffic pacing allows you to pace the transmission of packets so that the receiving router does not drop packets. To perform echo request traffic pacing, perform the following steps.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} [**interval** *ms*]]
   - **trace mpls ipv4** *destination-address/destination-mask*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	

Step 2	Do one of the following:
	• **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} [**interval** *ms*]]
	• **trace mpls ipv4** *destination-address/destination-mask*
	Example:
	`Router# ping mpls ipv4 10.131.159.251/32 interval 2`
	Example:
	`Router# trace mpls ipv4 10.131.159.251/32`
		Checks MPLS LSP connectivity.
		or
		Discovers MPLS LSP routes that packets take when traveling to their destinations.
		Note In this task, if you use the **ping mpls** command you must specify the **interval** keyword.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Interrogating the Transit Router for Its Downstream Information by Using Echo Request request-dsmap

The echo request request-dsmap capability troubleshooting feature, used in conjunction with the TTL flag, allows you to selectively interrogate a transit router. If there is a failure, you do not have to enter an **lsp traceroute** command for each previous failure; you can focus just on the failed hop.

A request-dsmap flag in the downstream mapping flags field, and procedures that specify how to trace noncompliant routers allow you to arbitrarily time-to-live (TTL) expire MPLS echo request packets with a wildcard downstream map (DSMAP).

Echo request DSMAPs received without labels indicate that the sender did not have any DSMAPs to validate. If the downstream router ID field of the DSMAP TLV in an echo request is set to the ALLROUTERs address (224.0.0.2) and there are no labels, the source router can arbitrarily query a transit router for its DSMAP information.

The **ping mpls** command allows an MPLS echo request to be TTL-expired at a transit router with a wildcard DSMAP for the explicit purpose of troubleshooting and querying the downstream router for its DSMAPs. The default is that the DSMAP has an IPv4 bitmap hashkey. You also can select hashkey 0 (none). The purpose of the **ping mpls** command is to allow the source router to selectively TTL expire an echo request at a transit router to interrogate the transit router for its downstream information. The ability to also select a multipath (hashkey) type allows the transmitting router to interrogate a transit router for load-balancing information as is done with multipath LSP traceroute, but without having to interrogate all subsequent nodes traversed between the source router and the router on which each echo request TTL expires. Use an echo request in conjunction with the TTL setting because if an echo request arrives at the egress of the LSP with an echo request, the responding routers never return DSMAPs.

To interrogate the transit router for its downstream information so that you can focus just on the failed hop if there is a failure, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} [**dsmap** [**hashkey** {**none** | **ipv4 bitmap** *bitmap-size*}]]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} [**dsmap** [**hashkey** {**none** | **ipv4 bitmap** *bitmap-size*}]] **Example:** `Router# ping mpls ipv4 10.161.251/32 dsmap hashkey ipv4 bitmap 16`	Checks MPLS LSP connectivity. **Note** In this task, you must specify the **dsmap** and **hashkey** keywords.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Interrogating a Router for Its DSMAP

The router can interrogate the software or hardware forwarding layer for the depth limit that needs to be returned in the DSMAP TLV. If forwarding does not provide a value, the default is 255.

To determine the depth limit, specify the **dsmap** and **ttl** keywords in the **ping mpls** command. The transit router will be interrogated for its DSMAP. The depth limit is returned with the echo reply DSMAP. A value of 0 means that the IP header is used for load balancing. Another value indicates that the IP header load balances up to the specified number of labels.

To interrogate a router for its DSMAP, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} **ttl** *time-to-live* **dsmap**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} **ttl** *time-to-live* **dsmap** **Example:** `Router# ping mpls ipv4 10.131.159.252/32 ttl 1 dsmap`	Checks MPLS LSP connectivity. **Note** You must specify the **ttl** and **dsmap** keywords.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Requesting that a Transit Router Validate the Target FEC Stack

An MPLS echo request tests a particular LSP. The LSP to be tested is identified by the FEC stack.

To request that a transit router validate the target FEC stack, set the V flag from the source router by entering the **flags fec** keyword in the **ping mpls** and **trace mpls** commands. The default is that echo request packets are sent with the V flag set to 0.

To request that a transit router validate the target FEC stack, perform the following steps.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} **flags fec**
   - **trace mpls ipv4** *destination-address/destination-mask* **flags fec**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose

Step 2	Do one of the following:
	• **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} **flags fec**
	• **trace mpls ipv4** *destination-address/destination-mask* **flags fec**
	Example:
	`Router# ping mpls ipv4 10.131.159.252/32 flags fec`
	Example:
	`Router# trace mpls ipv4 10.131.159.252/32 flags fec`
		Checks MPLS LSP connectivity.
		or
		Discovers MPLS LSP routes that packets actually take when traveling to their destinations.
		Note You must enter the **flags fec** keyword.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Enabling LSP Ping to Detect LSP Breakages Caused by Untagged Interfaces

For MPLS LSP ping and traceroute of LSPs carrying IPv4 FECs, you can force an explicit null label to be added to the MPLS label stack even though the label was unsolicited. This allows LSP ping to detect LSP breakages caused by untagged interfaces. LSP ping does not report that an LSP is operational when it is unable to send MPLS traffic.

An explicit null label is added to an MPLS label stack if MPLS echo request packets are forwarded from untagged interfaces that are directly connected to the destination of the LSP ping or if the IP TTL value for the MPLS echo request packets is set to 1.

When you enter an **lsp ping** command, you are testing the LSP's ability to carry IP traffic. Failure at untagged output interfaces at the penultimate hop are not detected. Explicit-null shimming allows you to test an LSP's ability to carry MPLS traffic.

To enable LSP ping to detect LSP breakages caused by untagged interfaces, specify the **force-explicit-null** keyword in the **ping mpls** or **trace mpls** commands as shown in the following steps.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} **force-explicit-null**
   - **trace mpls ipv4** *destination-address/destination-mask* **force-explicit-null**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	Do one of the following:
	• **ping mpls** {**ipv4** *destination-address/destination-mask* | **pseudowire** *ipv4-address* **vc-id** *vc-id*} **force-explicit-null**
	• **trace mpls ipv4** *destination-address/destination-mask* **force-explicit-null**
	Example:
	`Router# ping mpls ipv4 10.131.191.252/32 force-explicit null`
	Example:
	`Router# trace mpls ipv4 10.131.191.252/32 force-explicit-null`
		Checks MPLS LSP connectivity.
		or
		Discovers MPLS LSP routes that packets actually take when traveling to their destinations.
		Note You must enter the **force-explicit-null** keyword.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Viewing the AToM VCCV Capabilities Advertised to and Received from the Peer

To view the AToM VCCV capabilities advertised to and received from the peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show mpls l2transport binding**
3. **exit**

DETAILED STEPS

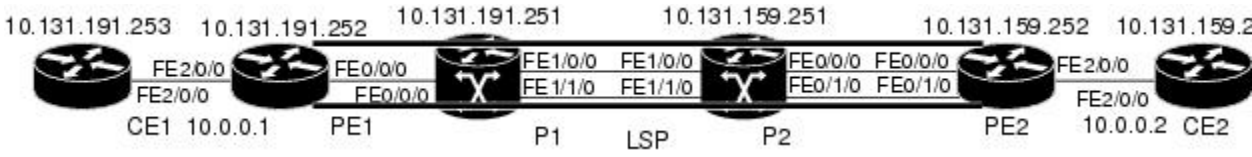
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show mpls l2transport binding** **Example:** `Router# show mpls l2transport binding`	Displays VC label binding information.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Configuration Examples for MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV

Examples for the MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV feature are based on the sample topology shown in the figure below.

Figure 18: Sample Topology for Configuration Examples



This section contains the following configuration examples:

Enabling Compatibility Between the MPLS LSP and Ping or Traceroute Implementation Example

The following example shows how to configure MPLS multipath LSP traceroute to interoperate with a vendor implementation that does not interpret RFC 4379 as Cisco does:

```
configure terminal
!
mpls oam
 echo revision 4
 no echo vendor-extension
 exit
```

The default echo revision number is 4, which corresponds to the IEFT draft 11.

Validating a Layer 2 FEC by Using MPLS LSP Ping Example

The following example validates a Layer 2 FEC:

```
Router# ping mpls pseudowire 10.10.10.15 108 vc-id 333
```

```
Sending 5, 100-byte MPLS Echos to 10.10.10.15,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
   'L' - labeled output interface, 'B' - unlabeled output interface,
   'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
   'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
   'P' - no rx intf label prot, 'p' - premature termination of LSP,
   'R' - transit router, 'I' - unknown upstream index,
   'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/32/40 ms PE-802#
```

Validating an LDP IPv4 FEC by Using MPLS LSP Ping and MPLS LSP Traceroute Example

The following example shows how to use the **ping mpls** command to test connectivity of an IPv4 LDP LSP:

```
Router# ping mpls ipv4 10.131.191.252/32 repeat 5 exp 5 verbose
Sending 5, 100-byte MPLS Echos to 10.131.191.252, timeout is 2 seconds:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!     10.131.191.230, return code 3
!     10.131.191.230, return code 3
!     10.131.191.230, return code 3
!     10.131.191.230, return code 3
!     10.131.191.230, return code 3
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/10
```

Using DSCP to Request a Specific Class of Service in an Echo Reply Example

The following example shows how to use DSCP to request a specific CoS in an echo reply:

```
Router# ping mpls ipv4 10.131.159.252/32 reply dscp 50
 <0-63>   Differentiated services codepoint value
 af11     Match packets with AF11 dscp (001010)
 af12     Match packets with AF12 dscp (001100)
 af13     Match packets with AF13 dscp (001110)
 af21     Match packets with AF21 dscp (010010)
 af22     Match packets with AF22 dscp (010100)
 af23     Match packets with AF23 dscp (010110)
 af31     Match packets with AF31 dscp (011010)
 af32     Match packets with AF32 dscp (011100)
 af33     Match packets with AF33 dscp (011110)
 af41     Match packets with AF41 dscp (100010)
 af42     Match packets with AF42 dscp (100100)
 af43     Match packets with AF43 dscp (100110)
 cs1      Match packets with CS1(precedence 1) dscp (001000)
 cs2      Match packets with CS2(precedence 2) dscp (010000)
 cs3      Match packets with CS3(precedence 3) dscp (011000)
 cs4      Match packets with CS4(precedence 4) dscp (100000)
 cs5      Match packets with CS5(precedence 5) dscp (101000)
```

```
 cs6      Match packets with CS6(precedence 6) dscp (110000)
 cs7      Match packets with CS7(precedence 7) dscp (111000)
 default  Match packets with default dscp (000000)
 ef       Match packets with EF dscp (101110)
```

Controlling How a Responding Router Replies to an MPLS Echo Request Example

The following example checks MPLS LSP connectivity by using ipv4 reply mode:

```
Router# ping mpls ipv4 10.131.191.252/32 reply mode ipv4
```

Preventing Possible Loops with MPLS LSP Ping Example

The following example shows how a loop operates if you use the following **ping mpls** command:

```
Router# ping mpls
 ipv4
 10.131.159.251/32 destination 127.0.0.1 127.0.0.2 1 repeat 2
sweep 1450 1475 25
Sending 2, [1450..1500]-byte MPLS Echos to 10.131.159.251/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
Destination address 127.0.0.1
!
!
Destination address 127.0.0.2
!
!
Destination address 127.0.0.1
!
!
Destination address 127.0.0.2
!
!
```

A **ping mpls** command is sent for each packet size range for each destination address until the end address is reached. For this example, the loop continues in the same manner until the destination address, 127.0.0.5, is reached. The sequence continues until the number is reached that you specified with the **repeat** *count* keyword and argument. For this example, the repeat count is 2. The MPLS LSP ping loop sequence is as follows:

```
repeat  = 1
  destination address 1 (address-start
)
    for (size from sweep minimum
 to maximum
, counting by size-increment
)
```

```
      send an lsp ping
  destination address 2 (address-start
 +
address-
increment
)
    for (size from sweep minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
  destination address 3 (address-start
 +
address-
increment
 +
address-
increment
)
     for (size from sweep minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
  .
  .
  .
  until destination address = address-end
  .
  .
  .
  until repeat = count 2
```

Preventing Possible Loops with MPLS LSP Traceroute Example

The following example shows how a loop occurs if you use the following **trace mpls** command:

```
Router# trace mpls ipv4 10.131.159.251/32 destination 127.0.0.1 127.0.0.3 1 ttl 5
Tracing MPLS Label Switched Path to 10.131.159.251/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
Destination address 127.0.0.1
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 40 ms
Destination address 127.0.0.2
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 40 ms
Destination address 127.0.0.3
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 48 ms
```

An **mpls trace** command is sent for each TTL from 1 to the maximum TTL (**ttl** *maximum-time-to-live* keyword and argument) for each destination address until the address specified with the destination *end-address*

argument is reached. In this example, the maximum TTL is 5 and the end destination address is 127.0.0.3. The MPLS LSP traceroute loop sequence is as follows:

```
destination address 1 (address-start
)
  for (ttl from 1 to maximum-time-to-live
)
    send an lsp trace
destination address 2 (address-start
 + address-increment
)
  for (ttl from 1 to 5
)
    send an lsp trace
destination address 3 (address-start
 + address-increment
 + address-increment
)
  for (ttl from 1 to
maximum-time-to-live)
    send an lsp trace
.
.
.
until destination address = 4
```

The following example shows that the trace encountered an LSP problem at the router that has an IP address of 10.6.1.6:

```
Router# traceroute mpls ipv4 10.6.7.4/32
Tracing MPLS Label Switched Path to 10.6.7.4/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.6.1.14 MRU 4470 [Labels: 22 Exp: 0]
R 1 10.6.1.5 MRU 4470 [Labels: 21 Exp: 0] 2 ms
R 2 10.6.1.6 4 ms                  <------ Router address repeated for 2nd to 30th TTL.
R 3 10.6.1.6 1 ms
R 4 10.6.1.6 1 ms
R 5 10.6.1.6 3 ms
R 6 10.6.1.6 4 ms
R 7 10.6.1.6 1 ms
R 8 10.6.1.6 2 ms
R 9 10.6.1.6 3 ms
R 10 10.6.1.6 4 ms
R 11 10.6.1.6 1 ms
R 12 10.6.1.6 2 ms
R 13 10.6.1.6 4 ms
R 14 10.6.1.6 5 ms
R 15 10.6.1.6 2 ms
R 16 10.6.1.6 3 ms
R 17 10.6.1.6 4 ms
R 18 10.6.1.6 2 ms
R 19 10.6.1.6 3 ms
R 20 10.6.1.6 4 ms
R 21 10.6.1.6 1 ms
R 22 10.6.1.6 2 ms
```

```
R 23 10.6.1.6 3 ms
R 24 10.6.1.6 4 ms
R 25 10.6.1.6 1 ms
R 26 10.6.1.6 3 ms
R 27 10.6.1.6 4 ms
R 28 10.6.1.6 1 ms
R 29 10.6.1.6 2 ms
R 30 10.6.1.6 3 ms                    <------ TTL 30.
```

If you know the maximum number of hops in your network, you can set the TTL to a lower value with the **trace mpls ttl** *maximum-time-to-live* command. The following example shows the same **traceroute** command as the previous example, except that this time the TTL is set to 5:

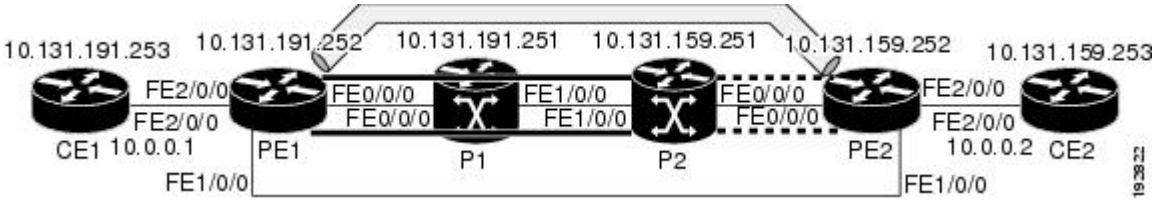
```
Router# traceroute mpls ipv4 10.6.7.4/32 ttl 5
Tracing MPLS Label Switched Path to 10.6.7.4/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.6.1.14 MRU 4470 [Labels: 22 Exp: 0]
R 1 10.6.1.5 MRU 4474 [No Label] 3 ms
R 2 10.6.1.6 4 ms                  <------ Router address repeated for 2nd to 5th TTL.
R 3 10.6.1.6 1 ms
R 4 10.6.1.6 3 ms
R 5 10.6.1.6 4 ms
```

Troubleshooting with LSP Ping or Traceroute Example

ICMP **ping** and **trace** commands are often used to help diagnose the root cause of a failure. When an LSP is broken, the packet may reach the target router by IP forwarding, thus making the ICMP ping and traceroute features unreliable for detecting MPLS forwarding problems. The MPLS LSP ping or traceroute and AToM VCCV features extend this diagnostic and troubleshooting ability to the MPLS network and handle inconsistencies (if any) between the IP and MPLS forwarding tables, inconsistencies in the MPLS control and data plane, and problems with the reply path.

The figure below shows a sample topology with an LDP LSP.

Figure 19: Sample Topology with LDP LSP



This section contains the following subsections:

Configuration for Sample Topology

These are sample topology configurations for the troubleshooting examples in the following sections (see the figure above). There are the six sample router configurations.

Router CE1 Configuration

Following is the configuration for the CE1 router:

```
!
version 2.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE1
!
boot-start-marker
boot-end-marker
!
enable password lab
!
clock timezone EST -5
ip subnet-zero
!
!
!
interface Loopback0
 ip address 10.131.191.253 255.255.255.255
 no ip directed-broadcast
 no clns route-cache
!
!
interface FastEthernet2/0/0
 no ip address
 no ip directed-broadcast
 no keepalive
 no cdp enable
 no clns route-cache
!
interface FastEthernet2/0/0.1
 encapsulation dot1Q 1000
 ip address 10.0.0.1 255.255.255.0
 no ip directed-broadcast
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
end
```

Router PE1 Configuration

Following is the configuration for the PE1 router:

```
!
version 2.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
```

```
!
boot-start-marker
boot-end-marker
!
logging snmp-authfail
enable password lab
!
clock timezone EST -5
ip subnet-zero
ip cef
no ip domain-lookup
!
mpls ldp discovery targeted-hello accept
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.131.191.252 255.255.255.255
 no clns route-cache
!
interface FastEthernet0/0/0
 ip address 10.131.191.230 255.255.255.252
ip rsvp bandwidth 1500 1500
ip rsvp signalling dscp 0
!
interface FastEthernet1/0/0
 ip address 10.131.159.246 255.255.255.252
 shutdown
 no clns route-cache
ip rsvp bandwidth 1500 1500
ip rsvp signalling dscp 0
!
interface FastEthernet2/0/0
 no ip address
 no cdp enable
 no clns route-cache
!
interface FastEthernet2/0/0.1
 encapsulation dot1Q 1000
 xconnect 10.131.159.252 333 encapsulation mpls
!
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.244 0.0.0.3 area 0
 network 10.131.191.228 0.0.0.3 area 0
 network 10.131.191.232 0.0.0.3 area 0
 network 10.131.191.252 0.0.0.0 area 0
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
!
end
```

Router P1 Configuration

Following is the configuration for the P1 router:

```
version 2.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1
!
boot-start-marker
boot-end-marker
!
logging snmp-authfail
enable password lab
!
clock timezone EST -5
ip subnet-zero
ip cef
no ip domain-lookup
!
!
mpls ldp discovery targeted-hello accept
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
!
!
 no clns route-cache
!
interface Loopback0
 ip address 10.131.191.251 255.255.255.255
 no clns route-cache
!
interface FastEthernet0/0/0
 ip address 10.131.191.229 255.255.255.252
 no clns route-cache
ip rsvp bandwidth 1500 1500
ip rsvp signalling dscp 0
!
interface FastEthernet1/0/0
 ip address 10.131.159.226 255.255.255.252
 no clns route-cache
ip rsvp bandwidth 1500 1500
ip rsvp signalling dscp 0
!
interface FastEthernet1/1/0
 ip address 10.131.159.222 255.255.255.252
 no clns route-cache
ip rsvp bandwidth 1500 1500
ip rsvp signalling dscp 0
!
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.220 0.0.0.3 area 0
 network 10.131.159.224 0.0.0.3 area 0
 network 10.131.191.228 0.0.0.3 area 0
 network 10.131.191.251 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
```

```
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
end
```

Router P2 Configuration

Following is the configuration for the P2 router:

```
!
version 2.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P2
!
boot-start-marker
boot-end-marker
!
enable password lab
!
clock timezone EST -5
ip subnet-zero
ip cef
no ip domain-lookup
!
mpls ldp discovery targeted-hello accept
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.131.159.251 255.255.255.255
 no ip directed-broadcast
!
interface FastEthernet0/0/0
 ip address 10.131.159.229 255.255.255.252
 no ip directed-broadcast
ip rsvp bandwidth 1500 1500
ip rsvp signalling dscp 0
!
interface FastEthernet0/1/0
 ip address 10.131.159.233 255.255.255.252
no ip directed-broadcast
ip rsvp signalling dscp 0
!
interface FastEthernet1/0/0
 ip address 10.131.159.225 255.255.255.252
no ip directed-broadcast
ip rsvp bandwidth 1500 1500
ip rsvp signalling dscp 0
!
interface FastEthernet1/1/0
 ip address 10.131.159.221 255.255.255.252
no ip directed-broadcast
```

```
ip rsvp signalling dscp 0
!
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.220 0.0.0.3 area 0
 network 10.131.159.224 0.0.0.3 area 0
 network 10.131.159.228 0.0.0.3 area 0
 network 10.131.159.232 0.0.0.3 area 0
 network 10.131.159.251 0.0.0.0 area 0
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
end
```

Router PE2 Configuration

Following is the configuration for the PE2 router:

```
!
version 2.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2
!
boot-start-marker
boot-end-marker
!
logging snmp-authfail
enable password lab
!
clock timezone EST -5
ip subnet-zero
ip cef
no ip domain-lookup
!
mpls ldp discovery targeted-hello accept
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
!
!
interface Loopback0
 ip address 10.131.159.252 255.255.255.255
 no clns route-cache
!
interface FastEthernet0/0/0
 ip address 10.131.159.230 255.255.255.252
 no clns route-cache
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
```

```
interface FastEthernet0/1/0
 ip address 10.131.159.234 255.255.255.252
 no clns route-cache
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface FastEthernet1/0/0
 ip address 10.131.159.245 255.255.255.252
 mpls ip
 no clns route-cache
!
interface FastEthernet3/0/0
 no ip address
 no cdp enable
 no clns route-cache
!
interface FastEthernet3/0/0.1
 encapsulation dot1Q 1000
 no snmp trap link-status
 no cdp enable
 xconnect 10.131.191.252 333 encapsulation mpls
!
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.122.0 0.0.0.3 area 0
 network 10.131.159.228 0.0.0.3 area 0
 network 10.131.159.232 0.0.0.3 area 0
 network 10.131.159.236 0.0.0.3 area 0
 network 10.131.159.244 0.0.0.3 area 0
 network 10.131.159.252 0.0.0.0 area 0
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
!
end
```

Router CE2 Configuration

Following is the configuration for the CE2 router:

```
!
version 2.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE2
!
boot-start-marker
boot-end-marker
!
enable password lab
!
clock timezone EST -5
```

```
ip subnet-zero
ip cef
no ip domain-lookup
!
!
interface Loopback0
 ip address 10.131.159.253 255.255.255.255
 no ip directed-broadcast
 no clns route-cache
!
interface FastEthernet3/0/0
 no ip address
 no ip directed-broadcast
 no keepalive
 no cdp enable
 no clns route-cache
!
interface FastEthernet3/0/0.1
 encapsulation dot1Q 1000
 ip address 10.0.0.2 255.255.255.0
 no ip directed-broadcast
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
end
```

Verification That the LSP Is Configured Correctly

Use the output from the **show** commands in this section to verify that the LSP is configured correctly.

A **show mpls forwarding-table** command shows that tunnel 1 is in the MPLS forwarding table.

```
PE1# show mpls forwarding-table 10.131.159.252
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     18
[T] 10.131.159.252/32 0          Tu1        point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

A **trace mpls** command issued at PE1 verifies that packets with 16 as the outermost label and 18 as the end-of-stack label are forwarded from PE1 to PE2.

```
PE1# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
   0 10.131.191.252 MRU 1496 [Labels: 16/18 Exp: 0/0] L 1 10.131.191.229
   MRU 1508 [Labels: 18 Exp: 0] 0 ms L 2 10.131.159.225
```

```
   MRU 1504 [Labels: implicit-null Exp: 0] 0 ms ! 3 10.131.159.234 20 ms
   PE1#
```

The MPLS LSP Traceroute to PE2 is successful, as indicated by the exclamation point (!).

Discovery of LSP Breaks

Use the output of the commands in this section to discover LSP breaks.

An LDP target session is established between routers PE1 and P2, as shown in the output of the following **show mpls ldp discovery** command:

```
PE1# show mpls ldp discovery
 Local LDP Identifier:
    10.131.191.252:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
        Tunnel1 (ldp): Targeted -> 10.131.159.251
    Targeted Hellos:
        10.131.191.252 -> 10.131.159.252 (ldp): active/passive, xmit/recv
            LDP Id: 10.131.159.252:0
        10.131.191.252 -> 10.131.159.251 (ldp): active, xmit/recv
LDP Id: 10.131.159.251:0
```

Enter the following command on the P2 router in global configuration mode:

```
P2(config)# no mpls ldp discovery targeted-hello accept
```

The LDP configuration change causes the targeted LDP session between the headend and tailend of the TE tunnel to go down. Labels for IPv4 prefixes learned by P2 are not advertised to PE1. Thus, all IP prefixes reachable by P2 are reachable by PE1 only through IP (not MPLS). In other words, packets destined for those prefixes through Tunnel 1 at PE1 will be IP switched at P2 (which is undesirable).

The following **show mpls ldp discovery** command shows that the LDP targeted session is down:

```
PE1# show mpls ldp discovery
 Local LDP Identifier:
    10.131.191.252:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
        Tunnel1 (ldp): Targeted -> 10.131.159.251
    Targeted Hellos:
        10.131.191.252 -> 10.131.159.252 (ldp): active/passive, xmit/recv
            LDP Id: 10.131.159.252:0
        10.131.191.252 -> 10.131.159.251 (ldp): active, xmit
```

Enter the **show mpls forwarding-table** command at the PE1 router. The display shows that the outgoing packets are untagged as a result of the LDP configuration changes.

```
PE1# show mpls forwarding-table 10.131.159.252
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     Untagged[T] 10.131.159.252/32 0          Tu1        point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

A **ping mpls** command entered at the PE1 router displays the following:

```
PE1# ping mpls ipv4 10.131.159.252/32 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
R
Success rate is 0 percent (0/1)
```

The **ping mpls** command fails. The R indicates that the sender of the MPLS echo reply had a routing entry but no MPLS FEC. Entering the **verbose** keyword with the**ping mpls** command displays the MPLS LSP echo reply sender address and the return code. You should be able to determine where the breakage occurred by telnetting to the replying router and inspecting its forwarding and label tables. You might need to look at the neighboring upstream router as well, because the breakage might be on the upstream router.

```
PE1# ping mpls ipv4 10.131.159.252/32 repeat 1 verbose
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
R   10.131.159.225, return code 6
Success rate is 0 percent (0/1)
```

Alternatively, use the LSP **traceroute** command to figure out which router caused the breakage. In the following example, for subsequent values of TTL greater than 2, the same router keeps responding (10.131.159.225). This suggests that the MPLS echo request keeps getting processed by the router regardless of the TTL. Inspection of the label stack shows that P1 pops the last label and forwards the packet to P2 as an IP packet. This explains why the packet keeps getting processed by P2. MPLS echo request packets cannot be forwarded by use of the destination address in the IP header because the address is set to a 127/8 address.

```
PE1# trace mpls ipv4 10.131.159.252/32 ttl 5
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.191.230 MRU 1496 [Labels: 22/19 Exp: 0/0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 40 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
pe1#
```

MTU Discovery in an LSP Example

The following example shows the results of a **trace mpls** command when the LSP is formed with labels created by LDP:

```
PE1# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.191.230 MRU 1496 [Labels: 22/19 Exp: 0/0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 40 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
pe1#
```

You can determine the MRU for the LSP at each hop through the use of the **show mpls forwarding detail** command:

```
PE1# show mpls forwarding 10.131.159.252 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     19          10.131.159.252/32 0          Tu1        point2point
        MAC/Encaps=14/22, MRU=1496, Tag Stack{22 19}, via Et0/0
        AABBCC009700AABBCC0098008847 0001600000013000
        No output feature configured
```

To determine how large an echo request will fit on the LSP, first calculate the size of the IP MTU by using the **show interface** *interface-name* command:

```
PE1# show interface e0/0
FastEthernet0/0/0 is up, line protocol is up
  Hardware is Lance, address is aabb.cc00.9800 (bia aabb.cc00.9800)
  Internet address is 10.131.191.230/30
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     377795 packets input, 33969220 bytes, 0 no buffer
     Received 231137 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     441772 packets output, 40401350 bytes, 0 underruns
     0 output errors, 0 collisions, 10 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The IP MTU in the **show interface** *interface-name* example is 1500 bytes. Subtract the number of bytes corresponding to the label stack from the MTU number. The output of the **show mpls forwarding** command indicates that the Tag stack consists of one label (21). Therefore, the largest MPLS echo request packet that can be sent in the LSP is 1500 - (2 x 4) = 1492.

You can validate this by using the following **mpls ping** command:

```
PE1# ping mpls ipv4 10.131.159.252/32 sweep 1492 1500 1 repeat 1
Sending 1, [1492..1500]-byte MPLS Echos to 10.131.159.252/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!QQQQQQQQ
Success rate is 11 percent (1/9), round-trip min/avg/max = 40/40/40 ms
```

In this command, echo packets that have a range in size from 1492 to 1500 bytes are sent to the destination address. Only packets of 1492 bytes are sent successfully, as indicated by the exclamation point (!). Packets of byte sizes 1493 to 1500 are source-quenched, as indicated by the Qs.

You can pad an MPLS echo request so that a payload of a given size can be tested. The pad TLV is useful when you use the MPLS echo request to discover the MTU that is supportable by an LSP. MTU discovery is extremely important for applications like AToM that contain non-IP payloads that cannot be fragmented.

Tracking Packets Tagged as Implicit Null Example

In the following example, Tunnel 1 is shut down, and only an LSP formed with LDP labels is established. An implicit null is advertised between the P2 and PE2 routers. Entering an MPLS LSP traceroute command at the PE1 router results in the following output that shows that packets are forwarded from P2 to PE2 with an implicit-null label. Address 10.131.159.229 is configured for the P2 Fast Ethernet 0/0/0 out interface for the PE2 router.

```
PE1# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.191.230 MRU 1496 [Labels: 22/19 Exp: 0/0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 40 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
pe1#
```

Tracking Untagged Packets Example

Untagged cases are valid configurations for IGP LSPs that could cause problems for MPLS VPNs.

A **show mpls forwarding-table** command and a **show mpls ldp discovery** command issued at the P2 router show that LDP is properly configured:

```
P2# show mpls forwarding-table 10.131.159.252
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
19     Pop tag     10.131.159.252/32 0          fe0/0/0    10.131.159.230
P2# show mpls ldp discovery
 Local LDP Identifier:
    10.131.159.251:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/0/0 (ldp): xmit/recv
            LDP Id: 10.131.159.252:0
        FastEthernet1/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
```

The **show mpls ldp discovery** command output shows that Fast Ethernet interface 0/0/0, which connects PE2 to P2, is sending and receiving packets.

If a **no mpls ip** command is entered on Fast Ethernet interface 0/0/0, this could prevent an LDP session between the P2 and PE2 routers from being established. A **show mpls ldp discovery** command entered on the PE router shows that the MPLS LDP session with the PE2 router is down.

```
P2# show mpls ldp discovery
 Local LDP Identifier:
    10.131.159.251:0
    Discovery Sources:
    Interfaces:

FastEthernet0/0/0 (ldp): xmit
        FastEthernet1/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
```

If the MPLS LDP session to PE2 goes down, the LSP to 10.131.159.252 becomes untagged, as shown by the **show mpls forwarding-table** command:

```
P2# show mpls forwarding-table 10.131.159.252/32
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
19     Untagged    10.131.159.252/32 864        fe0/0/0    10.131.159.230
```

Untagged cases would provide an MPLS LSP traceroute reply with packets tagged with No Label, as shown in the following display. You may need to reestablish an MPLS LSP session from interface P2 to PE2 by entering an **mpls ip** command on the output interface from P2 to PE2, which is Fast Ethernet 0/0/0 in this example:

```
PE1# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 20 Exp: 0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 80 ms
```

```
R 2 10.131.159.229 MRU 1504 [No Label] 28 ms       <----No MPLS session from P2 to PE2.
! 3 10.131.159.230 40 ms
```

Determining Why a Packet Could Not Be Sent Example

The following example shows a **ping mpls** command when an MPLS echo request is not sent. The transmission failure is shown by the returned Qs.

```
PE1# ping mpls ipv4 10.0.0.1/32
Sending 5, 100-byte MPLS Echos to 10.0.0.1/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
QQQQQ
Success rate is 0 percent (0/5)
```

The following **show mpls forwarding-table** command and **show ip route** command demonstrate that the IPv4 address (10.0.0.1)address is not in the LFIB or RIB routing table. Therefore, the MPLS echo request is not sent.

```
PE1# show mpls forwarding-table 10.0.0.1
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
PE1# show ip route 10.0.0.1
% Subnet not in table
```

Detecting LSP Breaks when Load Balancing Is Enabled for IPv4 LSPs Example

In the following examples, different paths are followed to the same destination. The output from these examples demonstrates that load balancing occurs between the originating router and the target router.

To ensure that Fast Ethernet interface 1/0/0 on the PE1 router is operational, enter the following commands on the PE1 router:

```
PE1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
PE1(config)# interface fastethernet 1/0/0
PE1(config-if)# no shutdown
PE1(config-if)# end
*Dec 31 19:14:10.034: %LINK-3-UPDOWN: Interface FastEthernet1/0/0, changed state to up
*Dec 31 19:14:11.054: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0/0,
changed state to upend
PE1#
*Dec 31 19:14:12.574: %SYS-5-CONFIG_I: Configured from console by console
*Dec 31 19:14:19.334: %OSPF-5-ADJCHG: Process 1, Nbr 10.131.159.252 on FastEthernet1/0/0
from LOADING to FULL, Loading Done
PE1#
```

The following **show mpls forwarding-table** command displays the possible outgoing interfaces and next hops for the prefix 10.131.159.251/32:

```
PE1# show mpls forwarding-table 10.131.159.251/32
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
21     19          10.131.159.251/32 0          fe0/0/0    10.131.191.229
       20          10.131.159.251/32 0          fe1/0/0    10.131.159.245
```

The following **ping mpls** command to 10.131.159.251/32 with a destination UDP address of 127.0.0.1 shows that the selected path has a path index of 0:

```
Router# ping mpls ipv4
 10.131.159.251/32 destination
 127.0.0.1/32
Sending 1, 100-byte MPLS Echos to 10.131.159.251/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
PE1#
*Dec 29 20:42:40.638: LSPV: Echo Request sent on IPV4 LSP, load_index 2,
pathindex 0, size 100
*Dec 29 20:42:40.638: 46 00 00 64 00 00 40 00 FF 11 9D 03 0A 83 BF FC
*Dec 29 20:42:40.638: 7F 00 00 01 94 04 00 00 0D AF 0D AF 00 4C 14 70
*Dec 29 20:42:40.638: 00 01 00 00 01 02 00 00 1A 00 00 1C 00 00 00 01
*Dec 29 20:42:40.638: C3 9B 10 40 A3 6C 08 D4 00 00 00 00 00 00 00 00
*Dec 29 20:42:40.638: 00 01 00 09 00 01 00 05 0A 83 9F FB 20 00 03 00
*Dec 29 20:42:40.638: 13 01 AB CD AB CD AB CD AB CD AB CD AB CD
*Dec 29 20:42:40.638: AB CD AB CD
*Dec 29 20:42:40.678: LSPV: Echo packet received: src 10.131.159.225,
dst 10.131.191.252, size 74
*Dec 29 20:42:40.678: AA BB CC 00 98 01 AA BB CC 00 FC 01 08 00 45 C0
*Dec 29 20:42:40.678: 00 3C 32 D6 00 00 FD 11 15 37 0A 83 9F E1 0A 83
*Dec 29 20:42:40.678: BF FC 0D AF 0D AF 00 28 D1 85 00 01 00 00 02 02
*Dec 29 20:42:40.678: 03 00 1A 00 00 1C 00 00 00 01 C3 9B 10 40 A3 6C
*Dec 29 20:42:40.678: 08 D4 C3 9B 10 40 66 F5 C3 C8
```

The following **ping mpls** command to 10.131.159.251/32 with a destination UDP address of 127.0.0.3 shows that the selected path has a path index of 1:

```
PE1# ping mpls ipv4 10.131.159.251/32 destination 127.0.0.3/32
Sending 1, 100-byte MPLS Echos to 10.131.159.251/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes:
    '!' - success, 'Q' - request not sent, '.' - timeout,
    'L' - labeled output interface, 'B' - unlabeled output interface,
    'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
    'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
    'P' - no rx intf label prot, 'p' - premature termination of LSP,
    'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
PE1#
*Dec 29 20:43:09.518: LSPV: Echo Request sent on IPV4 LSP, load_index 13,
pathindex 1, size 100
*Dec 29 20:43:09.518: 46 00 00 64 00 00 40 00 FF 11 9D 01 0A 83 BF FC
```

```
*Dec 29 20:43:09.518: 7F 00 00 03 94 04 00 00 0D AF 0D AF 00 4C 88 58
*Dec 29 20:43:09.518: 00 01 00 00 01 02 00 00 38 00 00 1D 00 00 00 01
*Dec 29 20:43:09.518: C3 9B 10 5D 84 B3 95 84 00 00 00 00 00 00 00 00
*Dec 29 20:43:09.518: 00 01 00 09 00 01 00 05 0A 83 9F FB 20 00 03 00
*Dec 29 20:43:09.518: 13 01 AB CD AB CD AB CD AB CD AB CD AB CD
*Dec 29 20:43:09.518: AB CD AB CD
*Dec 29 20:43:09.558: LSPV: Echo packet received: src 10.131.159.229,
dst 10.131.191.252, size 74
*Dec 29 20:43:09.558: AA BB CC 00 98 01 AA BB CC 00 FC 01 08 00 45 C0
*Dec 29 20:43:09.558: 00 3C 32 E9 00 00 FD 11 15 20 0A 83 9F E5 0A 83
*Dec 29 20:43:09.558: BF FC 0D AF 0D AF 00 28 D7 57 00 01 00 00 02 02
*Dec 29 20:43:09.558: 03 00 38 00 00 1D 00 00 00 01 C3 9B 10 5D 84 B3
*Dec 29 20:43:09.558: 95 84 C3 9B 10 5D 48 3D 50 78
```

To see the actual path chosen, enter the **debug mpls lspv** command with the **packet** and **data** keywords.

Note The load balancing algorithm attempts to uniformly distribute packets across the available output paths by hashing based on the IP header source and destination addresses. The selection of the *address-start* , *address-end* , and *address-increment* arguments for the **destination** keyword may not provide the expected results.

Specifying the Interface Through Which Echo Packets Leave a Router Example

The following example tests load balancing from the upstream router:

```
Router# ping mpls ipv4 10.131.161.251/32 ttl 1 repeat 1 dsmap hashkey ipv4 bitmap 8

Sending 1, 100-byte MPLS Echos to 10.131.161.251/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
L
Echo Reply received from 10.131.131.2
  DSMAP 0, DS Router Addr 10.131.141.130, DS Intf Addr 10.131.141.130
    Depth Limit 0, MRU 1500 [Labels: 54 Exp: 0]
    Multipath Addresses:
      127.0.0.3       127.0.0.5       127.0.0.7       127.0.0.8

  DSMAP 1, DS Router Addr 10.131.141.2, DS Intf Addr 10.131.141.2
    Depth Limit 0, MRU 1500 [Labels: 40 Exp: 0]
    Multipath Addresses:
      127.0.0.1       127.0.0.2       127.0.0.4       127.0.0.6
```

The following example validates that the transit router reported the proper results by determining the Echo Reply sender address two hops away and checking the rx label advertised upstream:

```
Success rate is 0 percent (0/1)
Router# trace mpls ipv4 10.131.161.251/32 destination 127.0.0.6 ttl 2
Tracing MPLS Label Switched Path to 10.131.161.251/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
```

```
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.131.1 10.131.131.2 MRU 1500 [Labels: 37 Exp: 0]
L 1 10.131.131.2 10.131.141.2 MRU 1500 [Labels: 40 Exp: 0] 0 ms, ret code 8
L 2 10.131.141.2 10.131.150.2 MRU 1504 [Labels: implicit-null Exp: 0] 0 ms, ret code 8
Router#
Router# telnet 10.131.141.2
Trying 10.131.141.2 ... Open
User Access Verification
Password:
Router> enable
The following example shows how the output interface
 keyword forces an LSP traceroute out FastEthernet interface 0/0/0:
Router# show mpls forwarding-table 10.131.159.251
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
20     19            10.131.159.251/32 0             fe1/0/0    10.131.159.245
       18            10.131.159.251/32 0             fe0/0/0    10.131.191.229
Router# trace mpls ipv4 10.131.159.251/32

Tracing MPLS Label Switched Path to 10.131.159.251/32, timeout is 2 seconds
Type escape sequence to abort.
  0 10.131.159.246 MRU 1500 [Labels: 19 Exp: 0]
L 1 10.131.159.245 MRU 1504 [Labels: implicit-null Exp: 0] 4 ms
! 2 10.131.159.229 20 ms
Router# trace mpls ipv4 10.131.159.251/32 output-interface fastethernet0/0/0
Tracing MPLS Label Switched Path to 10.131.159.251/32, timeout is 2 seconds
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 18 Exp: 0]
L 1 10.131.191.229 MRU 1504 [Labels: implicit-null Exp: 0] 0 ms
! 2 10.131.159.225 1 ms
```

Pacing the Transmission of Packets Example

The following example shows the pace of the transmission of packets:

```
Router# ping mpls ipv4 10.5.5.5/32 interval 100

Sending 5, 100-byte MPLS Echos to 10.5.5.5/32,
     timeout is 2 seconds, send interval is 100 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/36 ms PE-802
```

Interrogating the Transit Router for Its Downstream Information Example

The following example shows sample output when a router with two output paths is interrogated:

```
Router# ping mpls ipv4 10.161.251/32 ttl 4 repeat 1 dsmap hashkey ipv4 bitmap 16

Sending 1, 100-byte MPLS Echos to 10.131.161.251/32,
```

```
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
L
Echo Reply received from 10.131.131.2
  DSMAP 0, DS Router Addr 10.131.141.130, DS Intf Addr 10.131.141.130
    Depth Limit 0, MRU 1500 [Labels: 54 Exp: 0]
    Multipath Addresses:
      127.0.0.3       127.0.0.6       127.0.0.9       127.0.0.10
      127.0.0.12      127.0.0.13      127.0.0.14      127.0.0.15
      127.0.0.16
  DSMAP 1, DS Router Addr 10.131.141.2, DS Intf Addr 10.131.141.2
    Depth Limit 0, MRU 1500 [Labels: 40 Exp: 0]
    Multipath Addresses:
      127.0.0.1       127.0.0.2       127.0.0.4       127.0.0.5
      127.0.0.7       127.0.0.8       127.0.0.11
Success rate is 0 percent (0/1)
```

The multipath addresses cause a packet to transit to the router with the output label stack. The **ping mpls** command is useful for determining the number of output paths, but when the router is more than one hop away a router cannot always use those addresses to get the packet to transit through the router being interrogated. This situation exists because the change in the IP header destination address may cause the packet to be load-balanced differently by routers between the source router and the responding router. Load balancing is affected by the source address in the IP header. The following example tests load-balancing reporting from the upstream router:

```
Router# ping mpls ipv4 10.131.161.251/32 ttl 1 repeat 1 dsmap hashkey ipv4 bitmap 8

Sending 1, 100-byte MPLS Echos to 10.131.161.251/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
L
Echo Reply received from 10.131.131.2
  DSMAP 0, DS Router Addr 10.131.141.130, DS Intf Addr 10.131.141.130
    Depth Limit 0, MRU 1500 [Labels: 54 Exp: 0]
    Multipath Addresses:
      127.0.0.3       127.0.0.5       127.0.0.7       127.0.0.8

  DSMAP 1, DS Router Addr 10.131.141.2, DS Intf Addr 10.131.141.2
    Depth Limit 0, MRU 1500 [Labels: 40 Exp: 0]
    Multipath Addresses:
      127.0.0.1       127.0.0.2       127.0.0.4       127.0.0.6
To validate that the transit router reported the proper results, determine the Echo Reply
sender address that is two hops away and consistently check the rx label that is advertised
 upstream. The following is sample output:
Success rate is 0 percent (0/1)
Router# trace mpls ipv4 10.131.161.251/32 destination 127.0.0.6 ttl 2
Tracing MPLS Label Switched Path to 10.131.161.251/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
```

```
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.131.1 10.131.131.2 MRU 1500 [Labels: 37 Exp: 0]
L 1 10.131.131.2 10.131.141.2 MRU 1500 [Labels: 40 Exp: 0] 0 ms, ret code 8
L 2 10.131.141.2 10.131.150.2 MRU 1504 [Labels: implicit-null Exp: 0] 0 ms, ret code 8
Router#
Router# telnet 10.131.141.2

Trying 10.131.141.2 ... Open
User Access Verification
Password:
Router> enable
Router# show mpls forwarding-table 10.131.161.251

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
40     Pop tag     10.131.161.251/32 268        fe1/0/0    10.131.150.2
Router#
```

Interrogating a Router for Its DSMAP Example

The following example interrogates the software and hardware forwarding layer for their depth limit that needs to be returned in the DSMAP TLV.

```
Router# ping mpls ipv4 10.131.159.252/32 ttl 1 dsmap
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes:
  '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
L
Echo Reply received from 10.131.191.229
  DSMAP 0, DS Router Addr 10.131.159.225, DS Intf Addr 10.131.159.225
    Depth Limit 0, MRU 1508 [Labels: 18 Exp: 0]
    Multipath Addresses:
      127.0.0.1       127.0.0.2       127.0.0.3       127.0.0.4
      127.0.0.5       127.0.0.6       127.0.0.7       127.0.0.8
      127.0.0.9       127.0.0.10     127.0.0.11     127.0.0.12
      127.0.0.13     127.0.0.14     127.0.0.15     127.0.0.16
      127.0.0.17     127.0.0.18     127.0.0.19     127.0.0.20
      127.0.0.21     127.0.0.22     127.0.0.23     127.0.0.24
      127.0.0.25     127.0.0.26     127.0.0.27     127.0.0.28
      127.0.0.29     127.0.0.30     127.0.0.31     127.0.0.32
Success rate is 0 percent (0/1)
```

Requesting that a Transit Router Validate the Target FEC Stack Example

The following example causes a transit router to validate the target FEC stack by which an LSP to be tested is identified:

```
Router# trace mpls ipv4 10.5.5.5/32 flags fec
```

```
Tracing MPLS Label Switched Path to 10.5.5.5/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.2.3.2 10.2.3.3 MRU 1500 [Labels: 19 Exp: 0] L 1 10.2.3.3 10.3.4.4 MRU 1500 [Labels:
 19 Exp: 0] 40 ms, ret code 8 L 2 10.3.4.4 10.4.5.5 MRU 1504 [Labels: implicit-null Exp:
0] 32 ms, ret code 8 ! 3 10.4.5.5 40 ms, ret code 3
Router# ping mpls ipv4 10.5.5.5/32

Sending 5, 100-byte MPLS Echos to 10.5.5.5/32
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!    size 100, reply addr 10.4.5.5, return code 3
!    size 100, reply addr 10.4.5.5, return code 3
!    size 100, reply addr 10.4.5.5, return code 3
!    size 100, reply addr 10.4.5.5, return code 3
!    size 100, reply addr 10.4.5.5, return code 3
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
```

Enabling LSP Ping to Detect LSP Breakages Caused by Untagged Interfaces Example

The following example shows the extra label that is added to the end of the label stack when there is explicit-null label shimming:

```
Switch# trace mpls ipv4 10.131.159.252/32 force-explicit-null

Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes:
  '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.191.252 MRU 1492 [Labels: 16/18/explicit-null Exp: 0/0/0]
L 1 10.131.191.229 MRU 1508 [Labels: 18/explicit-null Exp: 0/0] 0 ms
L 2 10.131.159.225 MRU 1508 [Labels: explicit-null Exp: 0] 0 ms
! 3 10.131.159.234 4 ms
```

The following example shows the command output when there is not explicit-null label shimming:

```
Switch# trace mpls ipv4 10.131.159.252/32

Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
```

```
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.131.191.252 MRU 1496 [Labels: 16/18 Exp: 0/0]
L 1 10.131.191.229 MRU 1508 [Labels: 18 Exp: 0] 4 ms
L 2 10.131.159.225 MRU 1504 [Labels: implicit-null Exp: 0] 4 ms
! 3 10.131.159.234 4 ms
```

Viewing the AToM VCCV Capabilities Advertised to and Received from the Peer Example

The following example shows that router PE1 advertises both AToM VCCV Type 1 and Type 2 switching capabilities and that the remote router PE2 advertises only a Type 2 switching capability.

```
Router# show mpls l2transport binding

  Destination Address: 10.131.191.252,  VC ID: 333
    Local Label:  16
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV Capabilities: Type 1, Type 2 <----- Locally advertised VCCV capabilities
    Remote Label: 19
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV Capabilities: Type 2           <-----Remotely advertised VCCV capabilities
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS Transport Profile configuration document	MPLS Transport Profile

Standards and RFCs

Standard/RFC	Title
draft-ietf-mpls-tp-te-mib-02.txt	MPLS-TP Traffic Engineering (TE) Management Information Base (MIB)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LSP Ping Traceroute for LDP TE and LSP Ping for VCCV

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 20: Feature Information for MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV

Feature Name	Releases	Feature Information
MPLS Embedded Management LSP Ping/Traceroute for LDP	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.
MPLS Embedded Management LSP Ping/Trace for LDP and Resource Reservation Protocol (RSVP) IPv4 Forwarding Equivalence Classes (FECs)	Cisco IOS XE Release 2.3	The MPLS Embedded Management LSP Ping/Trace for LDP feature was modified to include support for RSVP IPv4 FECs.
MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV	Cisco IOS XE Release 2.3	The MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV feature helps service providers monitor label switched paths and quickly isolate MPLS forwarding problems. The following commands were introduced or modified: **debug mpls lspv**, **echo**, **mpls oam**, **ping mpls**, **show mpls oam**, **echo statistics**, **trace mpls**.

Glossary

FEC --forwarding equivalence class. A set of packets that can be handled equivalently for forwarding purposes and are thus suitable for binding to a single label. Examples include the set of packets destined for one address prefix and the packets in any flow.

flow --A set of packets traveling between a pair of hosts, or between a pair of transport protocol ports on a pair of hosts. For example, packets with the same source address, source port, destination address, and destination port might be considered a flow.

A flow is also a stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

fragmentation --The process of breaking a packet into smaller units when they are to be transmitted over a network medium that cannot support the original size of the packet.

ICMP -- Internet Control Message Protocol. A network layer Internet protocol that reports errors and provides other information relevant to IP packet processing. It is documented in RFC 792.

LFIB --Label Forwarding Information Base. A data structure and way of managing forwarding in which destinations and incoming labels are associated with outgoing interfaces and labels.

localhost --A name that represents the host router (device). The localhost uses the reserved loopback IP address 127.0.0.1.

LSP --label switched path. A connection between two routers in which MPLS forwards the packets.

LSPV --Label Switched Path Verification. An LSP Ping subprocess. It encodes and decodes MPLS echo requests and replies, and it interfaces with IP, MPLS, and AToM switching for sending and receiving MPLS echo requests and replies. At the MPLS echo request originator router, LSPV maintains a database of outstanding echo requests for which echo responses have not been received.

MPLS router alert label--An MPLS label of 1. An MPLS packet with a router alert label is redirected by the router to the Route Processor (RP) processing level for handling. This allows these packets to bypass any forwarding failures in hardware routing tables.

MRU --maximum receive unit. Maximum size, in bytes, of a labeled packet that can be forwarded through an LSP.

MTU --maximum transmission unit. Maximum packet size, in bytes, that a particular interface can send or receive.

punt --Redirect packets with a router alert from the line card or interface to Route Processor (RP) level processing for handling.

PW --pseudowire. A form of tunnel that carries the essential elements of an emulated circuit from one provider edge (PE) router to another PE router over a packet-switched network.

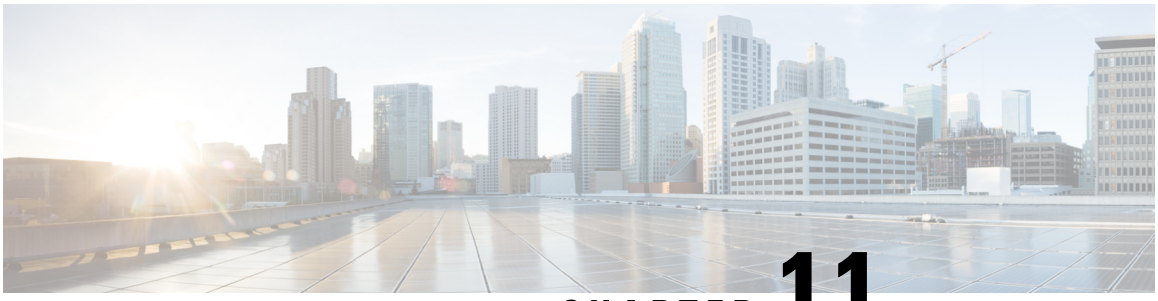
RP --Route Processor. The processor module in a Cisco 7000 series router that contains the CPU, system software, and most of the memory components that are used in the router. It is sometimes called a supervisory processor.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive. RSVP depends on IPv6. Is is also known as Resource Reservation Setup Protocol.

TLV --type, length, values. A block of information included in a Cisco Discovery Protocol address.

TTL hiding--Time-to-live is a parameter you can set that indicates the maximum number of hops a packet should take to reach its destination.

UDP --User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, so error processing and retransmission must be handled by other protocols. UDP is defined in RFC 768.



CHAPTER 11

MPLS LSP Ping, Traceroute, and AToM VCCV

As Multiprotocol Label Switching (MPLS) deployments increase and the traffic types they carry increase, the ability of service providers to monitor label switched paths (LSPs) and quickly isolate MPLS forwarding problems is critical to their ability to offer services. The MPLS LSP Ping, Traceroute, and AToM VCCV feature helps them mitigate these challenges.

The MPLS LSP Ping, Traceroute, and AToM VCCV feature can detect when an LSP fails to deliver user traffic.

- You can use MPLS LSP Ping to test LSP connectivity for IPv4 Label Distribution Protocol (LDP) prefixes, traffic engineering (TE) Forwarding Equivalence Classes (FECs), and AToM FECs.
- You can use MPLS LSP Traceroute to trace the LSPs for IPv4 LDP prefixes and TE tunnel FECs.
- Any Transport over MPLS Virtual Circuit Connection Verification (AToM VCCV) allows you to use MPLS LSP Ping to test the pseudowire (PW) section of an AToM virtual circuit (VC).

Internet Control Message Protocol (ICMP) ping and trace are often used to help diagnose the root cause when a forwarding failure occurs. The MPLS LSP Ping, Traceroute, and AToM VCCV feature extends this diagnostic and troubleshooting ability to the MPLS network and aids in the identification of inconsistencies between the IP and MPLS forwarding tables, inconsistencies in the MPLS control and data plane, and problems with the reply path.

The MPLS LSP Ping, Traceroute, and AToM VCCV feature uses MPLS echo request and reply packets to test LSPs. The Cisco implementation of MPLS echo request and echo reply are based on the Internet Engineering Task Force (IETF) Internet-Draft *Detecting MPLS Data Plane Failures.*

Prerequisites for MPLS LSP Ping, Traceroute, and AToM VCCV

Before you use the MPLS LSP Ping, Traceroute, and AToM VCCV feature, you should:

- Determine the baseline behavior of your Multiprotocol Label Switching (MPLS) network. For example:

  - What is the expected MPLS experimental (EXP) treatment?
  - What is the expected maximum size packet or maximum transmission unit (MTU) of the label switched path?
  - What is the topology? What are the expected label switched paths? How many links in the label switching path (LSP)? Trace the paths of the label switched packets including the paths for load balancing.
- Understand how to use MPLS and MPLS applications, including traffic engineering, Any Transport over MPLS (AToM), and Label Distribution Protocol (LDP). You need to
  - Know how LDP is configured
  - Understand AToM concepts
- Understand label switching, forwarding, and load balancing.

Restrictions for MPLS LSP Ping, Traceroute, and AToM VCCV

- You cannot use MPLS LSP Traceroute to trace the path taken by Any Transport over Multiprotocol Label Switching (AToM) packets. MPLS LSP Traceroute is not supported for AToM. (MPLS LSP Ping is supported for AToM.) However, you can use MPLS LSP Traceroute to troubleshoot the Interior Gateway Protocol (IGP) LSP that is used by AToM.
- You cannot use MPLS LSP Ping or Traceroute to validate or trace MPLS Virtual Private Networks (VPNs).
- You cannot use MPLS LSP Traceroute to troubleshoot label switching paths (LSPs) that employ time-to-live (TTL) hiding.

Information About MPLS LSP Ping, Traceroute, and AToM VCCV

MPLS LSP Ping Operation

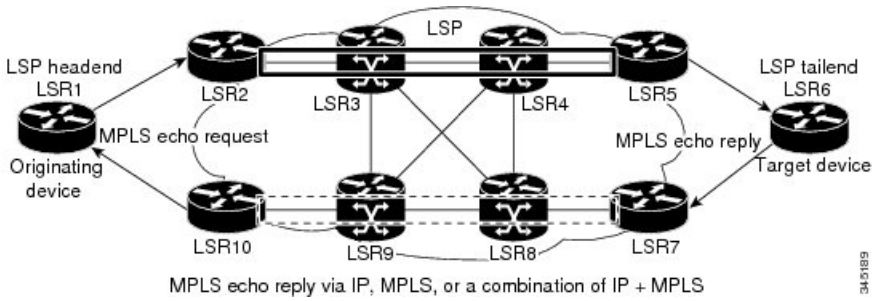
MPLS LSP Ping uses Multiprotocol Label Switching (MPLS) echo request and reply packets to validate a label switched path (LSP). Both an MPLS echo request and an MPLS echo reply are User Datagram Protocol (UDP) packets with source and destination ports set to 3503.

The MPLS echo request packet is sent to a target device through the use of the appropriate label stack associated with the LSP to be validated. Use of the label stack causes the packet to be switched inband of the LSP (that is, forwarded over the LSP itself). The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination address of the UDP packet is defined as a 127.*x* .*y* .*z* /8 address. This prevents the IP packet from being IP switched to its destination if the LSP is broken.

An MPLS echo reply is sent in response to an MPLS echo request. It is sent as an IP packet and forwarded using IP, MPLS, or a combination of both types of switching. The source address of the MPLS echo reply packet is an address from the device generating the echo reply. The destination address is the source address of the device in the MPLS echo request packet.

The figure below shows the echo request and echo reply paths for MPLS LSP Ping.

Figure 20: MPLS LSP Ping Echo Request and Echo Reply Paths



If you initiate an MPLS LSP Ping request at LSR1 to a Forwarding Equivalence Class (FEC), at LSR6, you get the results shown in the table below .

Table 21: MPLS LSP Ping Example

Step	Device	Action
1.	LSR1	Initiates an MPLS LSP Ping request for an FEC at the target device LSR6 and sends an MPLS echo request to LSR2.
1.	LSR2	Receives and forwards the MPLS echo request packet through transit devices LSR3 and LSR4 to the penultimate device LSR5.
1.	LSR5	Receives the MPLS echo request, pops the MPLS label, and forwards the packet to LSR6 as an IP packet.
1.	LSR6	Receives the IP packet, processes the MPLS echo request, and sends an MPLS echo reply to LSR1 through an alternate route.
1.	LSR7 to LSR10	Receive and forward the MPLS echo reply back toward LSR1, the originating device.
1.	LSR1	Receives the MPLS echo reply in response to the MPLS echo request.

You can use MPLS LSP Ping to validate IPv4 Label Distribution Protocol (LDP), Any Transport over MPLS (AToM), and IPv4 Resource Reservation Protocol (RSVP) FECs by using appropriate keywords and arguments with the command:

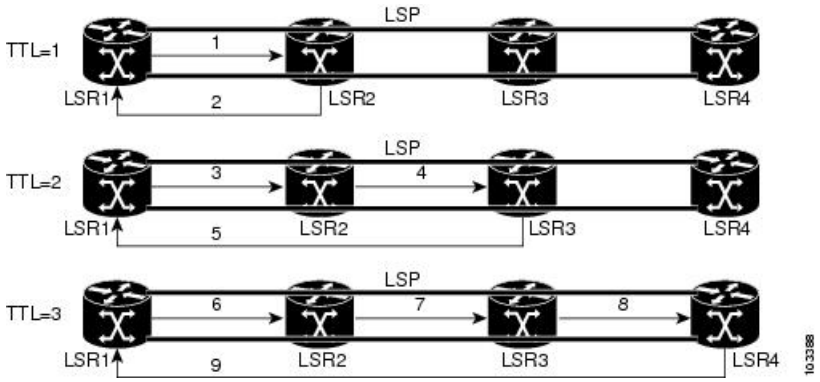
MPLS LSP Traceroute Operation

MPLS LSP Traceroute also uses Multiprotocol Label Switching (MPLS) echo request and reply packets to validate a label switched path (LSP). The echo request and echo reply are User Datagram Protocol (UDP) packets with source and destination ports set to 3503.

The MPLS LSP Traceroute feature uses time-to-live (TTL) settings to force expiration of the TTL along an LSP. MPLS LSP Traceroute incrementally increases the TTL value in its MPLS echo requests (TTL = 1, 2, 3, 4, ...) to discover the downstream mapping of each successive hop. The success of the LSP traceroute depends on the transit device processing the MPLS echo request when it receives a labeled packet with a TTL of 1. On Cisco devices, when the TTL expires, the packet is sent to the Route Processor (RP) for processing. The transit device returns an MPLS echo reply containing information about the transit hop in response to the TTL-expired MPLS packet.

The figure below shows an MPLS LSP Traceroute example with an LSP from LSR1 to LSR4.

Figure 21: MPLS LSP Traceroute Example



If you enter an LSP traceroute to a Forwarding Equivalence Class (FEC) at LSR4 from LSR1, you get the results shown in the table below.

Table 22: MPLS LSP Traceroute Example

Step	Device	MPLS Packet Type and Description	Device Action
1.	LSR1	MPLS echo request—With a target FEC pointing to LSR4 and to a downstream mapping.	• Sets the TTL of the label stack to 1. • Sends the request to LSR2.
1.	LSR2	MPLS echo reply.	Receives packet with TTL = 1. • Processes the UDP packet as an MPLS echo request. • Finds a downstream mapping, replies to LSR1 with its own downstream mapping based on the incoming label, and sends a reply.
1.	LSR1	MPLS echo request—With the same target FEC and the downstream mapping received in the echo reply from LSR2.	• Sets the TTL of the label stack to 2. • Sends the request to LSR2.
1.	LSR2	MPLS echo request.	Receives packet with TTL = 2. • Decrements the TTL. • Forwards the echo request to LSR3.
1.	LSR3	MPLS reply packet.	Receives packet with TTL = 1. • Processes the UDP packet as an MPLS echo request. • Finds a downstream mapping and replies to LSR1 with its own downstream mapping based on the incoming label.
1.	LSR1	MPLS echo request—With the same target FEC and the downstream mapping received in the echo reply from LSR3.	• Sets the TTL of the packet to 3. • Sends the request to LSR2.

Step	Device	MPLS Packet Type and Description	Device Action
1.	LSR2	MPLS echo request.	Receives packet with TTL = 3. • Decrements the TTL. • Forwards the echo request to LSR3.
1.	LSR3	MPLS echo request.	Receives packet with TTL = 2 • Decrements the TTL. • Forwards the echo request to LSR4.
1.	LSR4	MPLS echo reply.	Receives packet with TTL = 1. • Processes the UDP packet as an MPLS echo request. • Finds a downstream mapping and also finds that the device is the egress device for the target FEC. • Replies to LSR1.

You can use MPLS LSP Traceroute to validate IPv4 Label Distribution Protocol (LDP) and IPv4 RSVP FECs by using appropriate keywords and arguments with the **trace mpls** command:

By default, the TTL is set to 30. Therefore, the traceroute output always contains 30 lines, even if an LSP problem exists. This might mean duplicate entries in the output, should an LSP problem occur. The device address of the last point that the trace reaches is repeated until the output is 30 lines. You can ignore the duplicate entries. The following example shows that the trace encountered an LSP problem at the device that has an IP address of 10.6.1.6:

```
Device# traceroute mpls ipv4 10.6.7.4/32
Tracing MPLS Label Switched Path to 10.6.7.4/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.6.1.14 MRU 4470 [Labels: 22 Exp: 0]
R 1 10.6.1.5 MRU 4470 [Labels: 21 Exp: 0] 2 ms
R 2 10.6.1.6 4 ms                  <------ Router address repeated for 2nd to 30th TTL.
R 3 10.6.1.6 1 ms
R 4 10.6.1.6 1 ms
R 5 10.6.1.6 3 ms
R 6 10.6.1.6 4 ms
R 7 10.6.1.6 1 ms
R 8 10.6.1.6 2 ms
R 9 10.6.1.6 3 ms
R 10 10.6.1.6 4 ms
R 11 10.6.1.6 1 ms
R 12 10.6.1.6 2 ms
R 13 10.6.1.6 4 ms
R 14 10.6.1.6 5 ms
R 15 10.6.1.6 2 ms
R 16 10.6.1.6 3 ms
R 17 10.6.1.6 4 ms
R 18 10.6.1.6 2 ms
R 19 10.6.1.6 3 ms
```

```
R 20 10.6.1.6 4 ms
R 21 10.6.1.6 1 ms
R 22 10.6.1.6 2 ms
R 23 10.6.1.6 3 ms
R 24 10.6.1.6 4 ms
R 25 10.6.1.6 1 ms
R 26 10.6.1.6 3 ms
R 27 10.6.1.6 4 ms
R 28 10.6.1.6 1 ms
R 29 10.6.1.6 2 ms
R 30 10.6.1.6 3 ms                    <------ TTL 30.
```

If you know the maximum number of hops in your network, you can set the TTL to a smaller value with the **trace mpls ttl** *maximum-time-to-live* command. The following example shows the same **traceroute** command as the previous example, except that this time the TTL is set to 5.

```
Device# traceroute mpls ipv4 10.6.7.4/32 ttl 5
Tracing MPLS Label Switched Path to 10.6.7.4/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.6.1.14 MRU 4470 [Labels: 22 Exp: 0]
R 1 10.6.1.5 MRU 4474 [No Label] 3 ms
R 2 10.6.1.6 4 ms                  <------ Router address repeated for 2nd to 5th TTL.
R 3 10.6.1.6 1 ms
R 4 10.6.1.6 3 ms
R 5 10.6.1.6 4 ms
```

Any Transport over MPLS Virtual Circuit Connection Verification

AToM Virtual Circuit Connection Verification (AToM VCCV) allows the sending of control packets inband of an AToM pseudowire (PW) from the originating provider edge (PE) device. The transmission is intercepted at the destination PE device, instead of being forwarded to the customer edge (CE) device. This capability allows you to use MPLS LSP Ping to test the PW section of AToM virtual circuits (VCs).

AToM VCCV consists of the following:

- A signaled component in which the AToM VCCV capabilities are advertised during VC label signaling
- A switching component that causes the AToM VC payload to be treated as a control packet

AToM VCCV Signaling

One of the steps involved in Any Transport over Multiprotocol Label Switching (AToM) virtual circuit (VC) setup is the signaling of VC labels and AToM Virtual Circuit Connection Verification (VCCV) capabilities between AToM VC endpoints. The device uses an optional parameter, defined in the Internet Draft *draft-ieft-pwe3-vccv-01.txt,* to communicate the AToM VCCV disposition capabilities of each endpoint.

The AToM VCCV disposition capabilities are categorized as follows:

- Applications—MPLS LSP Ping and Internet Control Message Protocol (ICMP) Ping are applications that AToM VCCV supports to send packets inband of an AToM PW for control purposes.
- Switching modes—Type 1 and Type 2 are switching modes that AToM VCCV uses for differentiating between control and data traffic.

The table below describes AToM VCCV Type 1 and Type 2 switching modes.

Table 23: Type 1 and Type 2 AToM VCCV Switching Modes

Switching Mode	Description
Type 1	Uses a Protocol ID (PID) field in the AToM control word to identify an AToM VCCV packet.
Type 2	Uses an MPLS Router Alert Label above the VC label to identify an AToM VCCV packet.

Selection of AToM VCCV Switching Types

Cisco devices always use Type 1 switching, if available, when they send MPLS LSP Ping packets over an Any Transport over Multiprotocol Label Switching (AToM) virtual circuit (VC) control channel. Type 2 switching accommodates those VC types and implementations that do not support or interpret the AToM control word.

The table below shows the AToM Virtual Circuit Connection Verification (VCCV) switching mode advertised and the switching mode selected by the AToM VC.

Table 24: AToM VCCV Switching Mode Advertised and Selected by AToM Virtual Circuit

Type Advertised	Type Selected
AToM VCCV not supported	–
Type 1 AToM VCCV switching	Type 1 AToM VCCV switching
Type 2 AToM VCCV switching	Type 2 AToM VCCV switching
Type 1 and Type 2 AToM VCCV switching	Type 1 AToM VCCV switching

An AToM VC advertises its AToM VCCV disposition capabilities in both directions: that is, from the originating device (PE1) to the destination device (PE2), and from PE2 to PE1.

In some instances, AToM VCs might use different switching types if the two endpoints have different AToM VCCV capabilities. If PE1 supports Type 1 and Type 2 AToM VCCV switching and PE2 supports only Type 2 AToM VCCV switching, there are two consequences:

- LSP ping packets sent from PE1 to PE2 are encapsulated with Type 2 switching.
- LSP ping packets sent from PE2 to PE1 use Type 1 switching.

You can determine the AToM VCCV capabilities advertised to and received from the peer by entering the **show mpls l2transport binding** command at the PE device. For example:

```
Device# show mpls l2transport binding

  Destination Address: 10.131.191.252,  VC ID: 333
    Local Label:  16
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV Capabilities: Type 1, Type 2
    Remote Label: 19
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV Capabilities: Type 1
```

Command Options for ping mpls and trace mpls

MPLS LSP Ping and Traceroute command options are specified as keywords and arguments on the **ping mpls** and **trace mpls** commands.

The **ping mpls** command provides the options displayed in the command syntax below:

The **trace mpls** command provides the options displayed in the command syntax below:

Selection of FECs for Validation

A label switched path (LSP) is formed by labels. Devices learn labels through the Label Distribution Protocol (LDP), traffic engineering (TE), Any Transport over Multiprotocol Label Switching (AToM), or other MPLS applications. You can use MPLS LSP Ping and Traceroute to validate an LSP used for forwarding traffic for a given Forwarding Equivalence Class (FEC). The table below lists the keywords and arguments for the **ping mpls** and **traceroute mpls** commands that allow the selection of an LSP for validation.

Table 25: Selection of LSPs for Validation

FEC Type	ping mpls Keyword and Argument	traceroute mpls Keyword and Argument
LDP IPv4 prefix	**ipv4** *destination-address destination-mask*	**ipv4** *destination-address destination-mask*
MPLS TE tunnel	**traffic-eng** *tunnel-interface tunnel-number*	**traffic-eng** *tunnel-interface tunnel-number*
AToM VC	**pseudowire** *ipv4-address* **vc-id** *vc-id*	MPLS LSP Traceroute does not support the AToM tunnel LSP type for this release.

Reply Mode Options for MPLS LSP Ping and Traceroute

The reply mode is used to control how the responding device replies to a Multiprotocol Label Switching (MPLS) echo request sent by an MPLS LSP Ping or MPLS LSP Traceroute command. The table below describes the reply mode options.

Table 26: Reply Mode Options for a Responding Device

Option	Description
ipv4	Reply with an IPv4 User Datagram Protocol (UDP) packet (default). This is the most common reply mode selected for use with an MPLS LSP Ping and Traceroute command when you want to periodically poll the integrity of a label switched path (LSP). With this option, you do not have explicit control over whether the packet traverses IP or MPLS hops to reach the originator of the MPLS echo request. If the headend device fails to receive a reply, select the router-alert option, “Reply with an IPv4 UDP packet with a router alert.” The responding device sets the IP precedence of the reply packet to 6. You implement this option using the **reply mode ipv4** keywords.

Option	Description
router-alert	Reply with an IPv4 UDP packet with a device alert. This reply mode adds the router alert option to the IP header. This forces the packet to be special handled by the Cisco device at each intermediate hop as it moves back to the destination. This reply mode is more expensive, so use the router-alert option only if you are unable to get a reply with the ipv4 option, “Reply with an IPv4 UDP packet.” You implement this option using the **reply mode router-alert** keywords

The reply with an IPv4 UDP packet implies that the device should send an IPv4 UDP packet in reply to an MPLS echo request. If you select the ipv4 reply mode, you do not have explicit control over whether the packet uses IP or MPLS hops to reach the originator of the MPLS echo request. This is the mode that you would normally use to test and verify LSPs.

The reply with an IPv4 UDP packet that contains a device alert forces the packet to go back to the destination and be processed by the Route Processor (RP) process switching at each intermediate hop. This bypasses hardware/line card forwarding table inconsistencies. You should select this option when the originating (headend) devices fail to receive a reply to the MPLS echo request.

You can instruct the replying device to send an echo reply with the IP router alert option by using one of the following commands:

or

However, the reply with a router alert adds overhead to the process of getting a reply back to the originating device. This method is more expensive to process than a reply without a router alert and should be used only if there are reply failures. That is, the reply with a router alert label should only be used for MPLS LSP Ping or MPLS LSP Traceroute when the originating (headend) device fails to receive a reply to an MPLS echo request.

Packet Handling Along Return Path with an IP MPLS Router Alert

When an IP packet that contains an IP router alert option in its IP header or a Multiprotocol Label Switching (MPLS) packet with a router alert label as its outermost label arrives at a device, the device punts (redirects) the packet to the Route Processor (RP) process level for handling. This allows these packets to bypass the forwarding failures in hardware routing tables. The table below describes how IP and MPLS packets with an IP router alert option are handled by the device switching path processes.

Table 27: Switching Path Process Handling of IP and MPLS Router Alert Packets

Incoming Packet	Normal Switching Action	Process Switching Action	Outgoing Packet
IP packet—Router alert option in IP header	A rRouter alert option in the IP header causes the packet to be punted to the process switching path.	Forwards the packet as is.	IP packet—Router alert option in IP header.
	A router alert option in theIP header causes the packet to be punted to the process switching path.	Adds a router alert as the outermost label and forwards as an MPLS packet.	MPLS packet— Outermost label contains a router alert.

Incoming Packet	Normal Switching Action	Process Switching Action	Outgoing Packet
MPLS packet—Outermost label contains a router alert	If the router alert label is the outermost label, the packet is punted to the process switching path.	Removes the outermost router alert label, adds an IP router alert option to the IP header, and forwards as an IP packet.	IP packet—Router alert option in IP header.
	If the router alert label is the outermost label, the packet is punted to the process switching path.	Preserves the outermost router alert label and forwards the MPLS packet.	MPLS packet— Outermost label contains a router alert.

Other MPLS LSP Ping and Traceroute Command Options

The table below describes other MPLS LSP Ping and Traceroute command options that can be specified as keywords or arguments with the **ping mpls** command, or with both the **ping mpls** and **trace mpls** commands. Options available to use only on the **ping mpls** command are indicated as such.

Table 28: Other MPLS LSP Ping and Traceroute and AToM VCCV Options

Option	Description
Datagram size	Size of the packet with the label stack imposed. Specified with the **size** *packet-size* keyword and argument. The default size is 100. For use with the MPLS LSP Ping feature only.
Padding	Padding (the pad time-length-value [TLV]) is used as required to fill the datagram so that the MPLS echo request (User Datagram Protocol [UDP] packet with a label stack) is the size specified. Specify with the **pad** *pattern* keyword and argument. For use with the MPLS LSP Ping feature only.
Sweep size range	Parameter that enables you to send a number of packets of different sizes, ranging from a start size to an end size. This parameter is similar to the Internet Control Message Protocol (ICMP) ping sweep parameter. The lower boundary on the sweep range varies depending on the label switched path (LSP) type. You can specify a sweep size range when you use the **ping mpls** command. Use the **sweep** *minimum maximum size-increment* keyword and arguments. For use with the MPLS LSP Ping feature only.
Repeat count	Number of times to resend the same packet. The default is 5 times. You can specify a repeat count when you use the **ping mpls** command. Use the **repeat** *count* keyword and argument. For use with the MPLS LSP Ping feature only.
MPLS echo request source address	Routable address of the sender. The default address is loopback0. This address is used as the destination address in the Multiprotocol Label Switching (MPLS) echo response. Use the **source** *source-address* keyword and argument. For use with the MPLS LSP Ping and Traceroute features.

Option	Description
UDP destination address	A valid 127/8 address. You have the option to specify a single *x.y.z* or a range of numbers between 0.0.0 and *x.y.z* , where *x.y.z* are numbers between 0 and 255 and correspond to 127.*x.y.z*. Use the **destination** {*address* | *address-start address-end increment*} keyword and arguments. The MPLS echo request destination address in the UDP packet is not used to forward the MPLS packet to the destination device. The label stack that is used to forward the echo request routes the MPLS packet to the destination device. The 127/8 address guarantees that the packets are routed to the localhost (the default loopback address of the device processing the address) if the UDP packet destination address is used for forwarding. In addition, the destination address is used to affect load balancing when the destination address of the IP payload is used for load balancing. For use with IPv4 and Any Transport over MPLS (AToM) Forwarding Equivalence Classes (FECs) with the MPLS LSP Ping feature and with IPv4 FECs with the MPLS LSP Traceroute feature.
Time-to-live (TTL)	A parameter you can set that indicates the maximum number of hops a packet should take to reach its destination. The time-to-live (TTL) field in a packet is decremented by 1 each time it travels through a device. For MPLS LSP Ping, the TTL is a value after which the packet is discarded and an MPLS echo reply is sent back to the originating device. Use the **ttl** *time-to-live* keyword and argument. For MPLS LSP Traceroute, the TTL is a maximum time to live and is used to discover the number of downstream hops to the destination device. MPLS LSP Traceroute incrementally increases the TTL value in its MPLS echo requests (TTL = 1, 2, 3, 4, ...) to accomplish this. Use the **ttl** *time-to-live* keyword and argument.
Timeouts	A parameter you can specify to control the timeout in seconds for an MPLS request packet. The range is from 0 to 3600 seconds. The default is 2. Set with the **timeout** *seconds* keyword and argument. For use with the MPLS LSP Ping and Traceroute features.
Intervals	A parameter you can specify to set the time in milliseconds between successive MPLS echo requests. The default is 0. Set with the **interval** *msec* keyword and argument.
Experimental bits	Three experimental bits in an MPLS header used to specify precedence for the MPLS echo reply. (The bits are commonly called EXP bits.) The range is from 0 to 7, and the default is 0. Specify with the **exp** *exp-bits* keyword and argument. For use with the MPLS LSP Ping and Traceroute features.

Option	Description
Verbose	Option that provides additional information for the MPLS echo reply--source address and return codes. For the MPLS LSP Ping feature, this option is implemented with the **verbose** keyword. For use with the MPLS LSP Ping feature only.

MPLS LSP Ping options described in the table above can be implemented by using the following syntax:

```
ping mpls
{ipv4 destination-address destination-mask [destination address-start address-end increment]

 [ttl time-to-live] | pseudowire ipv4-address
 vc-id vc-id
[destination address-start address-end increment] | traffic-eng tunnel-interface
tunnel-number
[ttl time-to-live]}
[source source-address] [repeat count]
[{size packet-size} | {sweep minimum maximum size-Increment}]
[pad pattern]
[timeout seconds] [intervalmsec]
[exp exp-bits] [verbose
```

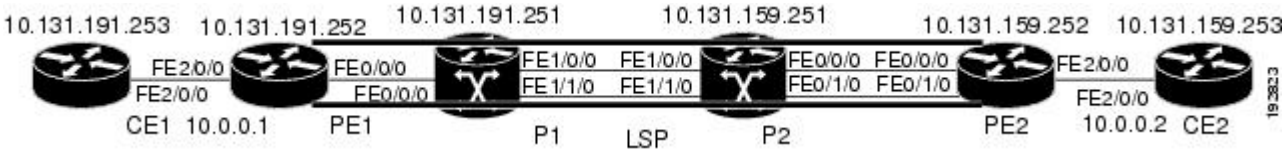
MPLS LSP Traceroute options described in the table below can be implemented by the use of the following syntax:

```
trace mpls
 {ipv4 destination-address destination-mask
 [destination address-start address-end address-increment] | traffic-eng tunnel-interface
tunnel-number}
[source source-address] [timeout seconds]
[ttl maximum-time-to-live]
[exp exp-bits]
```

Option Interactions and Loops

Usage examples for the MPLS LSP Ping and Traceroute and AToM VCCV feature in this and subsequent sections are based on the sample topology shown in the figure below.

Figure 22: Sample Topology for Configuration Examples



The interaction of some MPLS LSP Ping and Traceroute and AToM VCCV options can cause loops. See the following topic for a description of the loops you might encounter with the **ping mpls** and **trace mpls** commands:

Possible Loops with MPLS LSP Ping

With the MPLS LSP Ping feature, loops can occur if you use the repeat count option, the sweep size range option, or the User Datagram Protocol (UDP) destination address range option.

```
ping mpls
 {ipv4 destination-address/destination-mask
[destination address-start address-end increment] | pseudowire ipv4-address
 vc-id vc-id
 [destination address-start address-end increment] |
traffic-eng tunnel-interface tunnel-number}
[repeat count]
[sweep minimum maximum size-increment]
```

Following is an example of how a loop operates if you use the following keywords and arguments on the **ping mpls** command:

```
Device# ping mpls
 ipv4
 10.131.159.251/32 destination 127.0.0.1 127.0.0.1 0.0.0.1 repeat 2
sweep 1450 1475 25
Sending 2, [1450..1500]-byte MPLS Echos to 10.131.159.251/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
Destination address 127.0.0.1
!
!
Destination address 127.0.0.1
!
!
Destination address 127.0.0.1
!
!
Destination address 127.0.0.1
!
!
```

An **mpls ping** command is sent for each packet size range for each destination address until the end address is reached. For this example, the loop continues in the same manner until the destination address, 127.0.0.1, is reached. The sequence continues until the number is reached that you specified with the **repeat** *count* keyword and argument. For this example, the repeat count is 2. The MPLS LSP Ping loop sequence is as follows:

```
repeat  = 1
  destination address 1 (address-start
)
    for (size from sweep
 minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
  destination address 2 (address-start
 +
address-
increment
)
    for (size from sweep
 minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
```

```
  destination address 3 (address-start
 +
address-
increment
 +
address-
increment
)
     for (size from sweep
 minimum
 to maximum
, counting by size-increment
)
      send an lsp ping
.
.
.
  until destination address = address-end
.
.
.
until repeat = count
```

Possible Loop with MPLS LSP Traceroute

With the MPLS LSP Traceroute feature, loops can occur if you use the User Datagram Protocol (UDP) destination address range option and the time-to-live option.

Here is an example of how a loop operates if you use the following keywords and arguments on the **trace mpls** command:

```
Device# trace mpls
ipv4
 10.131.159.251/32 destination 127.0.0.1 127.0.0.1 1 ttl 5
Tracing MPLS Label Switched Path to 10.131.159.251/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
Destination address 127.0.0.1
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 40 ms
Destination address 127.0.0.2
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 40 ms
Destination address 127.0.0.3
  0 10.131.191.230 MRU 1500 [Labels: 19 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
! 2 10.131.159.225 48 ms
```

An **mpls trace** command is sent for each TTL from 1 to the maximum TTL (**ttl** *maximum-time-to-live* keyword and argument) for each destination address until the address specified with the destination *end-address* argument is reached. For this example, the maximum TTL is 5 and the end destination address is 127.0.0.1. The MPLS LSP Traceroute loop sequence is as follows:

```
destination address 1 (address-start
)
  for (ttl
 from 1 to maximum-time-to-live
```

```
)
    send an lsp trace
destination address 2 (address-start
 + address-increment
)
  for (ttl
 from 1 to maximum-time-to-live
)
    send an lsp trace
destination address 3 (address-start
 + address-increment
 + address-increment
)
  for (ttl
 from 1 to
maximum-time-to-live)
    send an lsp trace
.
.
.
until destination address = address-end
```

MPLS Echo Request Packets Not Forwarded by IP

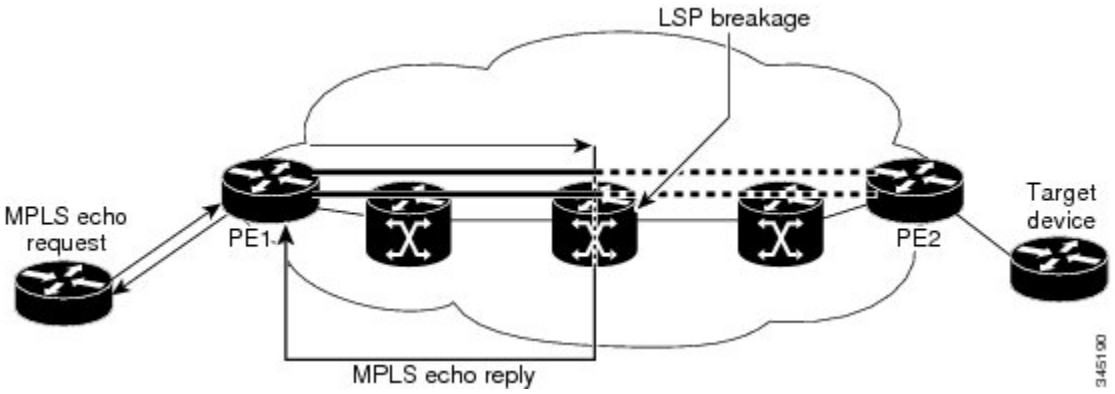
Multiprotocol Label Switching (MPLS) echo request packets sent during a label switched path (LSP) ping are never forwarded by IP. The IP header destination address field in an MPLS echo request packet is a 127.*x.y.z* /8 address. Devices should not forward packets using a 127.*x.y.z* /8 address. The 127.*x.y.z* /8 address corresponds to an address for the local host.

The use of a 127.*x .y .z* address as a destination address of the User Datagram Protocol (UDP) packet is significant in that the MPLS echo request packet fails to make it to the target device if a transit device does not label switch the LSP. This allows for the detection of LSP breakages.

- If an LSP breakage occurs at a transit device, the MPLS echo packet is not forwarded, but consumed by the device.
- If the LSP is intact, the MPLS echo packet reaches the target device and is processed by the terminal point of the LSP.

The figure below shows the path of the MPLS echo request and reply when a transit device fails to label switch a packet in an LSP.

Figure 23: Path When Transit Device Fails to Label Switch a Packet



Note An Any Transport over MPLS (AToM) payload does not contain usable forwarding information at a transit device because the payload might not be an IP packet. An MPLS virtual private network (VPN) packet, although an IP packet, does not contain usable forwarding information at a transit device because the destination IP address is only significant to the virtual routing and forwarding (VRF) instances at the endpoints of the MPLS network.

Information Provided by the Device Processing LSP Ping or LSP Traceroute

The table below describes the characters that the device processing an LSP ping or LSP traceroute packet returns to the sender about the failure or success of the request.

You can also view the return code for an MPLS LSP Ping operation if you enter the **ping mpls verbose** command.

Table 29: LSP Ping and Traceroute Reply Characters

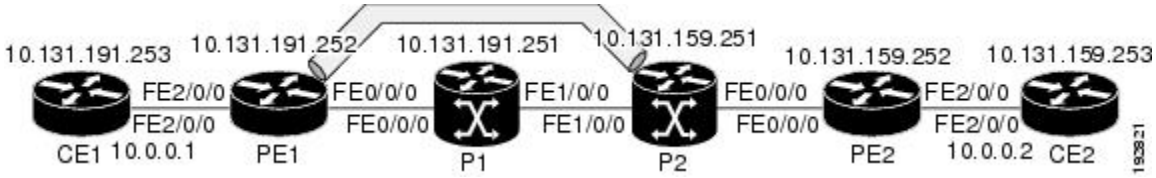
Character	Meaning
Period "."	A timeout occurs before the target device can reply.
U	The target device is unreachable.
R	The device processing the Multiprotocol Label Switching (MPLS) echo request is a downstream device but is not the destination.
Exclamation mark "!"	Replying device is an egress for the destination.
Q	Echo request was not successfully transmitted. This could be returned because of insufficient memory or more probably because no label switched path (LSP) exists that matches the Forwarding Equivalence Class (FEC) information.
C	Replying device rejected the echo request because it was malformed.

MTU Discovery in an LSP

During an MPLS LSP Ping, Multiprotocol Label Switching (MPLS) echo request packets are sent with the IP packet attribute set to do not fragment. That is, the DF bit is set in the IP header of the packet. This allows you to use the MPLS echo request to test for the MTU that can be supported for the packet through the label switched path (LSP) without fragmentation.

The figure below shows a sample network with a single LSP from PE1 to PE2 formed with labels advertised by means of LDP.

Figure 24: Sample Network with LSP—Labels Advertised by LDP



You can determine the maximum receive unit (MRU) at each hop by tracing the LSP using the MPLS Traceroute feature. The MRU is the maximum size of a labeled packet that can be forwarded through an LSP. The following example shows the results of a **trace mpls** command when the LSP is formed with labels created by the Label Distribution Protocol (LDP):

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1496 [Labels: 22/19 Exp: 0/0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 40 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
```

You can determine the MRU for the LSP at each hop through the use of the **show forwarding detail** command:

```
Device# show mpls forwarding 10.131.159.252 detail

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     19          10.131.159.252/32 0          Tu1        point2point
        MAC/Encaps=14/22, MRU=1496, Tag Stack{22 19}, via Et0/0
        AABBCC009700AABBCC0098008847 0001600000013000
        No output feature configured
```

To determine the maximum sized echo request that will fit on the LSP, you can find the IP MTU by using the **show interface** *type number* command.

```
Device# show interface e0/0

FastEthernet0/0/0 is up, line protocol is up
  Hardware is Lance, address is aabb.cc00.9800 (bia aabb.cc00.9800)
  Internet address is 10.131.191.230/30
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load ½55
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     377795 packets input, 33969220 bytes, 0 no buffer
     Received 231137 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     441772 packets output, 40401350 bytes, 0 underruns
```

```
     0 output errors, 0 collisions, 10 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The IP MTU in the **show interface** *type number* example is 1500 bytes. Subtract the number of bytes corresponding to the label stack from the MTU number. From the output of the **show mpls forwarding** command, the Tag stack consists of one label (21). Therefore, the largest MPLS echo request packet that can be sent in the LSP, shown in the figure above, is 1500 - (2 x 4) = 1492.

You can validate this by using the following **ping mpls** command:

```
Device# ping mpls ipv4 10.131.159.252/32 sweep 1492 1500 1 repeat 1
Sending 1, [1492..1500]-byte MPLS Echos to 10.131.159.252/32,
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
!QQQQQQQQ
Success rate is 11 percent (1/9), round-trip min/avg/max = 40/40/40 ms
```

In this command, only packets of 1492 bytes are sent successfully, as indicated by the exclamation point (!). Packets of byte sizes 1493 to 1500 are source-quenched, as indicated by the Q.

You can pad an MPLS echo request so that a payload of a given size can be tested. The pad TLV is useful when you use the MPLS echo request to discover the MTU supportable by an LSP. MTU discovery is extremely important for applications like AToM that contain non-IP payloads that cannot be fragmented.

LSP Network Management

To manage a Multiprotocol Label Switching (MPLS) network you must have the ability to monitor label switched paths (LSPs) and quickly isolate MPLS forwarding problems. You need ways to characterize the liveliness of an LSP and reliably detect when a label switched path fails to deliver user traffic.

You can use MPLS LSP Ping to verify the LSP that is used to transport packets destined for IPv4 Label Distribution Protocol (LDP) prefixes, traffic engineering (TE) tunnels, and Any Transport over MPLS pseudowire Forwarding Equivalence Classes (AToM PW FECs). You can use MPLS LSP Traceroute to trace LSPs that are used to carry packets destined for IPv4 LDP prefixes and TE tunnel FECs.

An MPLS echo request is sent through an LSP to validate it. A TTL expiration or LSP breakage causes the transit device to process the echo request before it gets to the intended destination and returns an MPLS echo reply that contains an explanatory reply code to the originator of the echo request.

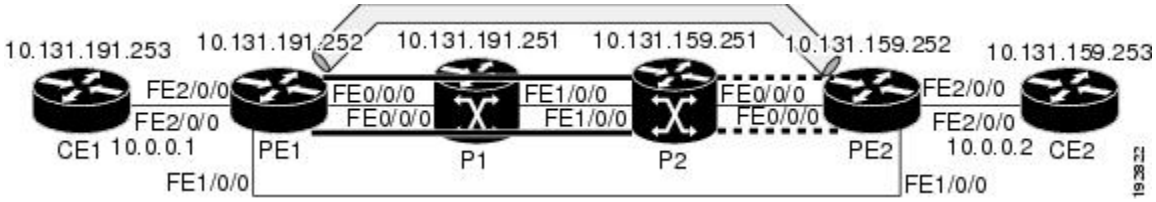
The successful echo request is processed at the egress of the LSP. The echo reply is sent via an IP path, an MPLS path, or a combination of both back to the originator of the echo request.

ICMP ping and trace Commands and Troubleshooting

Internet Control Message Protocol (ICMP) **ping** and **trace** commands are often used to help diagnose the root cause of a failure. When a label switched path (LSP) is broken, the packet might make its way to the target device by way of IP forwarding, thus making ICMP ping and traceroute unreliable for detecting Multiprotocol Label Switching (MPLS) forwarding problems. The MPLS LSP Ping, Traceroute and AToM VCCV feature extends this diagnostic and troubleshooting ability to the MPLS network and handles inconsistencies between the IP and MPLS forwarding tables, inconsistencies in the MPLS control and data plane, and problems with the reply path.

The figure below shows a sample topology with a Label Distribution Protocol (LDP) LSP and traffic engineering (TE) tunnel LSP.

Figure 25: Sample Topology with LDP and TE Tunnel LSPs



This section contains the following topics:

MPLS LSP Ping and Traceroute Discovers LSP Breakage

Configuration for Sample Topology

These are sample topology configurations for the troubleshooting examples in the following sections (see the figure above). There are the six sample device configurations.

Device CE1 Configuration

```
version 12.0
!
hostname ce1
!
enable password lab
!
interface Loopback0
 ip address 10.131.191.253 255.255.255.255
 no ip directed-broadcast
!
interface
 ip address 10.0.0.1 255.255.255.255
 no ip directed-broadcast
 no keepalive
 no cdp enable
!
end
```

Device PE1 Configuration

```
version 12.0
!
hostname pe1
!
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp discovery targeted-hello accept
!
interface Loopback0
 ip address 10.131.191.252 255.255.255.255
 no ip directed-broadcast
!
interface Tunnel1
```

```
 ip unnumbered Loopback0
 no ip directed-broadcast
 mpls label protocol ldp
 mpls ip
 tunnel destination 10.131.159.255
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth  512
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel2
 ip unnumbered Loopback0
 no ip directed-broadcast
 shutdown
 mpls label protocol ldp
 mpls ip
 tunnel destination 10.131.159.255
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  100
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface
 ip address 10.131.191.230 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface
 ip address 10.131.159.246 255.255.255.255
 no ip directed-broadcast
 no shutdown
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface
 no ip address
 no ip directed-broadcast
 no cdp enable
 xconnect 10.131.159.252 333 encapsulation mpls
!
interface
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.244 0.0.0.3 area 0
 network 10.131.191.228 0.0.0.3 area 0
 network 10.131.191.232 0.0.0.3 area 0
 network 10.131.191.252 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
ip classless

end
```

Device P1 Configuration

```
version 12.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname p1
!
enable password lab
!
ip cef
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp discovery targeted-hello accept
!
interface Loopback0
 ip address 10.131.191.251 255.255.255.255
 no ip directed-broadcast
!
interface
 ip address 10.131.191.229 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface
 ip address 10.131.159.226 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.224 0.0.0.3 area 0
 network 10.131.191.228 0.0.0.3 area 0
 network 10.131.191.251 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
end
```

Device P2 Configuration

```
version 12.0
hostname p2
!
ip cef
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp discovery directed-hello accept
!
!
```

```
interface Loopback0
 ip address 10.131.159.251 255.255.255.255
 no ip directed-broadcast
!
interface
 ip address 10.131.159.229 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface
 ip address 10.131.159.225 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
router ospf 1
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.159.224 0.0.0.3 area 0
 network 10.131.159.228 0.0.0.3 area 0
 network 10.131.159.251 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
end
```

Device PE2 Configuration

```
version 12.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pe2
!
logging snmp-authfail
enable password lab
!
clock timezone EST -5
ip subnet-zero
ip cef
no ip domain-lookup
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls ldp explicit-null
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
tag-switching tdp discovery directed-hello accept
frame-relay switching
!
!
interface Loopback0
 ip address 10.131.159.252 255.255.255.255
 no ip directed-broadcast
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
```

```
 tunnel destination 10.131.191.252
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng path-option 5 explicit name as1pe-long-path
!
interface
 ip address 10.131.159.230 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 tag-switching ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface
 ip address 10.131.159.245 255.255.255.255
 no ip directed-broadcast
 mpls traffic-eng tunnels
 tag-switching ip
 ip rsvp bandwidth 1500 1500
 ip rsvp signalling dscp 0
!
interface
 no ip address
 no ip directed-broadcast
 no cdp enable
 xconnect 10.131.191.252 333 encapsulation mpls
!
interface
 no ip address
 no ip directed-broadcast
!
interface
 no ip address
 no ip directed-broadcast
 shutdown
!
interface
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 passive-interface Loopback0
 network 10.131.122.0 0.0.0.3 area 0
 network 10.131.159.228 0.0.0.3 area 0
 network 10.131.159.232 0.0.0.3 area 0
 network 10.131.159.244 0.0.0.3 area 0
 network 10.131.159.252 0.0.0.0 area 0
!
ip classless
!
!
ip explicit-path name as1pe-long-path enable
 next-address 10.131.159.229
 next-address 10.131.159.226
 next-address 10.131.191.230
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
```

```
 exec-timeout 0 0
 password lab
 login
!
end
```

Device CE2 Configuration

```
version 12.0
!
hostname ce2
!
enable password lab
!
interface Loopback0
 ip address 10.131.159.253 255.255.255.255
 no ip directed-broadcast
!
interface
 ip address 10.0.0.2 255.255.255.255
 no ip directed-broadcast
 no keepalive
 no cdp enable
!
end
```

Verifying That the LSP Is Set Up Correctly

A **show mpls forwarding-table** command shows that tunnel 1 is in the Multiprotocol Label Switching (MPLS) forwarding table.

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     19
     [T] 10.131.159.252/32 0          Tu1
        point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

A **show mpls traffic-eng tunnels tunnel 1** command entered at PE1 displays information about tunnel 1 and verifies that it is forwarding packets with an out label of 22.

```
Device# show mpls traffic-eng tunnels tunnel 1

Name: PE1_t1                           (Tunnel1) Destination: 10.131.159.251
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type dynamic (Basis for Setup, path weight 20)
  Config Parameters:
    Bandwidth: 512      kbps (Global)  Priority: 2  2   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 512      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: dynamic path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet0/0/0, 22
  RSVP Signalling Info:
```

```
      Src 10.131.191.252, Dst 10.131.159.251, Tun_Id 1, Tun_Instance 28
    RSVP Path Info:
      My Address: 10.131.191.230
      Explicit Route: 10.131.191.229 10.131.159.226 10.131.159.225 10.131.159.251
      Record   Route:   NONE
      Tspec: ave rate=512 kbits, burst=1000 bytes, peak rate=512 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=512 kbits, burst=1000 bytes, peak rate=512 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 20 (TE)
    Explicit Route: 10.131.191.230 10.131.191.229 10.131.159.226 10.131.159.225
                    10.131.159.251
  History:
    Tunnel:
      Time since created: 9 days, 14 hours, 12 minutes
      Time since path change: 2 minutes, 18 seconds
    Current LSP:
      Uptime: 2 minutes, 18 seconds
    Prior LSP:
      ID: path option 1 [3]
      Removal Trigger: tunnel shutdown
```

A **trace mpls** command issued at PE1 verifies that packets with 22 as the outermost label and 19 as the end of stack label are forwarded from PE1 to PE2.

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1496 [Labels: 22/19
 Exp: 0/0]
R 1 10.131.159.226 MRU 1504 [Labels: 19 Exp: 0] 40 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
```

The MPLS LSP Traceroute to PE2 is successful, as indicated by the exclamation point (!).

Discovering LSP Breakage

A Label Distribution Protocol (LDP) target-session is established between devices PE1 and P2, as shown in the output of the following **show mpls ldp discovery** command:

```
Device# show mpls ldp discovery

 Local LDP Identifier:
    10.131.191.252:0
    Discovery Sources:
    Interfaces:
         (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
        Tunnel1 (ldp): Targeted -> 10.131.159.251
    Targeted Hellos:
        10.131.191.252 -> 10.131.159.252 (ldp): active/passive, xmit/recv
            LDP Id: 10.131.159.252:0
        10.131.191.252 -> 10.131.159.251 (ldp): active, xmit/recv
            LDP Id: 10.131.159.251:0
```

Enter the following command on the P2 device in global configuration mode:

```
Device# no mpls ldp discovery targeted-hello accept
```

The LDP configuration change causes the targeted LDP session between the headend and tailend of the traffic engineering (TE) tunnel to go down. Labels for IPv4 prefixes learned by P2 are not advertised to PE1. Thus, all IP prefixes reachable by P2 are reachable by PE1 only through IP (not MPLS). In other words, packets destined for those prefixes through Tunnel 1 at PE1 will be IP switched at P2 (which is undesirable).

The following **show mpls ldp discovery** command shows that the LDP targeted-session is down:

```
Device# show mpls ldp discovery

 Local LDP Identifier:
    10.131.191.252:0
    Discovery Sources:
    Interfaces:
        (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
        Tunnel1 (ldp): Targeted -> 10.131.159.251
    Targeted Hellos:
        10.131.191.252 -> 10.131.159.252 (ldp): active/passive, xmit/recv
            LDP Id: 10.131.159.252:0
        10.131.191.252 -> 10.131.159.251 (ldp): active, xmit
```

Enter the **show mpls forwarding-table** command at the PE1 device. The display shows that the outgoing packets are untagged as a result of the LDP configuration changes.

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
22     Untagged[T]
 10.131.159.252/32 0          Tu1        point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

A **ping mpls** command entered at the PE1 device displays the following:

```
Device# ping mpls ipv4 10.131.159.252/32 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
R
Success rate is 0 percent (0/1)
```

The **ping mpls** command fails. The R indicates that the sender of the Multiprotocol Label Switching (MPLS) echo reply had a routing entry but no MPLS Forwarding Equivalence Class (FEC) . Entering the **ping mpls verbose** command displays the MPLS label switched path (LSP) echo reply sender address and the return code. You should be able to solve the problem by Telneting to the replying device and inspecting its forwarding and label tables. You might need to look at the neighboring upstream device as well, because the breakage might be on the upstream device.

```
Device# ping mpls ipv4 10.131.159.252/32 repeat 1 verbose
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
```

```
       'R' - downstream router but not target
Type escape sequence to abort.
R   10.131.159.225, return code 6
Success rate is 0 percent (0/1)
```

Alternatively, use the LSP **traceroute** command to figure out which device caused the breakage. In the following example, for subsequent values of TTL greater than 2, the same device keeps responding (10.131.159.225). This suggests that the MPLS echo request keeps getting processed by the device regardless of the TTL. Inspection of the label stack shows that P1 pops the last label and forwards the packet to P2 as an IP packet. This explains why the packet keeps getting processed by P2. MPLS echo request packets cannot be forwarded by use of the destination address in the IP header because the address is set to a 127/8 address.

```
Device# trace mpls ipv4 10.131.159.252/32 ttl 5
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 22 Exp: 0]
R 1 10.131.159.226 MRU 1504 [implicit-null] 40 ms
R 2 10.131.159.225 40 ms
R 3 10.131.159.225 40 ms
R 4 10.131.159.225 40 ms
R 5 10.131.159.225 40 ms
```

MPLS LSP Traceroute Tracks Untagged Cases

This troubleshooting section contains examples of how to use MPLS LSP Traceroute to determine potential issues with packets that are tagged as implicit null and packets that are untagged.

Untagged output interfaces at a penultimate hop do not impact the forwarding of IP packets through a label switched path (LSP) because the forwarding decision is made at the penultimate hop through use of the incoming label. The untagged case causes Any Transport over Multiprotocol Label Switching (AToM) and MPLS virtual private network (VPN) traffic to be dropped at the penultimate hop.

Troubleshooting Implicit Null Cases

In the following example, Tunnel 1 is shut down, and only a label switched path (LSP) formed with Label Distribution Protocol (LDP) labels is established. An implicit null is advertised between the P2 and PE2 devices. Entering an MPLS LSP Traceroute at the PE1 device results in the following display:

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 20 Exp: 0]
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 80 ms
R 2 10.131.159.229 MRU 1504 [implicit-null] 28 ms
! 3 10.131.159.230 40 ms
```

This output shows that packets are forwarded from P2 to PE2 with an implicit-null label. Address 10.131.159.229 is configured for the P2 Fast Ethernet 0/0/0 out interface for the PE2 device.

Troubleshooting Untagged Cases

Untagged cases are valid configurations for Interior Gateway Protocol (IGP) label switched paths (LSPs) that could cause problems for Multiprotocol Label Switching (MPLS) virtual private networks (VPNs).

A **show mpls forwarding-table** command and a **show mpls ldp discovery** command issued at the P2 device show that the Label Distribution Protocol (LDP) is properly set up:

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
19     Pop tag     10.131.159.252/32 0          Et0/0      10.131.159.230
Device# show mpls ldp discovery
 Local LDP Identifier:
    10.131.159.251:0
    Discovery Sources:
    Interfaces:
           (ldp): xmit/recv
            LDP Id: 10.131.159.252:0
        FastEthernet1/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
```

The **show mpls ldp discovery** command output shows that, which connects PE2 to P2, is sending and receiving packets.

If a **no mpls ip** command is entered on , this could prevent an LDP session between the P2 and PE2 devices from being established. A **show mpls ldp discovery** command entered on the PE device shows that the MPLS LDP session with the PE2 device is down:

```
Device# show mpls ldp discovery

 Local LDP Identifier:
    10.131.159.251:0
    Discovery Sources:
    Interfaces:
        (ldp): xmit
        FastEthernet1/0/0 (ldp): xmit/recv
            LDP Id: 10.131.191.251:0
```

If the MPLS LDP session to PE2 goes down, the LSP to 10.131.159.252 becomes untagged, as shown by the **show mpls forwarding-table** command:

```
Device# show mpls forwarding-table 10.131.159.252

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
19     Untagged
    10.131.159.252/32 864        Et0/0      10.131.159.230
```

Untagged cases would provide an MPLS LSP Traceroute reply with packets tagged with No Label, as shown in the following display:

```
Device# trace mpls ipv4 10.131.159.252/32
Tracing MPLS Label Switched Path to 10.131.159.252/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
  0 10.131.191.230 MRU 1500 [Labels: 20 Exp: 0]
```

```
R 1 10.131.159.226 MRU 1500 [Labels: 19 Exp: 0] 80 ms
R 2 10.131.159.229 MRU 1504 [No Label] 28 ms
! 3 10.131.159.230 40 ms
```

MPLS LSP Ping and Traceroute Returns a Q

The Q return code always means that the packet could not be transmitted. The problem can be caused by insufficient memory, but it probably results because a label switched path (LSP) could not be found that matches the Forwarding Equivalence Class (FEC), information that was entered on the command line.

The reason that the packet was not forwarded needs to be determined. To do so, look at the Routing Information Base (RIB), the Forwarding Information Base (FIB), the Label Information Base (LIB), and the MPLS Label Forwarding Information Base (LFIB). Lack of an entry for the FEC in any one of these routing/forwarding bases would return a Q.

The table below lists commands that you can use for troubleshooting when the MPLS echo reply returns a Q.

Table 30: Troubleshooting a Q

Database	Command to View Contents
Routing Information Base	**show ip route**
Label Information Base and MPLS Forwarding Information Base	**show mpls forwarding-table detail**

The following example shows a **ping mpls** command where the MPLS echo request is not transmitted, as shown by the returned Qs:

```
Device# ping mpls ipv4 10.0.0.1/32
Sending 5, 100-byte MPLS Echos to 10.0.0.1/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
QQQQQ
Success rate is 0 percent (0/5)
```

A **show mpls forwarding-table** command and a **show ip route** command demonstrate that the address is not in either routing table:

```
Device# show mpls forwarding-table 10.0.0.1

Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
Device# show ip route 10.0.0.1

% Subnet not in table
```

The MPLS echo request is not transmitted because the IPv4 address (10.0.0.1) is not found in either the LFIB or the RIB routing table.

Load Balancing for IPv4 LDP LSPs

An Internet Control Message Protocol (ICMP) ping or trace follows one path from the originating device to the target device. Round robin load balancing of IP packets from a source device is used to discover the various output paths to the target IP address.

For MPLS LSP Ping and Traceroute, Cisco devices use the source and destination addresses in the IP header for load balancing when multiple paths exist through the network to a target device. The Cisco implementation of MPLS might check the destination address of an IP payload to accomplish load balancing (this checking depends on the platform).

To check for load balancing paths, you use the 127.*z.y.x* /8 destination address in the **ping mpls ipvr** *ip-address address-mask* **destination** *address-start address-end address-increment* command. The following examples show that different paths are followed to the same destination. This demonstrates that load balancing occurs between the originating device and the target device.

To ensure that the Fast Ethernet interface 1/0/0 on the PE1 device is operational, you enter the following commands on the PE1 device:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface fastethernet 1/0/0
Device(config-if)# no shutdown
Device(config-if)# end
*Dec 31 19:14:10.034: %LINK-3-UPDOWN: Interface FastEthernet1/0/0, changed state to up
*Dec 31 19:14:11.054: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0/0,
changed state to upend
PE1#
*Dec 31 19:14:12.574: %SYS-5-CONFIG_I: Configured from console by console
*Dec 31 19:14:19.334: %OSPF-5-ADJCHG: Process 1, Nbr 10.131.159.252 on FastEthernet1/0/0
from LOADING to FULL, Loading Done
PE1#
```

The following **show mpls forwarding-table** command displays the possible outgoing interfaces and next hops for the prefix 10.131.159.251/32:

```
Device# show mpls forwarding-table 10.131.159.251

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
21     19          10.131.159.251/32 0          FE0/0/0 10.131.191.229
       20          10.131.159.251/32 0          FE1/0/0 10.131.159.245
```

The following **ping mpls** command to 10.131.159.251/32 with a destination UDP address of 127.0.0.1 shows that the path selected has a path index of 0:

```
Device# ping mpls ipv4
 10.131.159.251/32 destination
 127.0.0.1 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.251/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
PE1#
*Dec 29 20:42:40.638: LSPV: Echo Request sent on IPV4 LSP, load_index 2,
pathindex 0
, size 100
*Dec 29 20:42:40.638: 46 00 00 64 00 00 40 00 FF 11 9D 03 0A 83 BF FC
*Dec 29 20:42:40.638: 7F 00 00 01 94 04 00 00 0D AF 0D AF 00 4C 14 70
*Dec 29 20:42:40.638: 00 01 00 00 01 02 00 00 1A 00 00 1C 00 00 00 01
*Dec 29 20:42:40.638: C3 9B 10 40 A3 6C 08 D4 00 00 00 00 00 00 00 00
*Dec 29 20:42:40.638: 00 01 00 09 00 01 00 05 0A 83 9F FB 20 00 03 00
```

```
*Dec 29 20:42:40.638: 13 01 AB CD AB CD AB CD AB CD AB CD AB CD
*Dec 29 20:42:40.638: AB CD AB CD
*Dec 29 20:42:40.678: LSPV: Echo packet received: src 10.131.159.225,
dst 10.131.191.252, size 74
*Dec 29 20:42:40.678: AA BB CC 00 98 01 AA BB CC 00 FC 01 08 00 45 C0
*Dec 29 20:42:40.678: 00 3C 32 D6 00 00 FD 11 15 37 0A 83 9F E1 0A 83
*Dec 29 20:42:40.678: BF FC 0D AF 0D AF 00 28 D1 85 00 01 00 00 02 02
*Dec 29 20:42:40.678: 03 00 1A 00 00 1C 00 00 00 01 C3 9B 10 40 A3 6C
*Dec 29 20:42:40.678: 08 D4 C3 9B 10 40 66 F5 C3 C8
```

The following **ping mpls** command to 10.131.159.251/32 with a destination UDP address of 127.0.0.1 shows that the path selected has a path index of 1:

```
Device# ping mpls ipv4 10.131.159.251/32 dest 127.0.0.1 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.251/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
*Dec 29 20:43:09.518: LSPV: Echo Request sent on IPV4 LSP, load_index 13,
pathindex 1
, size 100
*Dec 29 20:43:09.518: 46 00 00 64 00 00 40 00 FF 11 9D 01 0A 83 BF FC
*Dec 29 20:43:09.518: 7F 00 00 03 94 04 00 00 0D AF 0D AF 00 4C 88 58
*Dec 29 20:43:09.518: 00 01 00 00 01 02 00 00 38 00 00 1D 00 00 00 01
*Dec 29 20:43:09.518: C3 9B 10 5D 84 B3 95 84 00 00 00 00 00 00 00 00
*Dec 29 20:43:09.518: 00 01 00 09 00 01 00 05 0A 83 9F FB 20 00 03 00
*Dec 29 20:43:09.518: 13 01 AB CD AB CD AB CD AB CD AB CD AB CD
*Dec 29 20:43:09.518: AB CD AB CD
*Dec 29 20:43:09.558: LSPV: Echo packet received: src 10.131.159.229,
dst 10.131.191.252, size 74
*Dec 29 20:43:09.558: AA BB CC 00 98 01 AA BB CC 00 FC 01 08 00 45 C0
*Dec 29 20:43:09.558: 00 3C 32 E9 00 00 FD 11 15 20 0A 83 9F E5 0A 83
*Dec 29 20:43:09.558: BF FC 0D AF 0D AF 00 28 D7 57 00 01 00 00 02 02
*Dec 29 20:43:09.558: 03 00 38 00 00 1D 00 00 00 01 C3 9B 10 5D 84 B3
*Dec 29 20:43:09.558: 95 84 C3 9B 10 5D 48 3D 50 78
```

To see the actual path chosen, you use the **debug mpls lspv packet data** command.

Note The hashing algorithm is nondeterministic. Therefore, the selection of the *address-start* , *address-end* , and *address-increment* arguments for the **destination** keyword might not provide the expected results.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Related Topic	Document Title
Switching services commands	Cisco IOS IP Switching Command Reference
Concepts and tasks for configuring MPLS VPNs	*MPLS: Layer 3 VPNs Configuration Guide* (part of the *Multiprotocol Label Switching Configuration Guide Library*)

Standards and RFCs

Standards/RFCs	Title
draft-ietf-mpls-lsp-ping-03.txt	*Detecting MPLS Data Plane Failures*
draft-ietf-pwe3-vccv-01.txt	*Pseudo-Wire (PW) Virtual Circuit Connection Verification (VCCV)*
RFC 2113	*IP Router Alert Option*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LSP Ping, Traceroute, and AToM VCCV

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 31: Feature Information for MPLS LSP Ping, Traceroute, and AToM VCCV

Feature Name	Releases	Feature Information
MPLS LSP Ping, Traceroute, and AToM VCCV	12.0(27)S 12.2(28)SB 12.2(33)SXH Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.5S	You can use MPLS LSP Ping to test label switched path (LSP) connectivity for IPv4 Label Distribution Protocol (LDP) prefixes, traffic engineering (TE) Forwarding Equivalence Classes (FECs), and Any Transport over MPLS (AToM) FECs. You can use MPLS LSP Traceroute to trace the LSPs for IPv4 LDP prefixes and TE tunnel FECs. AToM VCCV allows you to use MPLS LSP Ping to test the pseudowire (PW) section of an AToM virtual circuit (VC). In Cisco IOS Release 12.2(28)SB, this feature was enhanced to support the Cisco 10000 series router. This feature was integrated into Cisco IOS Release 12.2(33)SXH and Cisco IOS XE Release 2.3. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **debug mpls lspv**, **ping mpls**, **trace mpls**.

Glossary

FEC—Forwarding Equivalence Class. A set of packets that can be handled equivalently for forwarding purposes and are thus suitable for binding to a single label. Examples include the set of packets destined for one address prefix and any flow.

flow—Generally, a set of packets traveling between a pair of hosts, or a pair of transport protocol ports on a pair of hosts. For example, packets with the same source address, source port, destination address, and destination port might be considered a flow.

A flow is also a stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

fragmentation—Process of breaking a packet into smaller units when transmitting over a network medium that cannot support the original size of the packet.

ICMP— Internet Control Message Protocol. A network layer Internet protocol that reports errors and provides other information relevant to IP packet processing. It is documented in RFC 792.

LFIB—label forwarding information base. A data structure and way of managing forwarding in which destinations and incoming labels are associated with outgoing interfaces and labels.

localhost—A name that represents the host name of a device. The localhost uses the reserved loopback IP address 127.0.0.1.

LSP—label switched path. A connection between two devices that uses MPLS to carry the packets.

LSPV—Label Switched Path Verification. An LSP Ping subprocess that encodes and decodes MPLS echo requests and replies; interfaces with IP, MPLS, and AToM switching for sending and receiving MPLS echo requests and replies; and, at the MPLS echo request originator device, maintains a database of outstanding echo requests for which echo responses have not been received.

MPLS router alert label—An MPLS label of 1. An MPLS packet with a router alert label is redirected by the device to the Route Processor (PR) processing level for handling. This allows these packets to bypass any forwarding failures in hardware routing tables.

MRU—maximum receive unit. Maximum size, in bytes, of a labeled packet that can be forwarded through an LSP.

MTU—maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

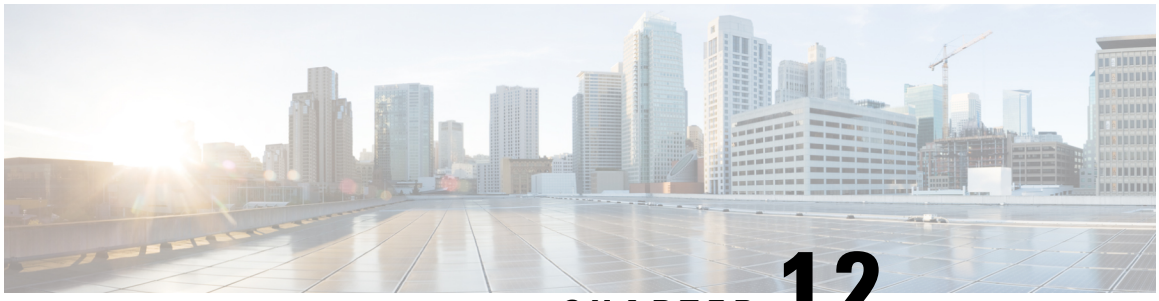
punt—Redirect packets with a router alert from the line card or interface to Route Processor (RP) level processing for handling.

PW—pseudowire. A mechanism that carries the essential elements of an emulated circuit from one provider edge (PE) device to another PE device over a packet-switched network.

RP—Route Processor. Processor module in the Cisco 7000 series routers that contains the CPU, system software, and most of the memory components that are used in the device. It is sometimes called a supervisory processor.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive. RSVP depends on IPv6. Is is also known as Resource Reservation Setup Protocol.

UDP—User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.



CHAPTER 12

MPLS EM - MPLS LSP Multipath Tree Trace

The MPLS EM—MPLS LSP Multipath Tree Trace feature provides the means to discover all possible equal-cost multipath (ECMP) routing paths of a label switched path (LSP) between an egress and ingress router. Once discovered, these paths can be retested on a periodic basis using Multiprotocol Label Switching (MPLS) LSP ping or traceroute. This feature is an extension to the MPLS LSP traceroute functionality for the tracing of IPv4 LSPs.

You can use the MPLS EM—MPLS LSP Multipath Tree Trace feature to discover all paths for an IPv4 LSP.

This implementation of the MPLS EM—MPLS LSP Multipath Tree Trace feature is based on RFC 4379, Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures .

For information on the use of MPLS LSP ping and traceroute, see the MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV feature module.

Cisco MPLS Embedded Management (EM) is a set of standards and value-added services that facilitate the deployment, operation, administration, and management of MPLS-based networks according to the fault, configuration, accounting, performance, and security (FCAPS) model.

Prerequisites for MPLS EMMPLS EM - MPLS LSP Multipath Tree Trace MPLS LSP Multipath Tree Trace

The following are prerequisites for using the MPLS EM—MPLS LSP Multipath Tree Trace feature:

- You must understand the concepts and know how to use MPLS LSP ping or traceroute as described in the MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV document.
- The routers in your network must be using an implementation based on RFC 4379, Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures .

- You should know the following about your MPLS network:
  - The topology
  - The number of links in your network
  - The expected number of LSPs, and how many LSPs
- Understand label switching, forwarding, and load balancing.

Restrictions for MPLS EM - MPLS LSP Multipath Tree Trace

- All restrictions that apply to the MPLS LSP Ping and LSP Traceroute features also apply to the MPLS EM—MPLS LSP Multipath Tree Trace feature:
  - You cannot use the MPLS LSP Multipath Tree Trace feature to trace the path taken by AToM packets. The MPLS LSP Multipath Tree Trace feature is not supported for AToM. (MPLS LSP Ping is supported for AToM.) However, you can use the MPLS LSP Multipath Tree Trace feature to troubleshoot the Interior Gateway Protocol (IGP) LSP that is used by AToM.
  - You cannot use the MPLS LSP Multipath Tree Trace feature to validate or trace MPLS Virtual Private Networks (VPNs). Multiple LSP paths are not discovered unless all routers in the MPLS core support an RFC 4379 implementation of Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures .
- MPLS LSP multipath tree trace is not expected to operate in networks that support time-to-live (TTL) hiding.

Information About MPLS EM - MPLS LSP Multipath Tree Trace

Overview of MPLS LSP Multipath Tree Trace

As the number of MPLS deployments increases, the number of traffic types the MPLS networks carry could increase. In addition, load balancing on label switch routers (LSRs) in the MPLS network provides alternate paths for carrying MPLS traffic to a target router. The ability of service providers to monitor LSPs and quickly isolate MPLS forwarding problems is critical to their ability to offer services.

Prior to the release of the MPLS EM—MPLS LSP Multipath Tree Trace feature no automated way existed to discover all paths between provider edge (PE) routers. Troubleshooting forwarding problems between PEs was cumbersome.

The release of the MPLS EM—MPLS LSP Multipath Tree Trace feature provides an automated way to discover all paths from the ingress PE router to the egress PE router in multivendor networks that use IPv4 load balancing at the transit routers. Once the PE-to-PE paths are discovered, use MPLS LSP ping and MPLS LSP traceroute to periodically test them.

The MPLS EM—MPLS LSP Multipath Tree Trace feature requires the Cisco RFC-compliant implementation that is based on RFC 4379. If you do not have a Cisco software release that supports RFC 379, MPLS LSP multipath tree trace does not operate to discover all PE-to-PE paths.

Discovery of IPv4 Load Balancing Paths by MPLS LSP Multipath Tree Trace

IPv4 load balancing at a transit router is based on the incoming label stack and the source and destination addresses in the IP header. The outgoing label stack and IP header source address remain constant for each branch being traced.

When you execute MPLS LSP multipath tree trace on the source LSR, the router needs to find the set of IP header destination addresses to use all possible output paths. The source LSR starts path discovery by sending a transit router a bitmap in an MPLS echo request. The transit router returns information in an MPLS echo request that contains subsets of the bitmap in a downstream map (DS Map) in an echo reply. The source router can then use the information in the echo reply to interrogate the next router. The source router interrogates each successive router until it finds one bitmap setting that is common to all routers along the path. The router uses TTL expiry to interrogate the routers to find the common bits.

For example, you could start path discovery by entering the following command at the source router:

```
Router# trace mpls multipath ipv4 10.131.101.129/32 hashkey ipv4 bitmap 16
```

This command sets the IP address of the target router as 10.131.101.192 255.255.255.255 and configures:

- The default hash key type to 8, which requests that an IPv4 address prefix and bit mask address set be returned in the DS Map in the echo reply.
- The bitmap size to 16. This means that MPLS LSP multipath tree trace uses 16 addresses (starting with 127.0.0.1) in the discovery of all paths of an LSP between the source router and the target router.

If you enter the **trace mpls multipath ipv4 10.131.101.129/32** command, MPLS LSP multipath tree trace uses the default hash type of 8 or IPv4 and a default bitmap size of 32. Your choice of a bitmap size depends on the number of routes in your network. If you have a large number of routes, you might need to use a larger bitmap size.

Echo Reply Return Codes Sent by the Router Processing Multipath LSP Tree Trace

The table below describes the characters that the router processing a multipath LSP tree trace packet returns to the sender about the failure or success of the request.

Table 32: Echo Reply Return Codes

Output Code	Echo Return Code	Meaning
Period "."	—	A timeout occurred before the target router could reply.
x	0	No return code.
M	1	Malformed request.
m	2	Unsupported type, length, values (TLVs).
!	3	Success.
F	4	No Forwarding Equivalence Class (FEC) mapping.
D	5	DS Map mismatch.

Output Code	Echo Return Code	Meaning
R	6	Downstream router but not target.
U	7	Reserved.
L	8	Labeled output interface.
B	9	Unlabeled output interface.
f	10	FEC mismatch.
N	11	No label entry.
P	12	No receive interface label protocol.
p	13	Premature termination of the LSP.
X	unknown	Undefined return code.

MPLS Embedded Management Configuration

Before using the **ping mpls**, **trace mpls**, or **trace mpls multipath** command, you should consider ensuring that the router is configured to encode and decode MPLS echo packets in a format that all receiving routers in the network can understand.

LSP ping drafts after Version 3 (draft-ietf-mpls-ping-03) have undergone numerous TLV format changes, but the implementations based on different drafts might not interoperate properly.

To allow later Cisco implementations to interoperate with draft Version 3 Cisco and non-Cisco implementations, a global configuration mode (MPLS OAM configuration) allows you to encode and decode echo packets in formats specified by draft Version 3 implementations.

Unless configured otherwise, a Cisco implementation encodes and decodes echo requests assuming the version on which the Internet Engineering Task Force (IETF) implementation is based.

To allow for seamless interoperability with earlier Revision 1 and 3 images, you can use MPLS Operation, Administration, and Maintenance (OAM) configuration mode parameters to force the default behavior of the Revision 4 images to be compliant or compatible in networks with Revision 1 or Revision 3 images.

To prevent failures reported by the replying router due to TLV version issues, you should configure all routers in the core. Encode and decode MPLS echo packets in the same draft version. For example, if the network is running RFC 4379 (Cisco Revision 4) implementations but one router is capable of only Version 3 (Cisco Revision 3), configure all routers in the network to operate in Revision 3 mode.

Cisco Revision 4 is the default version. The default version is the latest LSP Ping version supported by the image on the router.

How to Configure MPLS EM - MPLS LSP Multipath Tree Trace

Customizing the Default Behavior of MPLS Echo Packets

Perform the following task to customize the default behavior of MPLS echo packets. You might need to customize the default echo packet encoding and decoding behavior to allow later implementations of the Detecting MPLS Data Plane Failures (RFC 4379) to be deployed in networks running earlier versions of the draft.

Before you begin

MPLS LSP Multipath Tree Trace requires RFC 4379 (Revision 4).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls oam**
4. **echo revision** {**3** | **4**}
5. [**no**] **echo vendor-extension**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls oam** **Example:** `Router(config)# mpls oam`	Enters MPLS OAM configuration mode and customizes the default behavior of echo packets.
Step 4	**echo revision** {**3** | **4**} **Example:** `Router(config-mpls)# echo revision 4`	Customizes the default behavior of echo packets. • The **revision** keyword set echo packet attributes to one of the following: • **3** = draft-ietf-mpls-ping-03 (Revision 2) • **4** = RFC 4379 compliant (default)

	Command or Action	Purpose
		Note The MPLS LSP Multipath Tree Trace feature requires Revision 4.
Step 5	[**no**] **echo vendor-extension** **Example:** `Router(config-mpls)# echo vendor-extension`	Customizes the default behavior of echo packets. • The **vendor-extension** keyword sends the Cisco-specific extension of TLVs with the echo packets. • The **no** form of the command allows you to disable a Cisco vendor's extension TLVs that another vendor's noncompliant implementations may not support. The router default is **echo vendor-extension**.
Step 6	**end** **Example:** `Router(config-mpls)# end`	Exits to privileged EXEC mode.

Configuring MPLS LSP Multipath Tree Trace

Perform the following task to configure MPLS multipath LSP traceroute. This task helps discover all LSPs from an egress router to an ingress router.

Before you begin

Cisco LSP ping or traceroute implementations based on draft-ietf-mpls-lsp-ping-11 are capable in some cases of detecting the formatting of the sender of an MPLS echo request. However, certain cases exist in which an echo request or echo reply might not contain the Cisco extension TLV. To avoid complications due to certain cases where the echo packets are decoded assuming the wrong TLV formats, configure all routers in the network to operate in the same mode.

For an MPLS LSP multipath tree trace to be successful, the implementation in your routers must support RFC 4379 on all core routers.

If all routers in the network support RFC-4379 and another vendor's implementation exists that is not capable of properly handling Cisco's vendor TLV, the routers supporting the RFC-compliant or later configuration must include commands to disable the Cisco vendor TLV extensions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls oam**
4. **echo revision 4**
5. [**no**] **echo vendor-extension**
6. **end**
7. **trace mpls multipath ipv4** *destination-ip-address/destination mask-length*

8. **debug mpls lspv multipath**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls oam** **Example:** `Router(config)# mpls oam`	Enters MPLS OAM configuration mode.
Step 4	**echo revision 4** **Example:** `Router(config-mpls)# echo revision 4`	Customizes the default behavior of echo packets. • The **revision 4** keywords set echo packet attributes to the default Revision 4 (RFC 4379 compliant). **Note** The MPLS LSP Multipath Tree Trace feature requires Revision 4.
Step 5	[**no**] **echo vendor-extension** **Example:** `Router(config-mpls) echo vendor-extension`	(Optional) Customizes the default behavior of echo packets. • The **vendor-extension** keyword sends the Cisco-specific extension of TLVs with the echo packets. • The **no** form of the command allows you to disable a Cisco vendor's extension TLVs that another vendor's noncompliant implementations may not support. The router default is **echo vendor-extension**.
Step 6	**end** **Example:** `Router(config-mpls)# end`	Exits to privileged EXEC mode.
Step 7	**trace mpls multipath ipv4** *destination-ip-address/destination mask-length* **Example:** `Router# trace mpls multipath ipv4 10.131.161.251/32`	Discovers all LSPs from an egress router to an ingress router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address.

	Command or Action	Purpose
		• The *destination-ip-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required.
Step 8	**debug mpls lspv multipath** **Example:** `Router# debug mpls lspv multipath`	Displays multipath information related to the MPLS LSP Multipath Tree Trace feature.

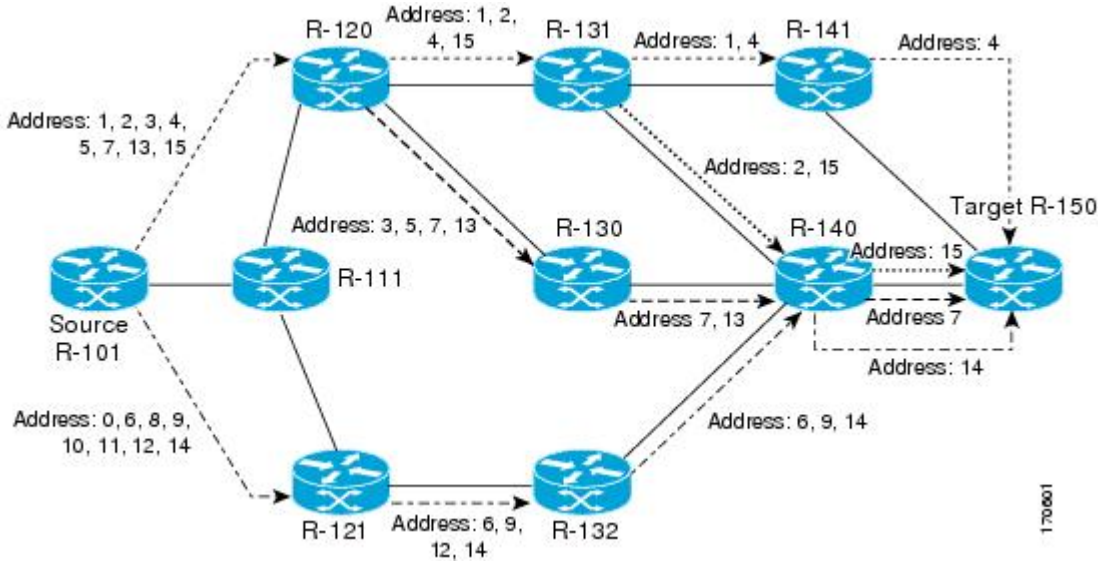
Discovering IPv4 Load Balancing Paths Using MPLS LSP Multipath Tree Trace

Perform the following task to discover IPv4 load balancing paths using MPLS LSP Multipath Tree Trace.

A Cisco router load balances MPLS packets based on the incoming label stack and the source and destination addresses in the IP header. The outgoing label stack and IP header source address remain constant for each path being traced. The router needs to find the set of IP header destination addresses to use all possible output paths. This might require exhaustive searching of the 127.x.y.z/8 address space. Once you discover all paths from the source LSR to the target or destination LSR with MPLS LSP Multipath Tree Trace, you can use MPLS LSP traceroute to monitor these paths.

The figure below shows how MPLS LSP Multipath Tree Trace discovers LSP paths in a sample network. In the figure below, the bitmap size is 16 and the numbers 0 to 15 represent the bitmapped addresses that MPLS LSP Multipath Tree Trace uses to discover all the paths from the source LSR R-101 to the target LSR R-150. The figure below illustrates how the **trace mpls multipath** command discovers all LSP paths in the sample network.

Figure 26: MPLS LSP Multipath Tree Trace Path Discovery in a Sample Network



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls oam**
4. **echo revision 4**
5. **end**
6. **trace mpls multipath ipv4** *destination-address*/*destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls oam** **Example:** `Router(config)# mpls oam`	Enters MPLS OAM configuration mode and sets the echo packet attribute to Revision 4 (RFC 4379 compliant).
Step 4	**echo revision 4** **Example:** `Router(config-mpls)# echo revision 4`	Customizes the default behavior of echo packets. • The **revision 4** keywords set echo packet attributes to the default Revision 4 (RFC 4379 compliant). **Note** The MPLS LSP Multipath Tree Trace feature requires Revision 4.
Step 5	**end** **Example:** `Router(config-mpls)# end`	Exits to privileged EXEC mode.
Step 6	**trace mpls multipath ipv4** *destination-address*/*destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size* **Example:** `Router# trace mpls multipath ipv4 10.131.161.251/32 hashkey ipv4 bitmap 16`	Discovers all MPLS LSPs from an egress router to an ingress router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested.

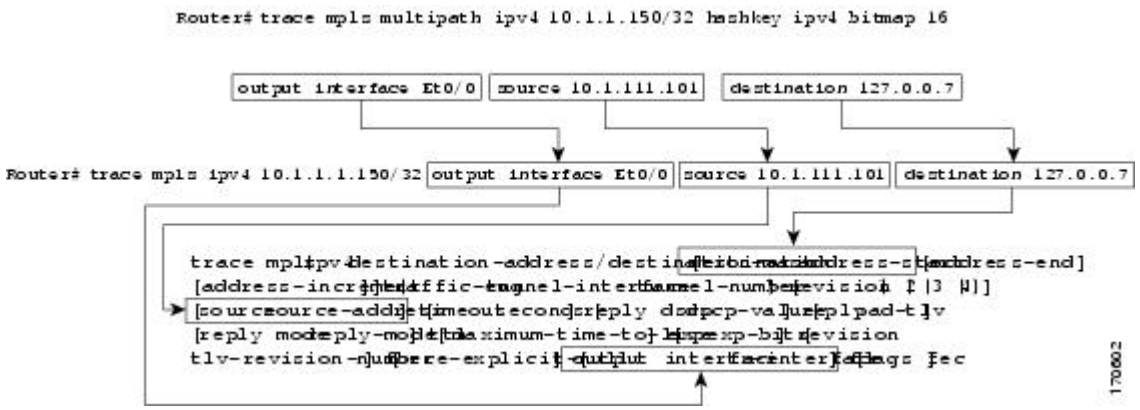
	Command or Action	Purpose
		• The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required. • The **hashkey ipv4** keywords set the hashkey type to IPv4 addresses. • The **bitmap** *bitmap-size* keyword and arguments set the bitmap size for multipath discovery.

Monitoring LSP Paths Discovered by MPLS LSP Multipath Tree Trace Using MPLS LSP Traceroute

Perform the following task to monitor LSP paths discovered by MPLS LSP Multipath Tree Trace using MPLS LSP traceroute. You can take output directly from the **trace mpls multipath** command and add it to a **trace mpls** command periodically to verify that the path is still operating.

The figure below shows the mapping of the output of a **trace mpls multipath** command to a **trace mpls** command.

Figure 27: Mapping of trace mpls multipath Command Output to a trace mpls Command



Each path you discover with MPLS LSP Multipath Tree Trace can be tested in this manner periodically to monitor the LSP paths in your network.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address*/*destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size*
3. **trace mpls ipv4** *destination-address*/*destination-mask-length* [**output interface** *tx-interface*] [**source** *source-address*] [**destination** *address-start*
4. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **trace mpls multipath ipv4** *destination-address*/*destination-mask-length* **hashkey ipv4 bitmap** *bitmap-size*

Use this command to discover all MPLS LSPs from an egress router to an ingress router. For example:

Example:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 hashkey ipv4 bitmap 16

Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 468 ms
```

The output of the **trace mpls multipath ipv4** command in the example shows the result of path discovery with MPLS LSP Multipath Tree Trace. In this example, the command sets the bitmap size to 16. Path discovery starts by MPLS LSP Multipath Tree Trace using 16 bitmapped addresses as it locates LSP paths from the source router to the target router with prefix and mask 10.1.1.150/32. MPLS LSP Multipath Tree Trace starts using the 127.x.y.z/8 address space with 127.0.0.1.

Step 3 **trace mpls ipv4** *destination-address*/*destination-mask-length* [**output interface** *tx-interface*] [**source** *source-address*] [**destination** *address-start*

Use this command to verify that the paths discovered when you entered a **trace mpls multipath ipv4** command are still operating. For example, the output for Path 0 in the previous **trace mpls multipath ipv4** command in Step 2 is:

Example:

```
output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
```

If you put the output for path 0 in the **trace mpls** command, you see the following results:

Example:

```
Router# trace mpls ipv4 10.1.1.150/32 output interface Fe0/0/0 source 10.1.111.101 destination
127.0.0.0

Tracing MPLS Label Switched Path to 10.1.1.150/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.1.111.101 MRU 1500 [Labels: 33 Exp: 0]
L 1 10.1.111.111 MRU 1500 [Labels: 34 Exp: 0] 40 ms
L 2 10.2.121.121 MRU 1500 [Labels: 34 Exp: 0] 32 ms
L 3 10.3.132.132 MRU 1500 [Labels: 32 Exp: 0] 16 ms
L 4 10.4.140.240 MRU 1504 [Labels: implicit-null Exp: 0] 20 ms
! 5 10.5.150.50 20 ms
```

You can take output directly from the**trace mpls multipath** command and add it to a **trace mpls** command periodically to verify that the path is still operating (see the figure above).

Step 4 **exit**

Use this command to exit to user EXEC mode. for example:

Example:

```
Router# exit
Router>
```

Using DSCP to Request a Specific Class of Service in an Echo Reply

A reply differentiated services code point (DSCP) option lets you request a specific class of service (CoS) in an echo reply.

The reply DSCP option is supported in the experimental mode for IETF draft-ietf-mpls-lsp-ping-03.txt. Cisco implemented a vendor-specific extension for the reply DSCP option rather than using a Reply TOS TLV. A Reply TOS TLV serves the same purpose as the **reply dscp** command in IETF draft-ietf-mpls-lsp-ping-11.txt. This draft provides a standardized method of controlling the reply DSCP.

Note Before RFC 4379, Cisco implemented the Reply DSCP option as an experimental capability using a Cisco vendor extension TLV. If a router is configured to encode MPLS echo packets for draft Version 3 implementations, a Cisco vendor extension TLV is used instead of the = Reply TOS TLV that was defined in draft Version 8.

To use DSCP to request a specific CoS in an echo reply, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**reply dscp** *dscp-value*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* [**reply dscp** *dscp-value*] **Example:** `Router# trace mpls multipath ipv4 10.131.191.252/32 reply dscp 50`	Discovers all MPLS LSPs from an ingress router to an egress router and controls the DSCP value of an echo reply. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required. • The **reply dscp** *dscp-value* keywords and argument are the DSCP value of an echo reply. A Reply TOS TLV serves the same purpose as the **reply dscp** command in IETF draft-ietf-mpls-lsp-ping-11.txt. **Note** To specify a DSCP value, you must enter the **reply dscp** *dscp-value* keywords and argument.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Controlling How a Responding Router Replies to an MPLS Echo Request

This section contains information about and instructions for controlling how a responding router replies to an MPLS echo request. You should understand the following information before you configure a reply mode for the echo request response:

Reply Modes for an MPLS LSP Multipath Tree Trace Echo Request Response

The reply mode controls how a responding router replies to an MPLS echo request sent by a **trace mpls multipath** command. There are two reply modes for an echo request packet:

- ipv4—Reply with an IPv4 User Datagram Protocol (UDP) packet (default)
- router-alert—Reply with an IPv4 UDP packet with router alert

Note Use the ipv4 and router-alert reply modes with each other to prevent false negatives. If you do not receive a reply via the ipv4 mode, send a test with the router-alert reply mode. If both fail, something is wrong in the return path. The problem might be due to an incorrect ToS setting.

IPv4 UDP Reply Mode: The IPv4 UDP reply mode is the most common reply mode used with a **trace mpls multipath** command when you want to periodically poll the integrity of an LSP. With this option, you do not have explicit control over whether the packet traverses IP or MPLS hops to reach the originator of the MPLS echo request. If the originating (headend) router fails to receive a reply to an MPLS echo request when you use the **reply mode ipv4** keywords, use the **reply mode router-alert** keywords.

Router-aler Reply Mode: The router-alert reply mode adds the router alert option to the IP header. When an IP packet that contains an IP router alert option in its IP header or an MPLS packet with a router alert label as its outermost label arrives at a router, the router punts (redirects) the packet to the Route Processor (RP) process level for handling. This forces the RP of each intermediate router to specifically handle the packet at each intermediate hop as it moves back to the destination. Hardware and line-card forwarding inconsistencies are thus bypassed. Router-alert reply mode is slower than IPv4 mode because the reply requires process-level RP handling at each hop.

The table below describes how an incoming IP packet with an IP router alert is handled by the router switching path processes when the outgoing packet is an IP packet or an MPLS packet. It also describes how an MPLS packet with a router alert option is handled by the router switching path processes when the outgoing packet is an IP packet or an MPLS packet.

Table 33: Path Process Handling of IP and MPLS Router Alert Packets

Incoming Packet	Outgoing Packet	Normal Switching Action	Process Switching Action
IP packet—Router alert option in IP header	IP packet—Router alert option in IP header	Router alert option in IP header causes the packet to be punted to the process switching path.	Forwards the packet as is
	MPLS packet		Forwards the packet as is
MPLS packet—Outermost label contains a router alert	IP packet—Router alert option in IP header	If the router alert label is the outermost label, it causes the packet to be punted to the process switching path.	Removes the outermost router alert label and forwards the packet as an IP packet
	MPLS packet— Outermost label contains a router alert		Preserves the outermost router alert label and forwards the MPLS packet

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* **reply mode** {**ipv4** | **router-alert**}
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**trace mpls multipath ipv4** *destination-address*/*destination-mask-length* **reply mode** {**ipv4** | **router-alert**}
	Example:
	`Router# trace mpls multipath ipv4 10.131.191.252/32 reply mode router-alert`
		Discovers all MPLS LSPs from an ingress router to an egress router and specifies the reply mode.
		• The **ipv4** keyword specifies the destination type as an LDP IPv4 address.
		• The *destination-address* argument is the address prefix of the target to be tested.
		• The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required.
		• The **reply mode** keyword requires that you enter one of the following keywords to specify the reply mode:
		  • The **ipv4** keyword—Reply with an IPv4 UDP packet (default).
		  • The **router-alert** keyword—Reply with an IPv4 UDP packet with router alert.
		Note To specify the reply mode, you must enter the **reply mode** keyword with the **ipv4** or **router-alert** keyword.

Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Specifying the Output Interface for Echo Packets Leaving a Router for MPLS LSP Multipath Tree Trace

Perform the following task to specify the output interface for echo packets leaving a router for the MPLS LSP Multipath Tree Trace feature. You can use this task to test the LSPs reachable through a given interface.

Echo Request Output Interface Control: You can control the interface through which packets leave a router. Path output information is used as input to LSP ping and traceroute.

The echo request output interface control feature allows you to force echo packets through the paths that perform detailed debugging or characterizing of the LSP. This feature is useful if a PE router connects to an MPLS cloud and there are broken links. You can direct traffic through a certain link. The feature also is helpful for troubleshooting network problems.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**output interface** *tx-interface*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* [**output interface** *tx-interface*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.251/32 output interface fastethernet0/0/0`	Discovers all MPLS LSPs from an ingress router to an egress router and specifies the interface through which echo packets leave a router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required. • The **output interface** *tx-interface* keywords and argument specify the output interface for the MPLS echo request. **Note** You must specify the **output interface** keywords.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Setting the Pace of MPLS Echo Request Packet Transmission for MPLS LSP Multipath Tree Trace

Perform the following task to set the pace of MPLS echo request packet transmission for the MPLS LSP Multipath Tree Trace feature. Echo request traffic pacing allows you to set the pace of the transmission of packets so that the receiving router does not drop packets. If you have a large amount of traffic on your network you might increase the size of the interval to help ensure that the receiving router does not drop packets.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**interval** *milliseconds*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* [**interval** *milliseconds*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.251/32 interval 100`	Discovers all MPLS LSPs from an egress router to an ingress router and sets the time in milliseconds between successive MPLS echo requests. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required. • The **interval** *milliseconds* keyword and argument set the time between successive MPLS echo requests in milliseconds. The default is 0 milliseconds. **Note** To pace the transmission of packets, you must specify the **interval** keyword.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Enabling MPLS LSP Multipath Tree Trace to Detect LSP Breakages

Perform the following task to enable MPLS LSP Multipath Tree Trace to detect LSP breakages caused by an interface that lacks an MPLS configuration. If an interface is not configured for MPLS, then it cannot forward MPLS packets.

Explicit Null Label Shimming Tests LSP Ability to Carry MPLS Traffic: For an MPLS LSP Multipath Tree Trace of LSPs carrying IPv4 FECs, you can force an explicit null label to be added to the MPLS label stack even though the label was unsolicited. This allows MPLS LSP Multipath Tree Trace to detect LSP breakages caused by an interface that is not configured for MPLS. MPLS LSP Multipath Tree Trace does not report that an LSP is functioning when it is unable to send MPLS traffic.

An explicit null label is added to an MPLS label stack if MPLS echo request packets are forwarded from an interface not configured for MPLS that is directly connected to the destination of the MPLS LSP Multipath Tree Trace or if the IP TTL value for the MPLS echo request packets is set to 1.

When you enter a **trace mpls multipath** command, you are looking for all MPLS LSP paths from an egress router to an ingress router. Failure at output interfaces that are not configured for MPLS at the penultimate hop are not detected. Explicit-null shimming allows you to test an LSP's ability to carry MPLS traffic.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* **force-explicit-null**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* **force-explicit-null** **Example:** `Router# trace mpls multipath ipv4 10.131.191.252/32 force-explicit-null`	Discovers all MPLS LSPs from an egress router to an ingress router and forces an explicit null label to be added to the MPLS label stack. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required. • The **force-explicit-null** keyword forces an explicit null label to be added to the MPLS label stack even though the label was unsolicited. **Note** You must enter the **force-explicit-null** keyword to enable MPLS LSP Multipath Tree Trace to detect LSP breakages caused by an interface that is not configured for MPLS.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Requesting That a Transit Router Validate the Target FEC Stack for MPLS LSP Multipath Tree Trace

Perform the following task to request that a transit router validate the target FEC stack for the MPLS LSP Multipath Tree Trace feature.

An MPLS echo request tests a particular LSP. The LSP to be tested is identified by the FEC stack.

During an MPLS LSP Multipath Tree Trace, the echo packet validation rules do not require that a transit router validate the target FEC stack TLV. A downstream map TLV containing the correct received labels must be present in the echo request for target FEC stack checking to be performed.

To request that a transit router validate the target FEC stack, set the V flag from the source router by entering the **flags fec** keywords in the **trace mpls multipath** command. The default is that echo request packets are sent with the V flag set to 0.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address/destination-mask-length* [**flags fec**] [**ttl** *maximum-time-to-live*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address/destination-mask-length* [**flags fec**] [**ttl** *maximum-time-to-live*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.252/32 flags fec ttl 5`	Discovers all MPLS LSPs from an egress router to an ingress router and requests validation of the target FEC stack by a transit router. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required. • The **flags fec** keywords requests that target FEC stack validation be done at a transit router. • The **ttl** *maximum-time-to-live* keyword and argument pair specify a maximum hop count.

	Command or Action	Purpose
		Note For a transit router to validate the target FEC stack, you must enter the **flags fec** and **ttl** keywords.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Setting the Number of Timeout Attempts for MPLS LSP Multipath Tree Trace

Perform the following task to set the number of timeout attempts for the MPLS LSP Multipath Tree Trace feature.

A retry is attempted if an outstanding echo request times out waiting for the corresponding echo reply.

SUMMARY STEPS

1. **enable**
2. **trace mpls multipath ipv4** *destination-address*/*destination-mask-length* [**retry-count** *retry-count-value*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**trace mpls multipath ipv4** *destination-address*/*destination-mask-length* [**retry-count** *retry-count-value*] **Example:** `Router# trace mpls multipath ipv4 10.131.159.252/32 retry-count 4`	Sets the number of retry attempts during an MPLS LSP Multipath Tree Trace. • The **ipv4** keyword specifies the destination type as an LDP IPv4 address. • The *destination-address* argument is the address prefix of the target to be tested. • The *destination-mask-length* argument is the number of bits in the network mask of the target address. The / keyword before this argument is required. • The **retry-count** *retry-count-value* keyword and argument sets the number of retry attempts after a timeout occurs. A retry-count value of "0" means infinite retries. A retry-count value from 0 to10 is suggested. You might want

	Command or Action	Purpose
		to increase the retry value to greater than 10, if 10 is too small a value. The default retry-count value is 3. **Note** To set the number of retries after a timeout, you must enter the **retry-count** keyword.
Step 3	**exit** **Example:** `Router# exit`	Returns to user EXEC mode.

Configuration Examples for MPLS EM - MPLS LSP Multipath Tree Trace

Customizing the Default Behavior of MPLS Echo Packets Example

The following example shows how to customize the behavior of MPLS echo packets so that the MPLS LSP Multipath Tree Trace feature interoperates with a vendor implementation that does not interpret RFC 4379 as Cisco does:

```
configure terminal
!
mpls oam
 echo revision 4
 no echo vendor-extension
 end
```

The **echo revision** command is included in this example for completeness. The default echo revision number is 4, which corresponds to RFC 4379.

Configuring MPLS LSP Multipath Tree Trace Example

The following example shows how to configure the MPLS LSP Multipath Tree Trace feature to interoperate with a vendor implementation that does not interpret RFC 4379 as Cisco does:

```
configure terminal
!
mpls oam
 echo revision 4
 no echo vendor-extension
 end
!
trace mpls multipath ipv4 10.131.161.151/32
```

The **echo revision** command is included in this example for completeness. The default echo revision number is 4, which corresponds to the RFC 4379.

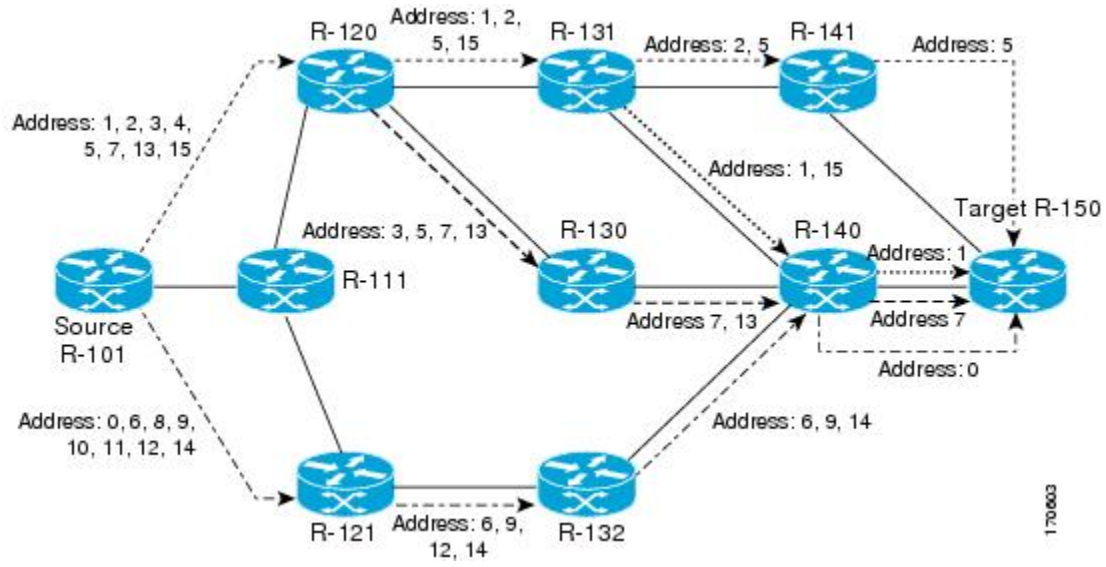
Discovering IPv4 Load Balancing Paths Using MPLS LSP Multipath Tree Trace Example

The following example shows how to use the MPLS LSP Multipath Tree Trace feature to discover IPv4 load balancing paths. The example is based on the sample network shown in the figure below. In this example, the bitmap size is set to 16. Therefore, path discovery starts by the MPLS LSP Multipath Tree Trace feature using 16 bitmapped addresses as it locates LSP paths from the source router R-101 to the target router R-150 with prefix and mask 10.1.1.150/32. The MPLS LSP Multipath Tree Trace feature starts using the 127.x.y.z/8 address space with 127.0.0.0.

```
Router# trace mpls multipath
ipv4 10.1.1.150/32 hashkey ipv4 bitmap 16
Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 468 ms
```

The output of the **trace mpls multipath** command in the example shows the result of path discovery with the MPLS LSP Multipath Tree Trace feature as shown in the figure below.

Figure 28: MPLS LSP Multipath Tree Trace Path Discovery in a Sample Network



Using DSCP to Request a Specific Class of Service in an Echo Reply Example

The following example shows how to use DSCP to request a specific CoS in an echo reply:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 reply dscp 50
Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 448 ms
```

Controlling How a Responding Router Replies to an MPLS Echo Request Example

The following example shows how to control how a responding router replies to an MPLS echo request:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 reply mode router-alert
Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0
 Total Time Elapsed 708 ms
```

Specifying the Output Interface for Echo Packets Leaving a Router for MPLS LSP Multipath Tree Trace Example

The following example shows how to specify the output interface for echo packets leaving a router for the MPLS LSP Multipath Tree Trace feature:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 output interface fastethernet0/0/0

Tracing MPLS Label Switched Path to 10.1.1.150/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
  0 10.1.111.101 MRU 1500 [Labels: 33 Exp: 0]
L
  1 10.1.111.111 MRU 1500 [Labels: 33 Exp: 0] 40 ms
L
  2 10.2.120.120 MRU 1500 [Labels: 33 Exp: 0] 20 ms
L
  3 10.3.131.131 MRU 1500 [Labels: 34 Exp: 0] 20 ms
L
```

```
  4 10.4.141.141 MRU 1504 [Labels: implicit-null Exp: 0] 20 ms !
  5 10.5.150.150 16 ms
```

Setting the Pace of MPLS Echo Request Packet Transmission for MPLS LSP Multipath Tree Trace Example

The following examples show how set the pace of MPLS echo request packet transmission for the MPLS LSP Multipath Tree Trace feature. The time between successive MPLS echo requests is set to 300 milliseconds in the first example and 400 milliseconds in the second example:

```
Router# trace mpls multipath ipv4 10.131.159.252/32 interval 300
Starting LSP Multipath Traceroute for 10.131.159.252/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LL!
Path 0 found,
 output interface Et1/0 source 10.2.3.2 destination 127.0.0.0
Paths (found/broken/unexplored) (1/0/0)
 Echo Request (sent/fail) (3/0)
 Echo Reply (received/timeout) (3/0)
 Total Time Elapsed 1604 ms
Router# trace mpls multipath ipv4 10.131.159.252/32 interval 400
Starting LSP Multipath Traceroute for 10.131.159.252/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LL!
Path 0 found,
 output interface Et1/0 source 10.2.3.2 destination 127.0.0.0
Paths (found/broken/unexplored) (1/0/0)
 Echo Request (sent/fail) (3/0)
 Echo Reply (received/timeout) (3/0)
 Total Time Elapsed 1856 ms
```

Notice that the elapsed time increases as you increase the interval size.

Enabling MPLS LSP Multipath Tree Trace to Detect LSP Breakages Example

The following examples shows how to enable the MPLS LSP Multipath Tree Trace feature to detect LSP breakages caused by an interface that lacks an MPLS configuration:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 force-explicit-null

Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
```

```
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 460 ms
```

This example shows the additional information provided when you add the **verbose** keyword to the command:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 force-explicit-null verbose
Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
 'L' - labeled output interface, 'B' - unlabeled output interface,
 'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
 'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
 'P' - no rx intf label prot, 'p' - premature termination of LSP,
 'R' - transit router, 'I' - unknown upstream index,
 'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
  1 10.1.111.111 10.2.121.121 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  2 10.2.121.121 10.3.132.132 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 1
L
  3 10.3.132.132 10.4.140.240 MRU 1500 [Labels: 32/explicit-null Exp: 0/0] ret code 8
multipaths 1
L
  4 10.4.140.240 10.5.150.50 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8 multipaths
 1 !
  5 10.5.150.50, ret code 3 multipaths 0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
  1 10.1.111.111 10.2.120.120 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  2 10.2.120.120 10.3.131.131 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  3 10.3.131.131 10.4.141.141 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 2
```

```
L
  4 10.4.141.141 10.5.150.150 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8 multipaths
 1
!
5 10.5.150.150, ret code 3 multipaths 0
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
 1 10.1.111.111 10.2.120.120 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
 2 10.2.120.120 10.3.131.131 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
 3 10.3.131.131 10.4.140.140 MRU 1500 [Labels: 32/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
 4 10.4.140.140 10.5.150.50 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8 multipaths
1 ! 5 10.5.150.50, ret code 3 multipaths 0
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
  0 10.1.111.101 10.1.111.111 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] multipaths 0
L
  1 10.1.111.111 10.2.120.120 MRU 1500 [Labels: 33/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  2 10.2.120.120 10.3.130.130 MRU 1500 [Labels: 34/explicit-null Exp: 0/0] ret code 8
multipaths 2
L
  3 10.3.130.130 10.4.140.40 MRU 1500 [Labels: 32/explicit-null Exp: 0/0] ret code 8
multipaths 1
L
  4 10.4.140.40 10.5.150.50 MRU 1504 [Labels: explicit-null Exp: 0] ret code 8 multipaths
1
!
  5 10.5.150.50, ret code 3 multipaths 0
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 492 ms
```

Requesting That a Transit Router Validate the Target FEC Stack for MPLS LSP Multipath Tree Trace Example

The following example shows how to request that a transit router validate the target FEC stack for the MPLS LSP Multipath Tree Trace feature:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 flags fec ttl 5


Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
```

```
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 464 ms
```

Target FEC stack validation is always done at the egress router when the **flags fec** keywords are specified in the **trace mpls multipath** command.

Setting the Number of Timeout Attempts for MPLS LSP Multipath Tree Trace Example

The following example sets the number of timeout attempts for the MPLS LSP Multipath Tree Trace feature to four:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 retry-count 4


Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLLL!
Path 0 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1
L!
Path 2 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.5
LL!
Path 3 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (4/0/0)
 Echo Request (sent/fail) (14/0)
 Echo Reply (received/timeout) (14/0)
 Total Time Elapsed 460 ms
```

The following output shows a **trace mpls multipath** command that found one unexplored path, one successful path, and one broken path:

```
Router# trace mpls multipath ipv4 10.1.1.150/32 retry-count 4

Starting LSP Multipath Traceroute for 10.1.1.150/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LLL....
Path 0 Unexplorable,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.0
LLL!
Path 1 found,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.1 B
Path 2 Broken,
 output interface Fe0/0/0 source 10.1.111.101 destination 127.0.0.7
Paths (found/broken/unexplored) (1/1/1)
 Echo Request (sent/fail) (12/0)
 Echo Reply (received/timeout) (8/4)
 Total Time Elapsed 7868 ms
```

Additional References

Related Documents

Related Topic	Document Title
MPLS-based functionalities	• *MPLS Label Distribution Protocol (LDP)* • *MPLS Label Switching Router MIB* • *MPLS Scalability Enhancements for the LSC LSR* • *MPLS Scalability Enhancements for the ATM LSR* • *MPLS Traffic Engineering (TE)—Automatic Bandwidth Adjustment for MPLS TE Tunnels* • *MPLS Traffic Engineering (TE)—Scalability Enhancements* • *MPLS Class of Service Enhancements* • *RFC 2233 Interfaces MIB*

Standards

Standard	Title
draft-ietf-mpls-te-mib-05	MPLS Traffic Engineering Management Information Base Using SMIv2

MIBs

MIB	MIBs Link
MPLS TE MIB Interfaces MIB MPLS TE STD MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2026	*The Internet Standards Process*
RFC 3812	Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Related Documents

Related Topic	Document Title
Concepts and configuration tasks for MPLS LSP ping or traceroute	MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2113	*IP Router Alert Option*
RFC 3443	*Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*
RFC 4377	*Operations and Management (OAM) Requirements for Multi-Protocol Label Switched (MPLS) Networks*
RFC 4378	*A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)*
RFC 4379	*Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS EM - MPLS LSP Multipath Tree Trace

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 34: Feature Information for MPLS EM—MPLS LSP Multipath Tree Trace

Feature Name	Releases	Feature Information
MPLS EM—MPLS LSP Multipath Tree Trace	Cisco IOS XE Release 2.3	The MPLS EM—MPLS LSP Multipath Tree Trace feature provides the means to discover all the possible paths of a label switched path (LSP) between an egress and ingress router. Once discovered, these paths can be retested on a periodic basis using Multiprotocol Label Switching (MPLS) LSP ping or traceroute. This feature is an extension to the MPLS LSP traceroute functionality for the tracing of IPv4 LSPs. MPLS Embedded Management (EM) is a set of standards and value-added services that facilitate the deployment, operation, administration, and management of MPLS-based networks in line with the fault, configuration, accounting, performance, and security (FCAPS) model. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
		The following commands were introduced or modified: **debug mpls lspv**, **echo**, **mpls oam**, **trace mpls**, **trace mpls multipath**.

Glossary

ECMP —equal-cost multipath. Multiple routing paths of equal cost that may be used for packet forwarding.

FEC —Forwarding Equivalence Class. A set of packets that can be handled equivalently for forwarding purposes and are thus suitable for binding to a single label. Examples include the set of packets destined for one address prefix and the packets in any flow.

flow —A set of packets traveling between a pair of hosts, or between a pair of transport protocol ports on a pair of hosts. For example, packets with the same source address, source port, destination address, and destination port might be considered a flow.

A flow is also a stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

localhost —A name that represents the host router (device). The localhost uses the reserved loopback IP address 127.0.0.1.

LSP —label switched path. A connection between two routers in which Multiprotocol Label Switching (MPLS) forwards the packets.

LSPV —Label Switched Path Verification. An LSP ping subprocess. It encodes and decodes Multiprotocol Label Switching (MPLS) echo requests and replies, and it interfaces with IP, MPLS, and AToM switching for sending and receiving MPLS echo requests and replies. At the MPLS echo request originator router, LSPV maintains a database of outstanding echo requests for which echo responses have not been received.

MPLS router alert label —An Multiprotocol Label Switching (MPLS) label of 1. An MPLS packet with a router alert label is redirected by the router to the Route Processor (PR) processing level for handling. This allows these packets to bypass any forwarding failures in hardware routing tables.

OAM —Operation, Administration, and Management.

punt —Redirect packets with a router alert from the line card or interface to Route Processor (RP) level processing for handling.

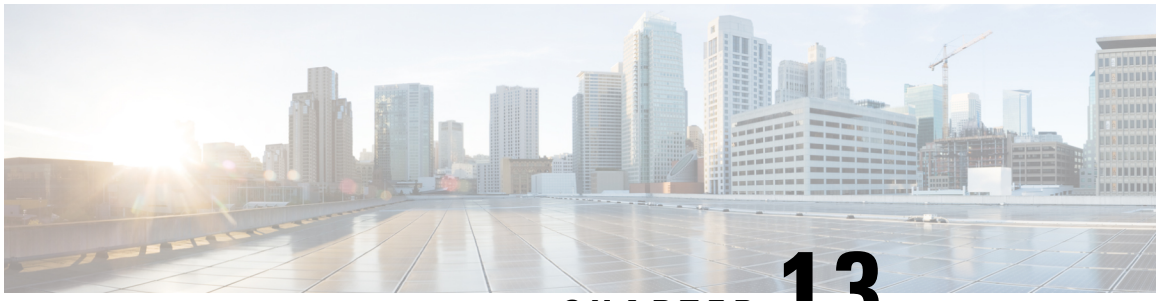
RP —Route Processor. The processor module contains the CPU, system software, and most of the memory components that are used in the router.

TTL —time-to-live. A parameter you can set that indicates the maximum number of hops a packet should take to reach its destination.

TLV —type, length, values. A block of information included in a Cisco Discovery Protocol address.

UDP —User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, so error processing and retransmission must be handled by other protocols. UDP is defined in RFC 768.

XDR —eXternal Data Representation. Standard for machine-independent data structures developed by Sun Microsystems. Used to transport messages between the Route Processor (RP) and the line card.



CHAPTER 13

MPLS Label Distribution Protocol MIB

This document describes the Simple Network Management Protocol (SNMP) agent support provided in Cisco software for the MPLS Label Distribution Protocol Management Information Base (MPLS LDP MIB).

Restrictions for MPLS LDP MIB

The MPLS LDP MIB is limited to read-only (RO) permission for MIB objects, except for MIB object mplsLdpSessionUpDownTrapEnable, which is writable by the SNMP agent.

Setting this object to a value of true enables both the mplsLdpSessionUp and mplsLdpSessionDown notifications on the Label Switched Router (LSR); conversely, setting this object to a value of false disables both of these notifications. The value of the mplsLdpSessionUpDownTrapEnable object is stored in NVRAM on the MPLS LDP MIB host.

For a description of notification events, see the Events Generating MPLS LDP MIB Notifications section.

Most MPLS LDP MIB objects are set up automatically during the LDP peer discovery (Hello) process and the subsequent negotiation of parameters and establishment of LDP sessions between the LDP peers.

Information About MPLS LDP MIB

MPLS LDP Overview

Multiprotocol Label Switching (MPLS) is a packet forwarding technology that uses a short, fixed-length value called a label in packets to determine the next hop for packet transport through an MPLS network by means of label switching routers (LSRs).

A fundamental MPLS principle is that LSRs in an MPLS network must agree on the definition of the labels being used for packet forwarding operations. Label agreement is achieved in an MPLS network by means of procedures defined in the Label Distribution Protocol (LDP).

LDP operations begin with a discovery (Hello) process, during which an LDP entity (a local LSR) finds a cooperating LDP peer in the network and negotiates basic operating procedures between them. The recognition and identification of a peer by means of this discovery process results in a Hello adjacency, which represents the context within which label binding information is exchanged between the local LSR and its LDP peer. An LDP function then creates an active LDP session between the two LSRs to effect the exchange of label binding information. The result of this process, when carried to completion with respect to all the LSRs in an MPLS network, is a label switched path (LSP), which constitutes an end-to-end packet transmission pathway between the communicating network devices.

By means of LDP, LSRs can collect, distribute, and release label binding information to other LSRs in an MPLS network, thereby enabling the hop-by-hop forwarding of packets in the network along normally routed paths.

MPLS LDP MIB Overview

The MPLS LDP MIB has been implemented to enable standard, SNMP-based network management of the label switching features in Cisco software. Providing this capability requires SNMP agent code to execute on a designated network management station (NMS) in the network. The NMS serves as the medium for user interaction with the network management objects in the MPLS LDP MIB.

The SNMP agent embodies a layered structure that is compatible with Cisco software and presents a network administrative and management interface to the objects in the MPLS LDP MIB and, thence, to the rich set of label switching capabilities supported by Cisco software.

By means of an SNMP agent, you can access MPLS LDP MIB objects using standard SNMP **get** operations to accomplish a variety of network management tasks. All the objects in the MPLS LDP MIB follow the conventions defined in the Internet Engineering Task Force (IETF) draft MIB entitled *draft-ietf-mpls-ldp-mib-08.txt*, which defines network management objects in a structured and standardized manner. This draft MIB is continually evolving toward the status of a standard. Accordingly, the MPLS LDP MIB will be implemented in a manner that tracks the evolution of this IETF document.

Slight differences that exist between the IETF draft MIB and the implementation of equivalent functions in Cisco software require some minor translations between the MPLS LDP MIB objects and the internal data structures of Cisco software. Such translations are accomplished by the SNMP agent, which runs in the background on the NMS workstation as a low priority process.

The MPLS LDP MIB provides the following functions:

- The MPLS LDP MIB can generate and send event notification messages to signal changes in the status of LDP sessions.
- You can enable and disable event notification messages by using SNMP CLI commands.
- You can specify the name or the IP address of an NMS workstation where event notification messages are sent to serve network administrative and management purposes.
- You can store the configuration pertaining to an event notification message in nonvolatile memory (NVRAM) of the NMS.

The structure of the MPLS LDP MIB conforms to Abstract Syntax Notation One (ASN.1), thereby forming a highly structured and idealized database of network management objects.

Using any standard SNMP application, you can retrieve and display information from the MPLS LDP MIB by means of standard SNMP GET operations. Similarly, you can traverse and display information in the MIB by means of SNMP GETNEXT operations.

Note Because the MPLS LDP MIB was not given an Internet Assigned Numbers Authority (IANA) Experimental OID at the time of its implementation, Cisco chose to implement the MIB under the Cisco Experimental OID number: ciscoExperiment 1.3.6.1.4.1.9.10 mplsLdpMIB 1.3.6.1.4.1.9.10.65 If the MPLS LDP MIB is assigned an IANA Experimental OID number, Cisco will deprecate all objects in the MIB under the Cisco Experimental OID and reposition the objects under the IANA Experimental OID.

Benefits of Using MPLS LDP MIB

The MPLS LDP MIB provides the following benefits:

- Establishing LDP sessions between peer devices in an MPLS network
- Retrieving MIB parameters relating to the operation of LDP entities, such as:
  - Well-known LDP discovery port
  - Maximum transmission unit (MTU)
  - Proposed KeepAlive timer interval
  - Loop detection
  - Session establishment thresholds
  - Range of VPI/VCI pairs to be used in forming labels
- Gathering statistics related to LDP operations, such as:
  - Count of the total established sessions for an LDP entity
  - Count of the total attempted sessions for an LDP entity
- Monitoring the time remaining for Hello adjacencies
- Monitoring the characteristics and status of LDP peers, such as:
  - Type of internetwork layer address of LDP peers
  - Actual internetwork layer address of LDP peers
  - Default MTU of the LDP peer
  - Number of seconds the LDP peer proposes as the value of the KeepAlive interval
  - Establishment of VPI/VCI label ranges to be made known to LDP peers
- Monitoring the characteristics and status of LDP sessions, such as:
  - Determining the LDP version being used by the LDP session
  - Determining the KeepAlive hold time remaining for an LDP session
  - Determining the state of an LDP session (whether the session is active or not)
  - Determining the range of VPI/VCI pairs to be used by an LDP session
  - Determining the last active interface of an LDP session

Description of MPLS LDP MIB Elements

The MPLS LDP MIB includes the following elements:

- LDP entity--Relates to an instance of LDP for purposes of exchanging label spaces.

- LDP peer--Refers to a remote LDP entity (that is, a nonlocal LSR).
- LDP session--Refers to an active LDP process between a local LSR and a remote LDP peer.
- Hello adjacency--Refers to the result of an LDP discovery process which affirms the state of two LSRs in an MPLS network as being adjacent to each other (that is, as being LDP peers).

A Hello adjacency constitutes the working context between two LSRs in an MPLS network. The adjacency is used for the exchange of label binding information.

These MPLS LDP MIB elements are briefly described under separate headings below.

In effect, the MPLS LDP MIB provides a network management database that supports real-time access to the various MIB objects within, reflecting the current state of MPLS LDP operations in the network. This network management information database is accessible by means of standard SNMP commands issued from an NMS in the MPLS/LDP operating environment.

The MPLS LDP MIB supports the following network management and administrative activities:

- Retrieving MPLS LDP MIB parameters pertaining to LDP operations
- Monitoring the characteristics and the status of LDP peers
- Monitoring the status of LDP sessions between LDP peers
- Monitoring Hello adjacencies in the network
- Gathering statistics regarding LDP sessions

LDP Entities

An LDP entity is uniquely identified by an LDP identifier having the object name *mplsLdpEntityLdpId* . This object consists of the router ID (four octets) and an interface number (two octets). The router ID encodes an IP address assigned to the LSR. The interface number identifies a specific label space available within the LSR.

An LDP entity represents a label space that is targeted for distribution to an LDP peer. In the case of an interface-specific LDP entity, the label space is distributed to a single LDP peer by means of a single LDP session.

Conversely, a platform-wide LDP entity can be associated with multiple LDP peers. In this case, the label space is distributed to multiple LDP peers by means of a separate LDP session pertaining to each peer.

LDP Peers

If an LSR has a label space to advertise to another LSR, or to multiple LSRs, there would be one LDP session for each LSR receiving the label space information. The receiver of the label space information is referred to as an LDP peer.

Per-interface label spaces are advertised to a single LDP peer by means of a single LDP session. *Per-platform* label spaces are advertised to multiple LDP peers by means of multiple LDP sessions.

The possible existence of multiple per-platform LDP peers dictates not only that an LDP entity be identified by its unique LDP tag, but also by its LDP index. In this case, the label space is the same, but the LDP Index differentiates the LDP session over which the label space is distributed to multiple LDP peers.

LDP Sessions

LDP sessions between local entities and remote peers distribute label spaces. There is always a one-to-one correspondence between an LDP peer and an LDP session. A single LDP session is a label distribution protocol instance that communicates across one or more network links with a single LDP peer. In the case of a platform-wide local LDP entity, there may be multiple LDP sessions and a corresponding number of remote LDP peers.

LDP Hello Adjacencies

An LDP session is an LDP instance that communicates across one or more network links to a peer protocol instance. An LDP Hello adjacency exists for each link on which LDP runs. Multiple link adjacencies exist whenever there are multiple links to the same LDP peer. In the case of a platform-wide label space, for example, there is a separate LDP peer/LDP session relationship for each LSR to which a label space may be advertised.

MPLS LDP MIB Object Categories

The MPLS LDP MIB contains numerous definitions of managed objects for the MPLS Label Distribution Protocol, as defined in the IETF draft document entitled *draft-ietf-mpls-ldp-08.txt* .

The managed objects in the MPLS LDP MIB are structured according to the following categories:

- MPLS LDP Textual Conventions
- MPLS LDP Objects
- MPLS Label Distribution Protocol Entity Objects
- LDP Entity Objects for Generic Labels
- LDP Entity Objects for ATM
- MPLS LDP Entity Configured ATM Label Range Table
- MPLS Entity Objects for Frame Relay
- Frame Relay Label Range Components
- MPLS LDP Entity Statistics Table
- MPLS LDP Entity Peer Table
- MPLS LDP Hello Adjacency Table
- MPLS LDP Sessions Table
- MPLS LDP ATM Session Information
- MPLS LDP Frame Relay Session Information
- MPLS LDP Session Statistics Table
- Address Message/Address Withdraw Message Information
- MPLS LDP LIB Table
- MPLS LDP FEC Table
- Notifications

- Module Conformance Statement

Events Generating MPLS LDP MIB Notifications

When you enable MPLS LDP MIB notification functionality by issuing the **snmp-server enable traps mpls ldp** command, notification messages are generated and sent to a designated NMS in the network to signal the occurrence of specific events within Cisco software.

The MPLS LDP MIB objects that announce LDP status transitions and event notifications include the following:

- mplsLdpSessionUp--This message is generated when an LDP entity (a local LSR) establishes an LDP session with another LDP entity (an adjacent LDP peer in the network).
- mplsLdpSessionDown--This message is generated when an LDP session between a local LSR and its adjacent LDP peer is terminated.

The up and down notifications indicate the last active interface in the LDP session.

- mplsLdpPathVectorLimitMismatch--This message is generated when a local LSR establishes an LDP session with its adjacent peer LSR, but the two LSRs have dissimilar path vector limits.

The value of the path vector limit can range from 0 to 255; a value of 0 indicates that loop detection is off; any value other than 0 up to 255 indicates that loop detection is on and, in addition, specifies the maximum number of hops through which an LDP message can pass before a loop condition in the network is sensed.

We recommend that all LDP-enabled routers in the network be configured with the same path vector limit. Accordingly, the mplsLdpPathVectorLimitMismatch object exists in the MPLS LDP MIB to provide a warning message to the NMS when two routers engaged in LDP operations have a dissimilar path vector limits.

- mplsLdpFailedInitSessionThresholdExceeded--This message is generated when a local LSR and an adjacent LDP peer attempt to set up an LDP session between them, but fail to do so after a specified number of attempts. The default number of attempts is 8. This default value is implemented in Cisco software and cannot be changed by either the CLI or an SNMP agent.

Eight failed attempts to establish an LDP session between a local LSR and an LDP peer, due to any type of incompatibility between the devices, causes this notification message to be generated.

In general, Cisco routers support the same features across multiple platforms. Therefore, the most likely incompatibility to occur between Cisco LSRs is a mismatch of their respective ATM VPI/VCI label ranges.

For example, if you specify a range of valid labels for an LSR that does not overlap the range of its adjacent LDP peer, the routers try eight times to create an LDP session between themselves before the mplsLdpFailedInitSessionThresholdExceeded notification is generated and sent to the NMS as an informational message.

Operationally, the LSRs whose label ranges do not overlap continue their attempt to create an LDP session between themselves after the eight retry limit is exceeded. In such cases, the LDP threshold exceeded notification alerts the network administrator to the existence of a condition in the network that may warrant attention.

RFC 3036, LDP Specification , details the incompatibilities that can exist between Cisco routers and/or other vendor LSRs in an MPLS network. Among such incompatibilities, for example, are the following:

- • Nonoverlapping ATM VPI/VCI ranges (as noted above) or nonoverlapping Frame-Relay DLCI ranges between LSRs attempting to set up an LDP session

- Unsupported label distribution method
- Dissimilar protocol data unit (PDU) sizes
- Dissimilar LDP feature support

How to Configure MPLS LDP MIB

Enabling the SNMP Agent for the MPLS LDP MIB

By default, the SNMP agent for the MPLS LDP MIB is disabled. To enable the SNMP agent on the host NMS workstation, perform the following procedure.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*]
5. **do copy running-config startup-config**
6. **exit**
7. **show running-config** [**interface** | **map-class**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. • If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*] **Example:**	Sets up the community access string to permit access to the SNMP protocol. • The *string* argument acts like a password and permits access to the SNMP protocol.

	Command or Action	Purpose
	`Router(config)# snmp-server community comaccess ro`	• The **view** *view-name* keyword and argument pair specifies the name of a previously defined view. The view defines the objects available to the community. • The **ro** keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects. • The **rw** keyword specifies read/write access. Authorized management stations are able to both retrieve and modify MIB objects. • The *acl-number* argument is an integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent.
Step 5	**do copy running-config startup-config** **Example:** `Router(config)# do copy running-config startup-config`	Saves the modified configuration to nonvolatile memory (NVRAM) as the startup configuration file. • The **do** command allows you to perform EXEC level commands in configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 7	**show running-config** [**interface** | **map-class**] **Example:** `Router# show running-config | include smnp-server`	(Optional) Displays the configuration information currently on the router, the configuration for a specific interface, or map-class information. • Use the **show running-config** command to check that the **snmp-server** statements appear in the output.

Configuring the Router to Send SNMP Traps

Perform this task to configure the router to send traps to a host.

The **snmp-server host** command specifies which hosts receive traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

For a host to receive a trap, an **snmp-server host** command must be configured for that host, and, generally, the trap must be enabled globally through the **snmp-server enable traps** command.

Note Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend that you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
4. **snmp-server enable traps mpls ldp** [**session-down**] [**session-up**] [**pv-limit**] [**threshold**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
	Example:
	`Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-ldp`
		Specifies the recipient of an SNMP notification operation.
		• The *host-addr* argument specifies the name or Internet address of the host (the targeted recipient).
		• The **traps** keyword sends SNMP traps to this host. This is the default.
		• The **informs** keyword sends SNMP informs to this host.
		• The **version** keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, because it allows packet encryption with the **priv** keyword. If you use the **version** keyword, you must specify one of the following:
		  • **1**--SNMPv1. This option is not available with informs.
		  • **2c**--SNMPv2C.
		  • **3**--SNMPv3. The following three optional keywords can follow the **version 3** keyword (**auth**, **noauth**, **priv**).
		• The *community-string* argument is a password-like community string sent with the notification operation.
		• The **udp-port** *port* keyword argument pair names the UDP port of the host to use. The default is 162.

	Command or Action	Purpose
		• The *notification-type* argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent. • The **vrf** *vrf-name* keyword argument pair specifies the VRF table that should be used to send SNMP notifications.
Step 4	**snmp-server enable traps mpls ldp** [**session-down**] [**session-up**] [**pv-limit**] [**threshold**] **Example:** `Router(config)# snmp-server enable traps mpls ldp session-down session-up`	Enables the router to send MPLS VPN- specific SNMP notifications (traps and informs). • The **session-down** keyword controls (enables or disables) LDP session down notifications (mplsLdpSessionDown). This message is generated when an LDP session between the router and its adjacent LDP peer is terminated. • The **session-up** keyword controls (enables or disables) LDP session up notifications (mplsLdpSessionUp). This notification is generated when the router establishes an LDP session with another LDP entity (an adjacent LDP peer in the network). • The **pv-limit** keyword controls (enables or disables) path-vector (PV) limit notifications (mplsLdpPathVectorLimitMismatch). This notification is generated when the router establishes an LDP session with its adjacent peer LSR, but the two LSRs have dissimilar path vector limits. • The **threshold** keyword controls (enables or disables) PV limit notifications (mplsLdpFailedInitSessionThresholdExceeded). This notification is generated after eight failed attempts to establish an LDP session between the router and an LDP peer. The failure can be the result of any type of incompatibility between the devices.
Step 5	**exit** **Example:** `Router(config)# exit`	(Optional) Exits to privileged EXEC mode.

Verifying the Status of the SNMP Agent

To verify that the SNMP agent has been enabled on the host NMS workstation, perform the following steps.

SUMMARY STEPS

1. **enable**

2. **show running-config**
3. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show running-config**

Use this command to display the running configuration on the host NMS and examine the output for SNMP information. For example:

Example:

```
Router# show running-config
.
.
.
snmp-server community public RO
snmp-server community private RO
```

The presence of any snmp-server statement in the output that takes the form shown above verifies that the SNMP agent has been enabled on the host NMS workstation.

Step 3 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS LDP MIB

Enabling the SNMP Agent Examples

The following example shows how to enable an SNMP agent on the host NMS:

```
Router# configure terminal
Router(config)# snmp-server community
```

The following example shows how to enable SNMPv1 and SNMPv2C on the host NMS. The configuration permits any SNMP agent to access all MPLS LDP MIB objects with read-only permission using the community string public.

```
Router(config)# snmp-server community public
```

The following example shows how to allow read-only access to all MPLS LDP MIB objects relating to members of access list 4 that specify the comaccess community string. No other SNMP agents will have access to any of the MPLS LDP MIB objects.

```
Router(config)# snmp-server community comaccess ro 4
```

The following example shows how to enable the session up and session down LDP notifications:

```
Router(config)# snmp-server enable traps mpls ldp session-up
Router(config)# snmp-server enable traps mpls ldp session-down
```

Additional References

Related Documents

Related Topic	Document Title
MPLS LDP configuration tasks	MPLS Label Distribution Protocol (LDP)
MPLS LDP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Multiprotocol Label Switching Command Reference*
SNMP commands	*Cisco IOS Network Management Command Reference*
SNMP configuration	"Configuring SNMP Support" in the *Network Management Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
• *MPLS LDP MIB*	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3036	*LDP Specification*
RFC 3037	*LDP Applicability*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

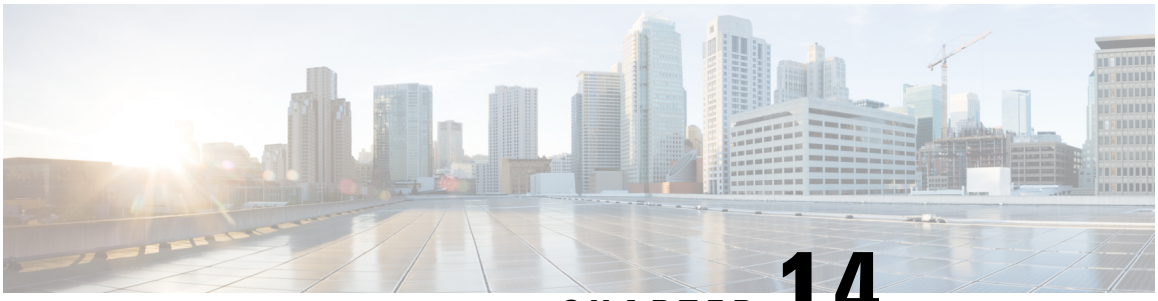
Feature Information for MPLS LDP MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 35: Feature Information for MPLS LDP MIB

Feature Name	Releases	Feature Information
MPLS LDP MIB	12.0(11)ST 12.2(2)T 12.0(21)ST 12.2(13)T 12.0(30)S 12.2(27)SBC 12.2(28)SB 12.2(33)SRA 12.2(33)SXH Cisco IOS XE Release 2.1	The MPLS LDP MIB has been implemented to enable standard, SNMP-based network management of the label switching features in Cisco software. In Cisco IOS Release 12.0(11)ST, this feature was introduced to provide SNMP agent support when using the MPLS LDP MIB on Cisco 7200, Cisco 7500, and Cisco 12000 series routers. In Cisco IOS Release 12.2(2)T, this feature was integrated to provide SNMP agent support when using the MPLS LDP MIB on Cisco 7200 and Cisco 7500 series routers. In Cisco IOS Release 12.0(21)ST, the **snmp-server enable traps mpls ldp** command was introduced. The **snmp-server enable traps mpls ldp** command was integrated into Cisco IOS Release 12.2(13)T. This feature was integrated into Cisco IOS Release 12.0(30)S. This feature was integrated into Cisco IOS Release 12.2(27)SBC. This feature was integrated into Cisco IOS Release 12.2(28)SB. This feature was integrated into Cisco IOS Release 12.2(33)SRA. This feature was integrated into Cisco IOS Release 12.2(33)SXH. This feature was integrated into Cisco IOS XE Release 2.1 and implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **snmp-server enable traps mpls ldp**
MPLS LDP—MIB Notifications	Cisco IOS XE Release 2.1	This feature provides SNMP traps for critical MPLS LDP events. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **snmp-server enable traps mpls ldp**



CHAPTER 14

MPLS Label Distribution Protocol MIB Version 8 Upgrade

The MPLS Label Distribution Protocol (LDP) MIB Version 8 Upgrade feature enhances the LDP MIB to support the Internet Engineering Task Force (IETF) draft Version 8.

Prerequisites for MPLS LDP MIB Version 8 Upgrade

- Simple Network Management Protocol (SNMP) must be installed and enabled on the label switch routers (LSRs).
- Multiprotocol Label Switching (MPLS) must be enabled on the LSRs.
- LDP must be enabled on the LSRs.

Restrictions for MPLS LDP MIB Version 8 Upgrade

This implementation of the MPLS LDP MIB is limited to read-only (RO) permission for MIB objects, except for MIB object *mplsLdpSessionUpDownTrapEnable* , which has been extended to be writable by the SNMP agent.

Setting this object to a value of true enables both the *mplsLdpSessionUp* and *mplsLdpSessionDown* notifications on the LSR; conversely, setting this object to a value of false disables both of these notifications.

For a description of notification events, see the Events Generating MPLS LDP MIB Notifications in MPLS LDP MIB Version 8 Upgrade section.

Most MPLS LDP MIB objects are set up automatically during the LDP peer discovery (hello) process and the subsequent negotiation of parameters and establishment of LDP sessions between the LDP peers.

The following tables are not implemented in this feature:

- mplsLdpEntityFrParmsTable
- mplsLdpEntityConfFrLRTable
- mplsLdpFrameRelaySesTable
- mplsFecTable
- mplsLdpSesInLabelMapTable
- mplsXCsFecsTable
- mplsLdpSesPeerAddrTable

Information About MPLS LDP MIB Version 8 Upgrade

Feature Design of MPLS LDP MIB Version 8 Upgrade

MPLS is a packet forwarding technology that uses a short, fixed-length value called a label in packets to specify the next hop for packet transport through an MPLS network by means of label switch routers (LSRs).

A fundamental MPLS principle is that LSRs in an MPLS network must agree on the definition of the labels being used for packet forwarding operations. Label agreement is achieved in an MPLS network by means of procedures defined in the LDP.

LDP operations begin with a discovery (hello) process, during which an LDP entity (a local LSR) finds a cooperating LDP peer in the network, and the two negotiate basic operating procedures. The recognition and identification of a peer by means of this discovery process results in a hello adjacency, which represents the context within which label binding information is exchanged between the local LSR and its LDP peer. LDP then creates an active LDP session between the two LSRs to effect the exchange of label binding information. When this process is carried to completion with respect to all of the LSRs in an MPLS network, the result is a label-switched path (LSP), which constitutes an end-to-end packet transmission pathway between the communicating network devices.

By means of LDP, LSRs can collect, distribute, and release label binding information to other LSRs in an MPLS network, thereby enabling the hop-by-hop forwarding of packets in the network along normally routed paths.

The MPLS LDP MIB has been implemented to enable standard, SNMP-based network management of the label switching features in Cisco software. Providing this capability requires SNMP agent code to execute on a designated network management station (NMS) in the network. The NMS serves as the medium for user interaction with the network management objects in the MPLS LDP MIB.

The SNMP agent code has a layered structure that is compatible with Cisco software and presents a network administrative and management interface to the objects in the MPLS LDP MIB and, thence, to the rich set of label switching capabilities supported by Cisco software.

By means of an SNMP agent, you can access MPLS LDP MIB objects using standard SNMP GET operations, and you can use those objects to accomplish a variety of network management tasks. All the objects in the

MPLS LDP MIB follow the conventions defined in the IETF draft MIB entitled *draft-ietf-mpls-ldp-mib-08.txt,* which defines network management objects in a structured and standardized manner. This draft MIB is evolving and is soon expected to be a standard. Accordingly, the MPLS LDP MIB will be implemented in such a way that it tracks the evolution of this IETF document.

However, slight differences exist between the IETF draft MIB and the implementation of equivalent Cisco functions. As a result, some minor translations between the MPLS LDP MIB objects and the internal Cisco data structures are needed. Such translations are accomplished by the SNMP agent, which runs in the background on the NMS workstation as a low-priority process.

The extensive Cisco label switching capabilities provide an integrated approach to managing the large volumes of traffic carried by WANs. These capabilities are integrated into the Layer 3 network services, thus optimizing the routing of high-volume traffic through Internet service provider backbones while, at the same time, ensuring the resistance of the network to link or node failures.

The MPLS Label Distribution Protocol MIB Version 8 Upgrade supports the following functions:

- Tag Distribution Protocol (TDP) (This protocol might not be supported in all software releases.)
- Generation and sending of event notification messages that signal changes in the status of LDP sessions
- Enabling and disabling of event notification messages by means of extensions to existing SNMP CLI commands
- Specification of the name or the IP address of an NMS workstation in the operating environment to which Cisco event notification messages are to be sent to serve network administrative and management purposes
- Storage of the configuration pertaining to an event notification message in NVRAM of the NMS

The structure of the MPLS LDP MIB conforms to Abstract Syntax Notation One (ASN.1), so the MIB forms a highly structured and idealized database of network management objects.

Using any standard SNMP application, you can retrieve and display information from the MPLS LDP MIB by means of standard SNMP GET and GETNEXT operations.

Note Because the MPLS LDP MIB was not given an Internet Assigned Numbers Authority (IANA) experimental object identifier (OID) at the time of its implementation, Cisco chose to implement the MIB under the ciscoExperimental OID number, as follows: ciscoExperimental 1.3.6.1.4.1.9.10 mplsLdpMIB 1.3.6.1.4.1.9.10.65 If the MPLS LDP MIB is assigned an IANA Experimental OID number, Cisco will replace all objects in the MIB under the ciscoExperimental OID and reposition the objects under the IANA Experimental OID.

Enhancements in Version 8 of the MPLS LDP MIB

Version 8 of the MPLS LDP MIB contains the following enhancements:

- TDP support (This protocol might not be supported in all software releases.)
- Upgraded objects
- New indexing that is no longer based on the number of sessions
- Multiple SNMP context support for Virtual Private Networks (VPNs)

Benefits of MPLS LDP MIB Version 8 Upgrade

- Supports TDP and LDP (TDP might not be supported in all software releases.)
- Establishes LDP sessions between peer devices in an MPLS network
- Retrieves MIB parameters relating to the operation of LDP entities, such as:
  - Well-known LDP discovery port
  - Maximum transmission unit (MTU)
  - Proposed keepalive timer interval
  - Loop detection
  - Session establishment thresholds
  - Range of virtual path identifier/virtual channel identifier (VPI/VCI) pairs to be used in forming labels
- Gathers statistics related to LDP operations, such as error counters.
- Monitors the time remaining for hello adjacencies
- Monitors the characteristics and status of LDP peers, such as:
  - Internetwork layer address of LDP peers
  - Loop detection of the LDP peers
  - Default MTU of the LDP peer
  - Number of seconds the LDP peer proposes as the value of the keepalive interval
- Monitors the characteristics and status of LDP sessions, such as:
  - Displaying the error counters.
  - Determining the LDP version being used by the LDP session
  - Determining the keepalive hold time remaining for an LDP session
  - Determining the state of an LDP session (whether the session is active or not)
  - Displaying the label ranges for platform-wide and interface-specific sessions
  - Displaying the ATM parameters.

Description of MPLS LDP MIB Elements for MPLS LDP MIB Version 8 Upgrade

LDP operations related to an MPLS LDP MIB involve the following functional elements:

- LDP entity--Relates to an instance of LDP for purposes of exchanging label spaces; describes a potential session.
- LDP peer--Refers to a remote LDP entity (that is, a nonlocal LSR).
- LDP session--Refers to an active LDP process between a local LSR and a remote LDP peer.
- Hello adjacency--Refers to the result of an LDP discovery process that affirms the state of two LSRs in an MPLS network as being adjacent to each other (that is, as being LDP peers). When the neighbor is discovered, the neighbor becomes a hello adjacency. An LDP session can be established with the hello adjacency. After the session is established, label bindings can be exchanged between the LSRs.

These MPLS LDP MIB elements are briefly described under separate headings below.

In effect, the MPLS LDP MIB provides a network management database that supports real-time access to the various MIB objects in the database. This database reflects the current state of MPLS LDP operations in the network. You can access this network management information database by means of standard SNMP commands issued from an NMS in the MPLS LDP operating environment.

The MPLS LDP MIB supports the following network management and administrative activities:

- Retrieving MPLS LDP MIB parameters pertaining to LDP operations
- Monitoring the characteristics and the status of LDP peers
- Monitoring the status of LDP sessions between LDP peers
- Monitoring hello adjacencies in the network
- Gathering statistics regarding LDP sessions

LDP Entities

An LDP entity is uniquely identified by an LDP identifier that consists of the mplsLdpEntityLdpId and the mplsLdpEntityIndex (see the figure below).

- The mplsLdpEntityLdpId consists of the local LSR ID (four octets) and the label space ID (two octets). The label space ID identifies a specific label space available within the LSR.
- The mplsLdpEntityIndex consists of the IP address of the peer active hello adjacency, which is the 32-bit representation of the IP address assigned to the peer LSR.

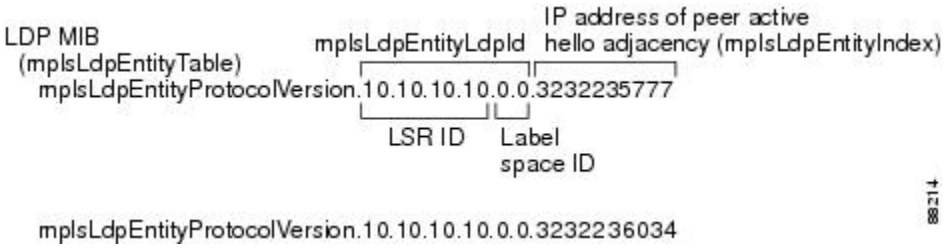
The mplsldpEntityProtocolVersion is a sample object from the mplsLdpEntityTable.

The figure shows the following indexing:

- mplsLdpEntityLdpId = 10.10.10.10.0.0
- LSR ID = 10.10.10.10
- Label space ID = 0.0

The mplsLdpEntityLdpId or the LDP ID consists of the LSR ID and the label space ID.

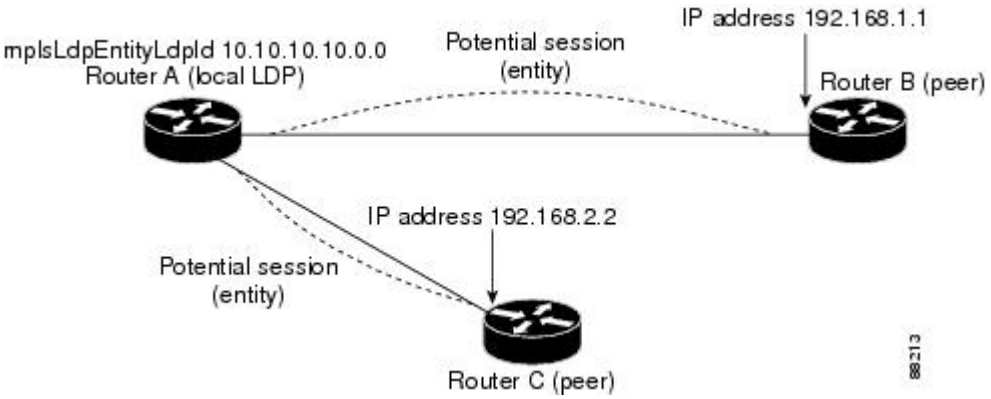
- The IP address of peer active hello adjacency or the mplsLdpEntityIndex = 3232235777, which is the 32-bit representation of the IP address assigned to the peer's active hello adjacency.



An LDP entity represents a label space that has the potential for a session with an LDP peer. An LDP entity is set up when a hello adjacency receives a hello message from an LDP peer.

In the figure below, Router A has potential sessions with two remote peers, Routers B and C. The mplsLdpEntityLdpId is 10.10.10.10.0.0, and the IP address of the peer active hello adjacency

(mplsLdpEntityIndex) is 3232235777, which is the 32-bit representation of the IP address 192.168.1.1 for Router B.



LDP Sessions and Peers

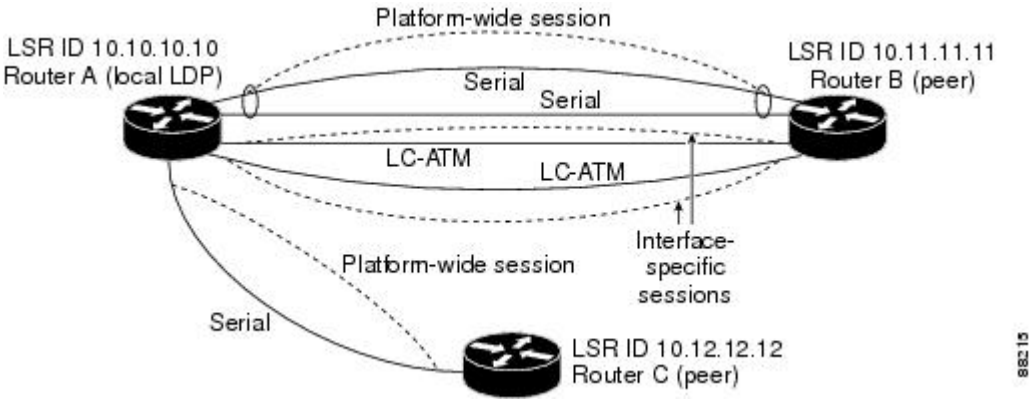
LDP sessions exist between local entities and remote peers for the purpose of distributing label spaces. There is always a one-to-one correspondence between an LDP peer and an LDP session. A single LDP session is an LDP instance that communicates across one or more network links with a single LDP peer.

LDP supports the following types of sessions:

- Interface-specific--An interface-specific session uses interface resources for label space distributions. For example, each label-controlled ATM (LC-ATM) interface uses its own VPIs/VCIs for label space distributions. Depending on its configuration, an LDP platform can support zero, one, or more interface-specific sessions. Each LC-ATM interface has its own interface-specific label space and a nonzero label space ID.
- Platform-wide--An LDP platform supports a single platform-wide session for use by all interfaces that can share the same global label space. For Cisco platforms, all interface types except LC-ATM use the platform-wide session and have a label space ID of zero.

When a session is established between two peers, entries are created in the mplsLdpPeerTable and the mplsLdpSessionTable because they have the same indexing.

In the figure below, Router A has two remote peers, Routers B and C. Router A has a single platform-wide session that consists of two serial interfaces with Router B and another platform-wide session with Router C. Router A also has two interface-specific sessions with Router B.



The figure below shows entries that correspond to the mplsLdpPeerTable and the mplsLdpSessionTable in the figure above.

In the figure below, mplsLdpSesState is a sample object from the mplsLdpSessionTable on Router A. There are four mplsLdpSesState sample objects shown (top to bottom). The first object represents a platform-wide session associated with two serial interfaces. The next two objects represent interface-specific sessions for the LC-ATM interfaces on Routers A and B. These interface-specific sessions have nonzero peer label space IDs. The last object represents a platform-wide session for the next peer, Router C.

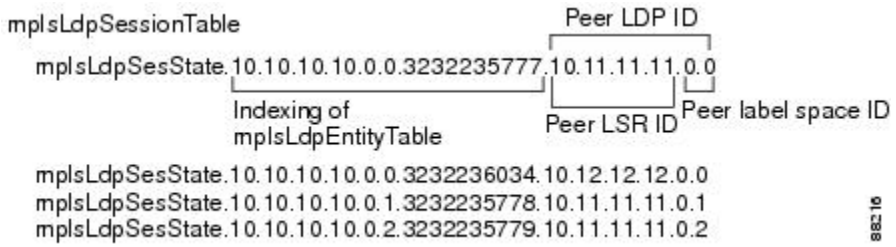
The indexing is based on the entries in the mplsLdpEntityTable. It begins with the indexes of the mplsLdpEntityTable and adds the following:

- Peer LDP ID = 10.11.11.11.0.0

The peer LDP ID consists of the peer LSR ID (four octets) and the peer label space ID (two octets).

- Peer LSR ID = 10.11.11.11
- Peer label space ID = 0.0

The peer label space ID identifies a specific peer label space available within the LSR.



LDP Hello Adjacencies

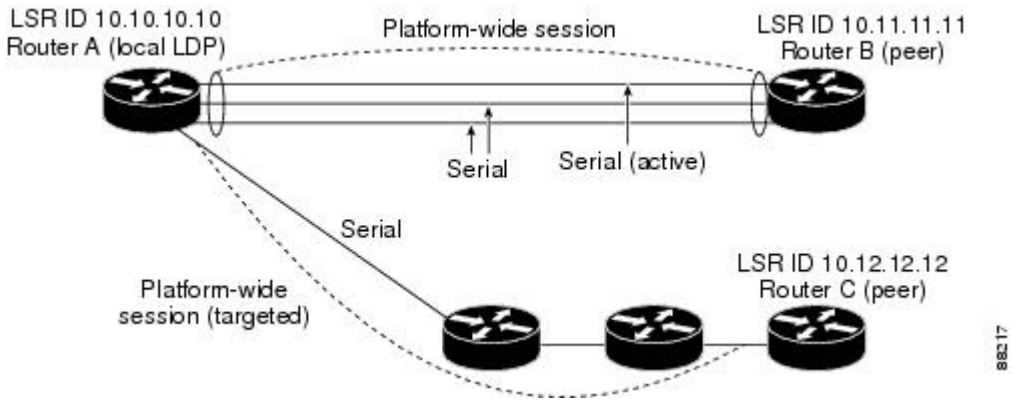
An LDP hello adjacency is a network link between a router and its peers. An LDP hello adjacency enables two adjacent peers to exchange label binding information.

An LDP hello adjacency exists for each link on which LDP runs. Multiple LDP hello adjacencies exist whenever there is more than one link in a session between a router and its peer, such as in a platform-wide session.

A hello adjacency is considered active if it is currently engaged in a session, or nonactive if it is not currently engaged in a session.

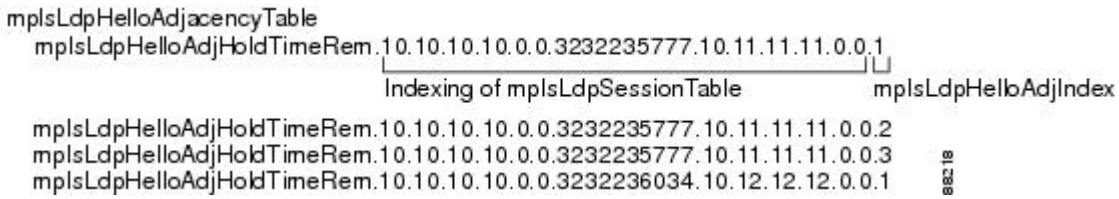
A targeted hello adjacency is not directly connected to its peer and has an unlimited number of hops between itself and its peer. A linked hello adjacency is directly connected between two routers.

In the figure below, Router A has two remote peers, Routers B and C. Router A has a platform-wide session with Router B that consists of three serial interfaces, one of which is active and another platform-wide (targeted) session with Router C.



The figure below shows entries in the mplsLdpHelloAdjacencyTable. There are four mplsLdpHelloAdjHoldTime sample objects (top to bottom). They represent the two platform-wide sessions and the four serial links shown in the figure above.

The indexing is based on the mplsLdpSessionTable. When the mplsLdpHelloAdjIndex enumerates the different links within a single session, the active link is mplsLdpHelloAdjIndex = 1.



Events Generating MPLS LDP MIB Notifications in MPLS LDP MIB Version 8 Upgrade

When you enable MPLS LDP MIB notification functionality by issuing the **snmp-server enable traps mpls ldp** command, notification messages are generated and sent to a designated NMS in the network to signal the occurrence of specific events within the network.

The MPLS LDP MIB objects involved in LDP status transitions and event notifications include the following:

- mplsLdpSessionUp--This message is generated when an LDP entity (a local LSR) establishes an LDP session with another LDP entity (an adjacent LDP peer in the network).
- mplsLdpSessionDown--This message is generated when an LDP session between a local LSR and its adjacent LDP peer is terminated.
- mplsLdpPathVectorLimitMismatch--This message is generated when a local LSR establishes an LDP session with its adjacent peer LSR, but the two LSRs have dissimilar path vector limits.

The value of the path vector limit can range from 0 through 255; a value of 0 indicates that loop detection is off; any value other than zero up to 255 indicates that loop detection is on and, in addition, specifies the maximum number of hops through which an LDP message can pass before a loop condition in the network is sensed.

We recommend that all LDP-enabled routers in the network be configured with the same path vector limit. Accordingly, the mplsLdpPathVectorLimitMismatch object exists in the MPLS LDP MIB to provide a warning message to the NMS when two routers engaged in LDP operations have different path vector limits.

Note This notification is generated only if the distribution method is downstream-on-demand.

- mplsLdpFailedInitSessionThresholdExceeded--This message is generated when a local LSR and an adjacent LDP peer attempt to set up an LDP session between them, but fail to do so after a specified number of attempts. The default number of attempts is 8. This default value is implemented and cannot be changed.

Eight failed attempts to establish an LDP session between a local LSR and an LDP peer, due to any type of incompatibility between the devices, causes this notification message to be generated. Cisco routers support the same features across multiple platforms.

Therefore, the most likely incompatibility to occur between Cisco LSRs is a mismatch of their respective ATM VPI/VCI label ranges.

For example, if you specify a range of valid labels for an LSR that does not overlap the range of its adjacent LDP peer, the routers try eight times to create an LDP session between themselves before the mplsLdpFailedInitSessionThresholdExceeded notification is generated and sent to the NMS as an informational message.

The LSRs whose label ranges do not overlap continue their attempt to create an LDP session between themselves after the eight-retry threshold is exceeded.

In such cases, the LDP threshold exceeded notification alerts the network administrator about a condition in the network that might warrant attention.

RFC 3036, *LDP Specification* , details the incompatibilities that can exist between Cisco routers and/or other vendor LSRs in an MPLS network.

Among such incompatibilities, for example, are the following:

- • Nonoverlapping ATM VPI/VCI ranges (as noted above) or nonoverlapping Frame-Relay DLCI ranges between LSRs attempting to set up an LDP session
  - Unsupported label distribution method
  - Dissimilar protocol data unit (PDU) sizes
  - Dissimilar types of LDP feature support

MIB Tables in MPLS LDP MIB Version 8 Upgrade

Version 8 of the MPLS LDP MIB consists of the following tables:

- mplsLdpEntityTable --Contains entries for every active LDP hello adjacency. Nonactive hello adjacencies appear in the mplsLdpHelloAdjacencyTable, rather than this table. This table is indexed by the local LDP identifier for the interface and the IP address of the peer active hello adjacency.

The advantage of showing the active hello adjacency instead of sessions in this table is that the active hello adjacency can exist even if an LDP session is not active (cannot be established). Previous implementations of the IETF MPLS-LDP MIB used sessions as the entries in this table. This approach was inadequate because as sessions went down, the entries in the entity table would disappear completely because the agent code could no longer access them. This resulted in the MIB failing to provide information about failed LDP sessions.

Directed adjacencies are also shown in this table. These entries, however, are always up administratively (adminStatus) and operationally (operStatus), because the adjacencies disappear if the directed session fails.

Nondirected adjacencies might disappear from the MIB on some occasions, because adjacencies are deleted if the underlying interface becomes operationally down, for example.

- mplsLdpEntityConfGenLRTable --Contains entries for every LDP-enabled interface that is in the global label space. (For Cisco, this applies to all interfaces except LC-ATM. LC-ATM entities are shown in the mplsLdpEntityConfAtmLRTable instead.) Indexing is the same as it is for the mplsLdpEntityTable, except two indexes have been added, mplsLdpEntityConfGenLRMin and mplsLdpEntityConfGenLRMax. These additional indexes allow more than one label range to be defined. However, in the current Cisco implementation, only one global label range is allowed.
- mplsLdpEntityAtmParmsTable --Contains entries for every LDP-enabled LC-ATM interface. This table is indexed the same as the mplsLdpEntityTable although only LC-ATM interfaces are shown.
- mplsLdpEntityConfAtmLRTable --Contains entries for every LDP-enabled LC-ATM interface. Indexing is the same as it is for the mplsLdpEntityTable, except two indexes have been added, mplsLdpEntityConfAtmLRMinVpi and mplsLdpEntityConfAtmLRMinVci. These additional indexes allow more than one label range to be defined. However, in the current Cisco implementation, only one label range per LC-ATM interface is allowed.
- mplsLdpEntityStatsTable --Augments the mplsLdpEntityTable and shares the exact same indexing for performing GETand GETNEXT operations. This table shows additional statistics for entities.
- mplsLdpPeerTable --Contains entries for all peer sessions. This table is indexed by the local LDP identifier of the session, the IP address of the peer active hello adjacency, and the peer's LDP identifier.
- mplsLdpHelloAdjacencyTable --Contains entries for all hello adjacencies. This table is indexed by the local LDP identifier of the associated session, the IP address of the peer active hello adjacency, the LDP identifier for the peer, and an arbitrary index that is set to the list position of the adjacency.
- mplsLdpSessionTable --Augments the mplsLdpPeerTable and shares the same indexing for performing GET and GETNEXT operations. This table shows all sessions.
- mplsLdpAtmSesTable --Contains entries for LC-ATM sessions. Indexing is the same as it is for the mplsLdpPeerTable, except two indexes have been added, mplsLdpSesAtmLRLowerBoundVpi and mplsLdpSesAtmLRLowerBoundVci. These additional indexes allow more than one label range to be defined. However, in the current Cisco implementation, only one label range per LC-ATM interface is allowed.
- mplsLdpSesStatsTable --Augments the mplsLdpPeerTable and shares the exact same indexing for performing GET and GETNEXT operations. This table shows additional statistics for sessions.

mplsLdpEntityTable

The table below lists the mplsLdpEntityTable objects and their descriptions.

Table 36: mplsLdpEntityTable Objects and Descriptions

Object	Description
mplsLdpEntityEntry	Represents an LDP entity, which is a potential session between two peers.
mplsLdpEntityLdpId	The LDP identifier (not accessible) consists of the local LSR ID (four octets) and the label space ID (two octets).

Object	Description
mplsLdpEntityIndex	A secondary index that identifies this row uniquely. It consists of the IP address of the peer active hello adjacency, which is the 32-bit representation of the IP address assigned to the LSR (not accessible).
mplsLdpEntityProtocolVersion	The version number of the LDP protocol to be used in the session initialization message.
mplsLdpEntityAdminStatus	The administrative status of this LDP entity is always up. If the hello adjacency fails, this entity disappears from the mplsLdpEntityTable.
mplsLdpEntityOperStatus	The operational status of this LDP entity. Values are unknown(0), enabled(1), and disabled(2).
mplsLdpEntityTcpDscPort	The TCP discovery port for LDP or TDP. The default value is 646 (LDP).
mplsLdpEntityUdpDscPort	The UDP discovery port for LDP or TDP. The default value is 646 (LDP).
mplsLdpEntityMaxPduLength	The maximum PDU length that is sent in the common session parameters of an initialization message.
mplsLdpEntityKeepAliveHoldTimer	The two-octet value that is the proposed keepalive hold time for this LDP entity.
mplsLdpEntityHelloHoldTimer	The two-octet value that is the proposed hello hold time for this LDP entity.
mplsLdpEntityInitSesThreshold	The threshold for notification when this entity and its peer are engaged in an endless sequence of initialization messages. The default value is 8 and cannot be changed by SNMP or CLI.
mplsLdpEntityLabelDistMethod	The specified method of label distribution for any given LDP session. Values are downstreamOnDemand(1) and downstreamUnsolicited(2).
mplsLdpEntityLabelRetentionMode	Can be configured to use either conservative(1) for LC-ATM or liberal(2) for all other interfaces.
mplsLdpEntityPVLMisTrapEnable	Indicates whether the mplsLdpPVLMismatch trap should be generated. If the value is enabled(1), the trap is generated. If the value is disabled(2), the trap is not generated. The default is disabled(2). **Note** The mplsLdpPVLMismatch trap is generated only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).

Object	Description
mplsLdpEntityPVL	If the value of this object is 0, loop detection for path vectors is disabled. Otherwise, if this object has a value greater than zero, loop detection for path vectors is enabled, and the path vector limit is this value. **Note** The mplsLdpEntityPVL object is non-zero only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).
mplsLdpEntityHopCountLimit	If the value of this object is 0, loop detection using hop counters is disabled. If the value of this object is greater than 0, loop detection using hop counters is enabled, and this object specifies this entity's maximum allowable value for the hop count. **Note** The mplsLdpEntityHopCountLimit object is non-zero only if mplsLdpEntityLabelDistMethod is downstreamOnDemand(1).
mplsLdpEntityTargPeer	If this LDP entity uses a targeted adjacency, this object is set to true(1). The default value is false(2).
mplsLdpEntityTargPeerAddrType	The type of the internetwork layer address used for the extended discovery. This object indicates how the value of mplsLdpEntityTargPeerAddr is to be interpreted.
mplsLdpEntityTargPeerAddr	The value of the internetwork layer address used for the targeted adjacency.
mplsLdpEntityOptionalParameters	Specifies the optional parameters for the LDP initialization message. If the value is generic(1), no optional parameters are sent in the LDP initialization message associated with this entity. LC-ATM uses atmParameters(2) to specify that a row in the mplsLdpEntityAtmParmsTable corresponds to this entry. **Note** Frame Relay parameters are not supported.
mplsLdpEntityDiscontinuityTime	The value of sysUpTime on the most recent occasion when one or more of this entity's counters suffered a discontinuity. The relevant counters are the specific instances of any Counter32 or Counter64 object contained in the mplsLdpEntityStatsTable that are associated with this entity. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, this object contains a 0 value.
mplsLdpEntityStorType	The storage type for this entry is a read-only implementation that is always volatile.
mplsLdpEntityRowStatus	This object is a read-only implementation that is always active.

mplsLdpEntityConfGenLRTable

The table below lists the mplsLdpEntityConfGenLRTable objects and their descriptions.

Table 37: mplsLdpEntityConfGenLRTable Objects and Descriptions

Object	Description
mplsLdpEntityConfGenLREntry	A row in the LDP Entity Configurable Generic Label Range table. One entry in this table contains information on a single range of labels; the range is defined by an upper boundary (VPI/VCI pair) and a lower boundary (VPI/VCI pair). The current implementation supports one label range per entity.
mplsLdpEntityConfGenLRMin	The minimum label configured for this range (not accessible).
mplsLdpEntityConfGenLRMax	The maximum label configured for this range (not accessible).
mplsLdpEntityConfGenIfIndxOrZero	This value represents the SNMP IF-MIB index for the platform-wide entity. If the active hello adjacency is targeted, the value is 0.
mplsLdpEntityConfGenLRStorType	The storage type for this entry is a read-only implementation that is always volatile.
mplsLdpEntityConfGenLRRowStatus	This object is a read-only implementation that is always active.

mplsLdpEntityAtmParmsTable

The table below lists the mplsLdpEntityAtmParmsTable objects and their descriptions.

Table 38: mplsLdpEntityAtmParmsTable Objects and Descriptions

Object	Description
mplsLdpEntityAtmParmsEntry	Represents the ATM parameters and ATM information for this LDP entity.
mplsLdpEntityAtmIfIndxOrZero	This value represents the SNMP IF-MIB index for the interface-specific LC-ATM entity.
mplsLdpEntityAtmMergeCap	Denotes the merge capability of this entity.
mplsLdpEntityAtmLRComponents	Number of label range components in the initialization message. This also represents the number of entries in the mplsLdpEntityConfAtmLRTable that correspond to this entry.
mplsLdpEntityAtmVcDirectionality	If the value of this object is bidirectional(0), a given VCI within a given VPI is used as a label for both directions independently of one another. If the value of this object is unidirectional(1), a given VCI within a VPI designates one direction.

Object	Description
mplsLdpEntityAtmLsrConnectivity	The peer LSR can be connected indirectly by means of an ATM VP, so that the VPI values can be different on the endpoints. For that reason, the label must be encoded entirely within the VCI field. Values are direct(1), the default, and indirect(2).
mplsLdpEntityDefaultControlVpi	The default VPI value for the non-MPLS connection.
mplsLdpEntityDefaultControlVci	The default VCI value for the non-MPLS connection.
mplsLdpEntityUnlabTrafVpi	VPI value of the VCC supporting unlabeled traffic. This non-MPLS connection is used to carry unlabeled (IP) packets.
mplsLdpEntityUnlabTrafVci	VCI value of the VCC supporting unlabeled traffic. This non-MPLS connection is used to carry unlabeled (IP) packets.
mplsLdpEntityAtmStorType	The storage type for this entry is a read-only implementation that is always volatile.
mplsLdpEntityAtmRowStatus	This object is a read-only implementation that is always active.

mplsLdpEntityConfAtmLRTable

The table below lists the mplsLdpEntityConfAtmLRTable objects and their descriptions.

Table 39: mplsLdpEntityConfAtmLRTable Objects and Descriptions

Object	Description
mplsLdpEntityConfAtmLREntry	A row in the LDP Entity Configurable ATM Label Range Table. One entry in this table contains information on a single range of labels; the range is defined by an upper boundary (VPI/VCI pair) and a lower boundary (VPI/VCI pair). This is the same data used in the initialization message. This label range should overlap the label range of the peer.
mplsLdpEntityConfAtmLRMinVpi	The minimum VPI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMinVci	The minimum VCI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMaxVpi	The maximum VPI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRMaxVci	The maximum VCI number configured for this range (not accessible).
mplsLdpEntityConfAtmLRStorType	The storage type for this entry is a read-only implementation that is always volatile.
mplsLdpEntityConfAtmLRRowStatus	This object is a read-only implementation that is always active.

mplsLdpEntityStatsTable

The table below lists the mplsLdpEntityStatsTable objects and their descriptions.

Table 40: mplsLdpEntityStatsTable Objects and Descriptions

Object	Description
mplsLdpEntityStatsEntry	These entries augment the mplsLdpEntityTable by providing additional information for each entry.
mplsLdpAttemptedSessions	Not supported in this feature.
mplsLdpSesRejectedNoHelloErrors	A count of the session rejected/no hello error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedAdErrors	A count of the session rejected/parameters advertisement mode error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedMaxPduErrors	A count of the session rejected/parameters max PDU length error notification messages sent or received by this LDP entity.
mplsLdpSesRejectedLRErrors	A count of the session rejected/parameters label range notification messages sent or received by this LDP entity.
mplsLdpBadLdpIdentifierErrors	A count of the number of bad LDP identifier fatal errors detected by the session associated with this LDP entity.
mplsLdpBadPduLengthErrors	A count of the number of bad PDU length fatal errors detected by the session associated with this LDP entity.
mplsLdpBadMessageLengthErrors	A count of the number of bad message length fatal errors detected by the session associated with this LDP entity.
mplsLdpBadTlvLengthErrors	A count of the number of bad Type-Length-Value (TLV) length fatal errors detected by the session associated with this LDP entity.
mplsLdpMalformedTlvValueErrors	A count of the number of malformed TLV value fatal errors detected by the session associated with this LDP entity.
mplsLdpKeepAliveTimerExpErrors	A count of the number of session keepalive timer expired errors detected by the session associated with this LDP entity.
mplsLdpShutdownNotifReceived	A count of the number of shutdown notifications received related to the session associated with this LDP entity.
mplsLdpShutdownNotifSent	A count of the number of shutdown notifications sent related to the session associated with this LDP entity.

mplsLdpPeerTable

The table below lists the mplsLdpPeerTable objects and their descriptions.

Table 41: mplsLdpPeerTable Objects and Descriptions

Object	Description
mplsLdpPeerEntry	Information about a single peer that is related to a session (not accessible). **Note** This table is augmented by the mplsLdpSessionTable.
mplsLdpPeerLdpId	The LDP identifier of this LDP peer (not accessible) consists of the peer LSR ID (four octets) and the peer label space ID (two octets).
mplsLdpPeerLabelDistMethod	For any given LDP session, the method of label distribution. Values are downstreamOnDemand(1) and downstreamUnsolicited(2).
mplsLdpPeerLoopDetectionForPV	An indication of whether loop detection based on path vectors is disabled or enabled for this peer. For downstream unsolicited distribution (mplsLdpPeerLabelDistMethod is downstreamUnsolicited(2)), this object always has a value of disabled(0) and loop detection is disabled. For downstream-on-demand distribution (mplsLdpPeerLabelDistMethod is downstreamOnDemand(1)), this object has a value of enabled(1), provided that loop detection based on path vectors is enabled.
mplsLdpPeerPVL	If the value of mplsLdpPeerLoopDetectionForPV for this entry is enabled(1), this object represents that path vector limit for this peer. If the value of mplsLdpPeerLoopDetectionForPV for this entry is disabled(0), this value should be 0.

mplsLdpHelloAdjacencyTable

The table below lists the mplsLdpHelloAdjacencyTable objects and their descriptions.

Table 42: mplsLdpHelloAdjacencyTable Objects and Descriptions

Object	Description
mplsLdpHelloAdjacencyEntry	Each row represents a single LDP hello adjacency. An LDP session can have one or more hello adjacencies (not accessible).
mplsLdpHelloAdjIndex	An identifier for this specific adjacency (not accessible). The active hello adjacency has mplsLdpHelloAdjIndex equal to 1.
mplsLdpHelloAdjHoldTimeRem	The time remaining for this hello adjacency. This interval changes when the next hello message, which corresponds to this hello adjacency, is received.
mplsLdpHelloAdjType	This adjacency is the result of a link hello if the value of this object is link(1). Otherwise, this adjacency is a result of a targeted hello and its value is targeted(2).

mplsLdpSessionTable

The table below lists the mplsLdpSessionTable objects and their descriptions.

Table 43: mplsLdpSessionTable Objects and Descriptions

Object	Description
mplsLdpSessionEntry	An entry in this table represents information on a single session between an LDP entity and an LDP peer. The information contained in a row is read-only. This table augments the mplsLdpPeerTable.
mplsLdpSesState	The current state of the session. All of the states are based on the LDP or TDP state machine for session negotiation behavior. The states are as follows: • nonexistent(1) • initialized(2) • openrec(3) • opensent(4) • operational(5)
mplsLdpSesProtocolVersion	The version of the LDP protocol which this session is using. This is the version of the LDP protocol that has been negotiated during session initialization.
mplsLdpSesKeepAliveHoldTimeRem	The keepalive hold time remaining for this session.
mplsLdpSesMaxPduLen	The value of maximum allowable length for LDP PDUs for this session. This value could have been negotiated during the session initialization.
mplsLdpSesDiscontinuityTime	The value of sysUpTime on the most recent occasion when one or more of this session's counters suffered a discontinuity. The relevant counters are the specific instances of any Counter32 or Counter64 object contained in the mplsLdpSesStatsTable associated with this session. The initial value of this object is the value of sysUpTime when the entry was created in this table.

mplsLdpAtmSesTable

The table below lists the mplsLdpAtmSesTable objects and their descriptions.

Table 44: mplsLdpAtmSesTable Objects and Descriptions

Objects	Description
mplsLdpAtmSesEntry	An entry in this table represents information on a single label range intersection between an LDP entity and an LDP peer (not accessible).

Objects	Description
mplsLdpAtmSesLRLowerBoundVpi	The minimum VPI number for this range (not accessible).
mplsLdpAtmSesLRLowerBoundVci	The minimum VCI number for this range (not accessible).
mplsLdpAtmSesLRUpperBoundVpi	The maximum VPI number for this range (read-only).
mplsLdpAtmSesLRUpperBoundVci	The maximum VCI number for this range (read-only).

mplsLdpSesStatsTable

The table below lists the mplsLdpSesStatsTable objects and their descriptions.

Table 45: mplsLdpSesStatsTable Objects and Descriptions

Object	Description
mplsLdpSesStatsEntry	An entry in this table represents statistical information on a single session between an LDP entity and an LDP peer. This table augments the mplsLdpPeerTable.
mplsLdpSesStatsUnkMesTypeErrors	This object is the count of the number of unknown message type errors detected during this session.
mplsLdpSesStatsUnkTlvErrors	This object is the count of the number of unknown TLV errors detected during this session.

VPN Contexts in MPLS LDP MIB Version 8 Upgrade

Within an MPLS Border Gateway Protocol (BGP) 4 Virtual Private Network (VPN) environment, separate LDP processes can be created for each VPN. These processes and their associated data are called LDP contexts. Each context is independent from all others and contains data specific only to that context.

This feature adds support for different contexts for different MPLS VPNs. Users of the MIB can view MPLS LDP processes for a given MPLS VPN. The VPN Aware LDP MIB feature does not change the syntax of the IETF MPLS-LDP MIB. It changes the number and types of entries within the tables.

The IETF MPLS-LDP MIB can show information about only one context at a time. You can specify a context, either a global context or an MPLS VPN context, using an SMNP security name.

The following sections describe topics related to the VPN Aware LDP MIB feature:

SNMP Context

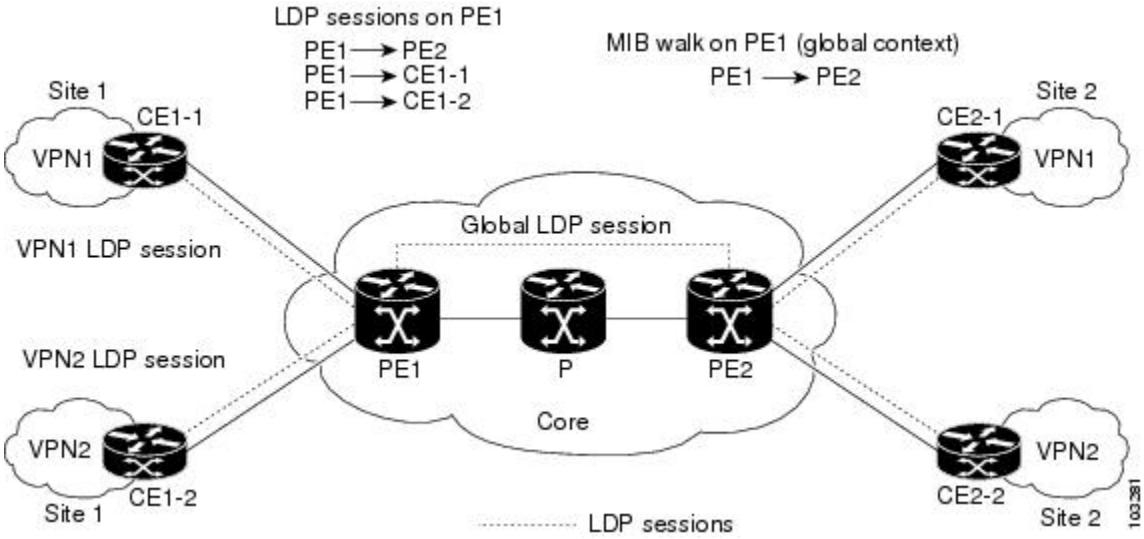
SNMP contexts provide VPN users with a secure way of accessing MIB data. When a VPN is associated with a context, that VPN's specific MIB data exists in that context. Associating a VPN with a context enables service providers to manage networks with multiple VPNs. Creating and associating a context with a VPN enables a provider to prevent the users of one VPN from accessing information about users of other VPNs on the same networking device.

VPN Aware LDP MIB Sessions

Before the VPN Aware LDP MIB features, an SNMP query to the MPLS LDP MIB returned information about global sessions only. A query did not return information about LDP sessions in a VPN context. The IETF MPLS LDP MIB retrieved information from global routing tables, but did not retrieve information from VPN routing and forwarding instances (VRFs) that store per-VPN routing data. The MPLS LDP MIB looked only at LDP processes in the global context and ignored all other sessions. A query on a VRF returned no information. You can view LDP processes in a VPN context.

The figure below shows a sample MPLS VPN network with the MPLS LDP sessions prior to the implementation of the VPN Aware LDP MIB feature.

Figure 29: MPLS LDP Sessions Setup Before VPN Aware LDP MIB Feature



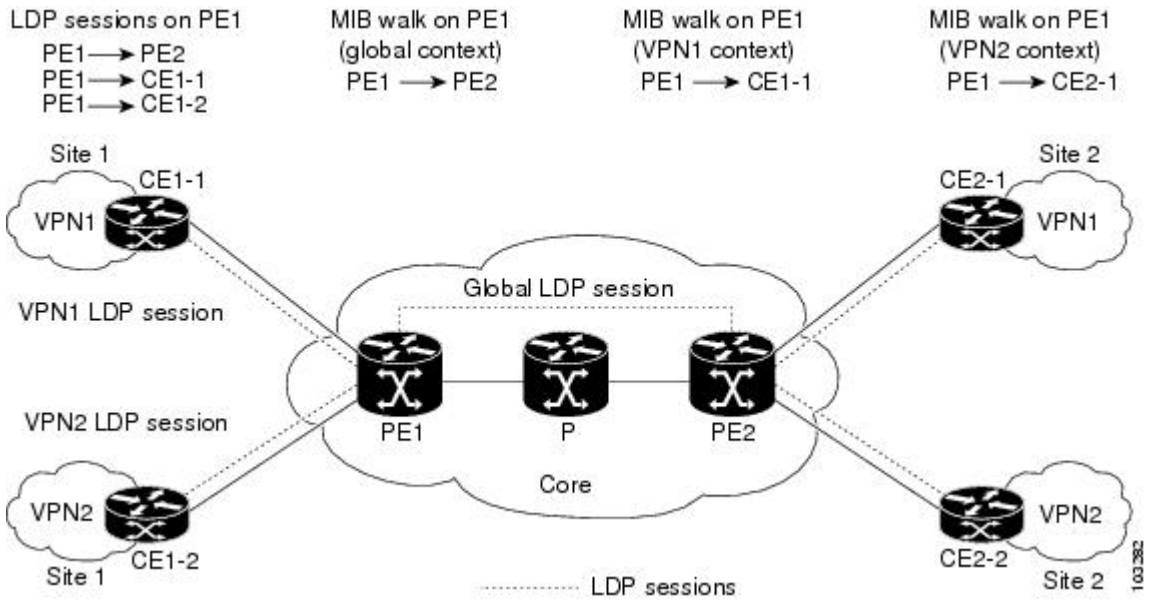
A MIB walk prior to this software release displayed only global session information.

With the VPN Aware LDP MIB enhancement, an SNMP query to the IETF MPLS-LDP-MIB supports both global and VPN contexts. This feature allows you to enter LDP queries on any VRF and on the core (global context). A query can differentiate between LDP sessions from different VPNs. LDP session information for a VPN stays in the context of that VPN. Therefore, the information from one VPN is not available to a user of a different VPN. The VPN Aware update to the LDP MIB also allows you to view LDP processes operating in a Carrier Supporting Carrier (CSC) network.

In an MPLS VPN, a service provider edge router (PE) might contain VRFs for several VPNs as well as a global routing table. To set up separate LDP processes for different VPNs on the same device, you need to configure each VPN with a unique securityName, contextName, and View-based Access Control Model (VACM) view. The VPN securityName must be configured for the IETF MPLS LDP MIB.

The figure below shows LDP sessions for a sample MPLS VPN network with the VPN Aware LDP MIB feature.

Figure 30: MPLS LDP Sessions with the VPN Aware LDP MIB Feature



With the VPN Aware LDP MIB feature, you can do MIB queries or MIB walks for an MPLS VPN LDP session or a global LDP session.

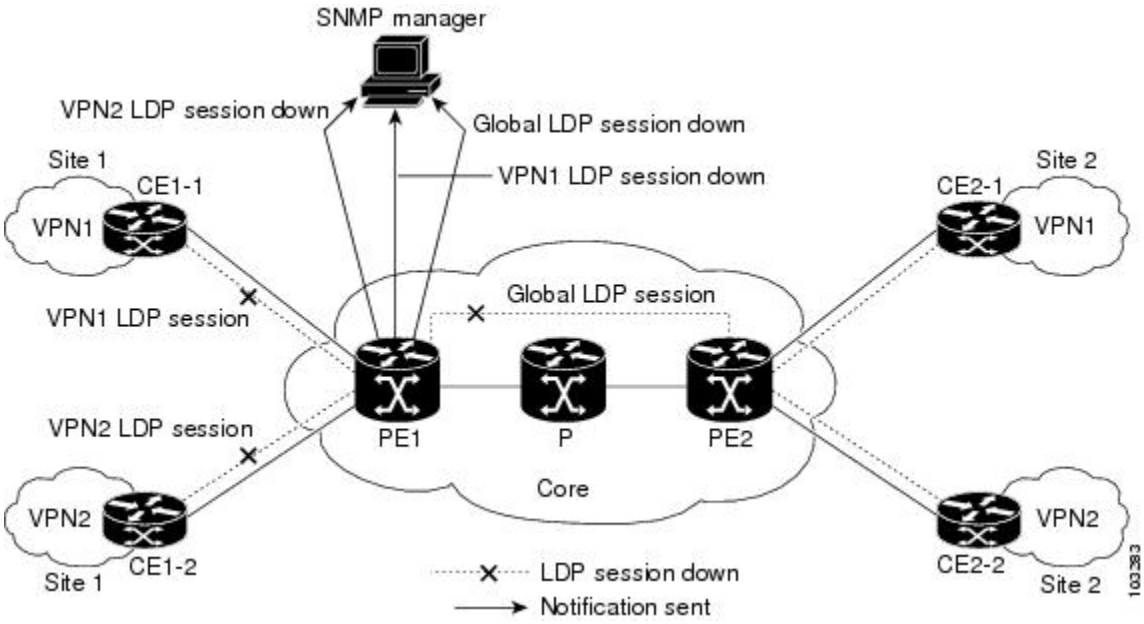
Note To verify LDP session information for a specific VPN, use the **show mpls ldp neighbor vrf** *vpn-name* **detail** command.

VPN Aware LDP MIB Notifications

Before the VPN Aware LDP MIB feature, all notification messages for MPLS LDP sessions were sent to the same designated network management station (NMS) in the network. The notifications were enabled with the **snmp-server enable traps mpls ldp** command.

The figure below shows LDP notifications that were sent before the implementation of the VPN Aware LDP MIB feature.

Figure 31: LDP Notifications Sent Before the VPN Aware LDP MIB Feature



The VPN Aware LDP MIB feature supports LDP notifications for multiple LDP contexts for VPNs. LDP notifications can be generated for the core (global context) and for different VPNs. You can cause notifications be sent to different NMS hosts for different LDP contexts. LDP notifications associated with a specific VRF are sent to the NMS designated for that VRF. LDP global notifications are sent to the NMS configured to receive global traps.

To enable LDP context notifications for the VPN Aware LDP MIB feature, use either the SNMP object mplsLdpSessionsUpDownEnable (in the global LDP context only) or the following extended global configuration commands.

To enable LDP notifications for the global context, use the following commands on a PE router:

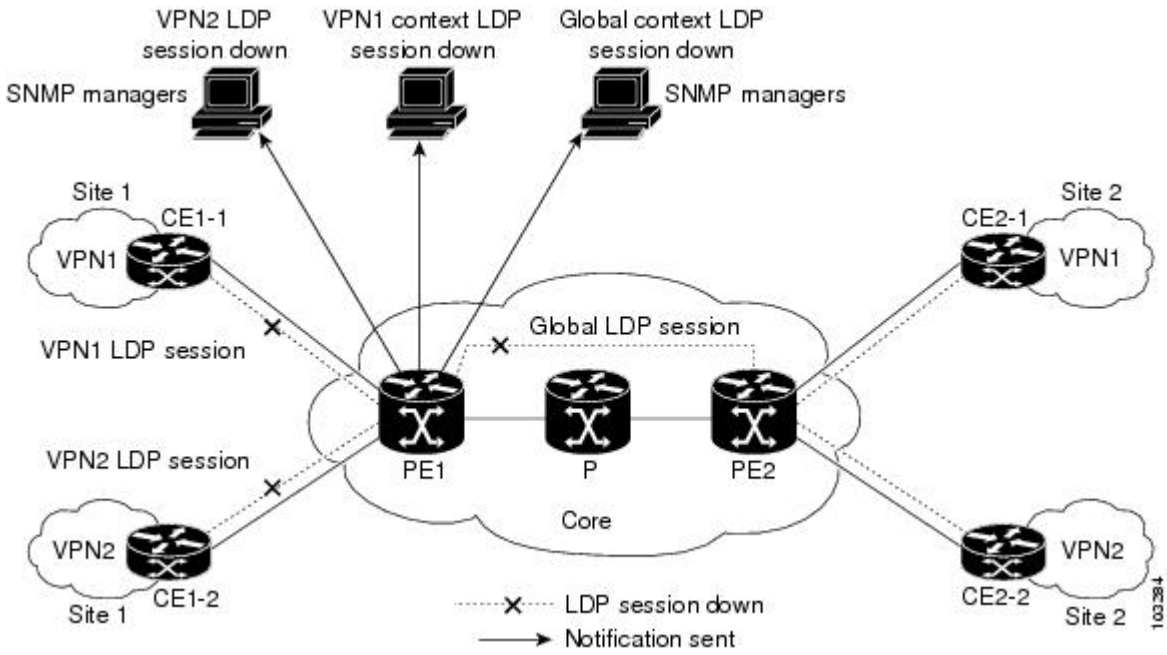
```
Router(config)# snmp-server host host-address traps community mpls-ldp
Router(config)# snmp-server enable traps mpls ldp
```

To enable LDP notifications for a VPN context, use the following commands on a PE router:

```
Router(config)# snmp-server host host-address vrf vrf-name version {v1|v2c|v3}
community community-string udp-port upd-port mpls-ldp
Router(config)# snmp-server enable traps mpls ldp
```

The figure below shows LDP notifications with the VPN Aware LDP MIB feature.

Figure 32: LDP Notifications With the VPN Aware LDP MIB Feature



How to Configure MPLS LDP MIB Version 8 Upgrade

Enabling the SNMP Agent

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** *number*]
5. **end**
6. **write memory**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** *number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) community strings for the MPLS Label Distribution Protocol (LDP) MIB. • The *string* argument functions like a password, permitting access to SNMP functionality on label switch routers (LSRs) in an MPLS network. • The optional **ro** keyword configures read-only (ro) access to the objects in the MPLS LDP MIB.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.
Step 7	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If you see any snmp-server statements, SNMP has been enabled on the router. If any SNMP information is displayed, you can modify the information or change it as desired.

Enabling Distributed Cisco Express Forwarding

Perform this task to enable Cisco Express Forwarding or distributed Cisco Express Forwarding.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling MPLS Globally

Perform this task to enable MPLS globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Router(config)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for the platform.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling LDP Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol** {**ldp** | **tdp**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol** {**ldp**	**tdp**} **Example:** `Router(config)# mpls label protocol ldp`	Specifies the platform default label distribution protocol. TDP might not be supported in all software releases.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling MPLS on an Interface

Perform this task to enable MPLS on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot/port* [**.***subinterface-number*]
4. **mpls ip**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/subslot/port* [**.***subinterface-number*] **Example:** `Router(config)# interface FastEthernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for a particular interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Enabling LDP on an Interface

Perform this task to enable LDP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / subslot / port* [**.** *subinterface-number*]
4. **mpls label protocol ldp**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / subslot / port* [**.** *subinterface-number*] **Example:** `Router(config)# interface FastEthernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**mpls label protocol ldp** **Example:** `Router(config-if)# mpls label protocol ldp`	Specifies the label distribution protocol to be used on a given interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuring a VPN Aware LDP MIB

Configuring SNMP Support for a VPN

Perform this task to configure SNMP support for a Virtual Private Network (VPN) or a remote VPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
4. **snmp-server engineID remote** *ip-address* [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*] **Example:** `Router(config)# snmp-server host example.com vrf trap-vrf`	Specifies the recipient of an SNMP notification operation and specifies the Virtual Private Network (VPN) routing and forwarding (VRF) instance table to be used for the sending of SNMP notifications.
Step 4	**snmp-server engineID remote** *ip-address* [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string* **Example:** `Router(config)# snmp-server engineID remote 172.16.20.3 vrf traps-vrf 80000009030000B064EFE100`	Configures a name for the remote SNMP engine on a router.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring an SNMP Context for a VPN

Perform this task to configure an SNMP context for a VPN. This sets up a unique SNMP context for a VPN, which allows you to access the VPN's LDP session information.

SNMP Context

SNMP contexts provide VPN users with a secure way of accessing MIB data. When a VPN is associated with a context, that VPN's specific MIB data exists in that context. Associating a VPN with a context enables service providers to manage networks with multiple VPNs. Creating and associating a context with a VPN enables a provider to prevent the users of one VPN from accessing information about users of other VPNs on the same networking device.

VPN Route Distinguishers

A route distinguisher (RD) creates routing and forwarding tables for a VPN. Cisco software adds the RD to the beginning of the customer's IPv4 prefixes to change them into globally unique VPN-IPv4 prefixes.

Either the RD is an autonomous system number (ASN)-relative RD, in which case it is composed of an autonomous system number and an arbitrary number, or it is an IP-address-relative RD, in which case it is composed of an IP address and an arbitrary number. You can enter an RD in either of these formats:

- 16-bit ASN: your 32-bit number, for example, 101:3.
- 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server context** *context-name*
4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **context** *context-name*
7. **route-target** [**import** | **export** | **both**] *route-target-ext-community*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server context** *context-name* **Example:** `Router(config)# snmp-server context context1`	Creates and names an SNMP context.

	Command or Action	Purpose
Step 4	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Configures a Virtual Private Network (VPN) routing and forwarding instance (VRF) table and enters VRF configuration mode.
Step 5	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:120`	Creates a VPN route distinguisher.
Step 6	**context** *context-name* **Example:** `Router(config-vrf)# context context1`	Associates an SNMP context with a particular VRF.
Step 7	**route-target** [**import** | **export** | **both**] *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 100:1000`	(Optional) Creates a route-target extended community for a VRF.
Step 8	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Associating an SNMP VPN Context with SNMPv1 or SNMPv2

Perform this task to associate an SNMP VPN context with SNMPv1 or SNMPv2. This allows you to access LDP session information for a VPN using SNMPv1 or SNMPv2.

SNMPv1 or SNMPv2 Security: SNMPv1 and SNMPv2 are not as secure as SNMPv3. SNMP Versions 1 and 2 use plain text communities and do not perform the authentication or security checks that SNMP Version 3 performs.

To configure the VPN Aware LDP MIB feature when using SNMP Version 1 or SNMP Version 2, you need to associate a community name with a VPN. This association causes SNMP to process requests coming in for a particular community string only if they come in from the configured VRF. If the community string contained in the incoming packet does not have an associated VRF, the packet is processed only if it came in through a non-VRF interface. This process prevents users outside the VPN from using a clear text community string to query the VPN data. However, this is not as secure as using SNMPv3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server user** *username group-name* [**remote** *host* [**udp-port** *port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** *access-list*]

4. **snmp-server group** *group-name* {**v1** | **v2c** | **v3**{**auth** | **noauth** | **priv**}} [**context** *context-name*] [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
5. **snmp-server view** *view-name oid-tree* {**included** | **excluded**}
6. **snmp-server enable traps** [*notification-type*]
7. **snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [**notification-type**] [**vrf** *vrf-name*]
8. **snmp mib community-map** *community-name* [**context** *context-name*] [**engineid** *engine-id*] [**security-name** *security-name*] **target-list** *vpn-list-name*
9. **snmp mib target list** *vpn-list-name* {**vrf** *vrf-name* | **host** *ip-address*}
10. **no snmp-server trap authentication vrf**
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server user** *username group-name* [**remote** *host* [**udp-port** *port*]] {**v1**	**v2c**	**v3** [**encrypted**] [**auth** {**md5**	**sha**} *auth-password*]} [**access** *access-list*] **Example:** `Router(config)# snmp-server user customer1 group1 v1`	Configures a new user to an SNMP group.
Step 4	**snmp-server group** *group-name* {**v1**	**v2c**	**v3**{**auth**	**noauth**	**priv**}} [**context** *context-name*] [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] **Example:** `Router(config)# snmp-server group group1 v1 context context1 read view1 write view1 notify view1`	Configures a new SNMP group or a table that maps SNMP users to SNMP views. • Use the **context** *context-name* keyword and argument to associate the specified SNMP group with a configured SNMP context.
Step 5	**snmp-server view** *view-name oid-tree* {**included**	**excluded**} **Example:** `Router(config)# snmp-server view view1 ipForward included`	Creates or updates a view entry.

	Command or Action	Purpose
Step 6	**snmp-server enable traps** [*notification-type*] **Example:** `Router(config)# snmp-server enable traps`	Enables all SNMP notifications (traps or informs) available on your system.
Step 7	**snmp-server host** *host-address* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [**notification-type**] [**vrf** *vrf-name*] **Example:** `Router(config)# snmp-server host 10.0.0.1 vrf customer1 public udp-port 7002`	Specifies the recipient of an SNMP notification operation.
Step 8	**snmp mib community-map** *community-name* [**context** *context-name*] [**engineid** *engine-id*] [**security-name** *security-name*] **target-list** *vpn-list-name* **Example:** `Router(config)# snmp mib community-maps community1 context context1 target-list commAVpn`	Associates an SNMP community with an SNMP context, Engine ID, or security name.
Step 9	**snmp mib target list** *vpn-list-name* {**vrf** *vrf-name* | **host** *ip-address*} **Example:** `Router(config)# snmp mib target list commAVpn vrf vrf1`	Creates a list of target VRFs and hosts to associate with an SNMP community.
Step 10	**no snmp-server trap authentication vrf** **Example:** `Router(config)# no snmp-server trap authentication vrf`	(Optional) Disables all SNMP authentication notifications (traps and informs) generated for packets received on VRF interfaces. • Use this command to disable authentication traps only for those packets on VRF interfaces with incorrect community associations.
Step 11	**exit** **Example:** `Router(config) exit`	Exits to privileged EXEC mode.

Verifying MPLS LDP MIB Version 8 Upgrade

Perform a MIB walk using your SNMP management tool to verify that the MPLS LDP MIB Version 8 Upgrade feature is functioning.

Configuration Examples for MPLS LDP MIB Version 8 Upgrade

MPLS LDP MIB Version 8 Upgrade Examples

The following example shows how to enable an SNMP agent on the host NMS:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# snmp-server community
```

The following example shows how to enable SNMPv1 and SNMPv2C on the host NMS. The configuration permits any SNMP agent to access all MPLS LDP MIB objects that have read-only permission using the community string public.

```
Router(config)# snmp-server community public
```

The following example shows how to allow read-only access to all MPLS LDP MIB objects relating to members of access list 4 that specify the comaccess community string. No other SNMP agents will have access to any of the MPLS LDP MIB objects.

```
Router(config)# snmp-server community comaccess ro 4
```

The following example shows how to enable LDP globally and then on an interface:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mpls label protocol ldp
Router(config)# interface FastEthernet1/0/0
Router(config-if)# mpls label protocol ldp
Router(config-if)# end
```

Configuring a VPN Aware SNMP Context for SNMPv1 or SNMPv2 Example

The following configuration example shows how to configure a VPN Aware SNMP context for the MPLS LDP MIB Version 8 with SNMPv1 or SNMPv2:

```
snmp-server context A
snmp-server context B
ip vrf CustomerA
 rd 100:110
 context A
 route-target export 100:1000
 route-target import 100:1000
!
ip vrf CustomerB
 rd 100:120
 context B
 route-target export 100:2000
 route-target import 100:2000
!
interface FastEthernet0/3/1
 description Belongs to VPN A
 ip vrf forwarding CustomerA
 ip address 10.0.0.0 255.255.0.0
```

```
interface FastEthernet0/3/2
 description Belongs to VPN B
 ip vrf forwarding CustomerB
 ip address 10.0.0.1 255.255.0.0
snmp-server user commA grp1A v1
snmp-server user commA grp2A v2c
snmp-server user commB grp1B v1
snmp-server user commB grp2B v2c
snmp-server group grp1A v1 context A read viewA write viewA notify viewA
snmp-server group grp1B v1 context B read viewB write viewB notify viewB
snmp-server view viewA ipForward included
snmp-server view viewA ciscoPingMIB included
snmp-server view viewB ipForward included
snmp-server view viewB ciscoPingMIB included
snmp-server enable traps
snmp-server host 10.0.0.3 vrf CustomerA commA udp-port 7002
snmp-server host 10.0.0.4 vrf CustomerB commB udp-port 7002
snmp mib community-map  commA context A target-list commAvpn
! Configures source address validation
snmp mib community-map  commB context B target-list commBvpn
! Configures source address validation
snmp mib target list commAvpn vrf CustomerA
! Configures a list of VRFs or from which community commA is valid
snmp mib target list commBvpn vrf CustomerB
! Configures a list of VRFs or from which community commB is valid
```

Additional References

Related Documents

Related Topic	Document Title
MPLS LDP configuration tasks	MPLS Label Distribution Protocol (LDP)
A description of SNMP agent support for the MPLS Traffic Engineering MIB (MPLS TE MIB)	MPLS Traffic Engineering (TE) MIB
A description of MPLS differentiated types of service across an MPLS network	MPLS Quality of Service
SNMP commands	*Network Management Command Reference*
SNMP configuration SNMP Support for VPNs	Configuring SNMP Support

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
• MPLS Label Distribution Protocol MIB (draft-ietf-mpls-ldp-mib-08.txt) • SNMP-VACM-MIB The View-based Access Control Model (ACM) MIB for SNMP	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2233 The LDP implementation supporting the MPLS LDP MIB fully complies with the provisions of Section 10 of RFC 2026, which, in effect, states that the implementation of LDP is recommended for network devices that perform MPLS forwarding along normally routed paths, as determined by destination-based routing protocols.	*Interfaces MIB*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS LDP MIB Version 8 Upgrade

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 46: Feature Information for MPLS LDP MIB Version 8 Upgrade

Feature Name	Releases	Feature Information
MPLS LDP MIB Version 08 Upgrade	12.0(11)ST 12.2(2)T 12.0(21)ST 12.0(22)S 12.0(24)S 12.2(18)S 12.2(33)SRB 12.2(33)SB Cisco IOS XE Release 2.1	The MPLS Label Distribution Protocol (LDP) MIB Version 8 Upgrade feature enhances the LDP MIB to support the Internet Engineering Task Force (IETF) draft Version 8. In Cisco IOS Release 12.0(11)ST, this feature was introduced to provide SNMP agent support for the MPLS LDP MIB on the Cisco 7200, Cisco 7500, and Cisco 12000 series routers. In Cisco IOS Release 12.2(2)T, this feature was added to this release to provide SNMP agent support for the MPLS LDP MIB on Cisco 7200 and Cisco 7500 series routers. In Cisco IOS Release 12.0(21)ST, this feature was added to this release to provide SNMP agent and LDP notification support for the MPLS LDP MIB on Cisco 7200, Cisco 7500, and Cisco 12000 series Internet routers. In Cisco IOS Release 12.0(22)S, Version 1 was integrated into Cisco IOS Release 12.0(22)S. In Cisco IOS Release 12.0(24)S, this feature was upgraded to Version 8 in Cisco IOS Release 12.0(24)S. This feature was integrated into Cisco IOS Release 12.2(18)S. In Cisco IOS Release 12.2(33)SRB, this MIB was deprecated and replaced by MPLS-LDP-STD-MIB (RVC 3815). In Cisco IOS Release 12.2(33)SB, this MIB was deprecated and replaced by MPLS-LDP-STD-MIB (RVC 3815). This feature was integrated into Cisco IOS XE Release 2.1 and implemented on Cisco ASR 1000 Series Aggregation Services Routers.
		The following commands were introduced or modified: **context**, **show mpls ldp neighbor**, **snmp mib community-map**, **snmp mib target list**, **snmp-server community**, **snmp-server context**, **snmp-server enable traps** (MPLS), **snmp-server group**, **snmp-server host**, **snmp-server trap authentication vrf**.
MPLS VPN-VPN Aware LDP MIB	12.0(27)S 12.2(28)SB 12.2(33)SRA 12.2(33)SXH 12.4(20)T	The MPLS VPN-VPN Aware LDP MIB allows you to enter LDP queries on any VRF and on the core (global context). In Cisco IOS Release 12.0(27)S, support for the MPLS VPN-VPN Aware LDP MIB feature was added. In Cisco IOS Release 12.2(28)SB, this feature was integrated. In Cisco IOS Release 12.2(33)SRA, this feature was integrated. In Cisco IOS Release 12.2(33)SXH, this feature was integrated. In Cisco IOS Release 12.4(20)T, this feature was integrated.

Glossary

ATM -- Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3 .

downstream-on-demand distribution--A label distribution method in which a downstream label switch router (LSR) sends a binding upstream only if the upstream LSR requests it.

downstream unsolicited distribution--A label distribution method in which labels are dispersed if a downstream label switch router (LSR) needs to establish a new binding with its neighboring upstream LSR. For example, an edge LSR might enable a new interface with another subnet. The LSR then announces to the upstream router a binding to reach this network.

informs --A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, but a trap notification does not.

label --A short, fixed-length data identifier that tells switching nodes how to forward data (packets or cells).

label distribution--The techniques and processes that are used by label switch routers (LSRs) to exchange label binding information for supporting hop-by-hop forwarding along normally routed paths.

LDP --Label Distribution Protocol. The protocol that supports Multiprotocol Label Switching (MPLS) hop-by-hop forwarding and the distribution of bindings between labels and network prefixes.

LSP --label switched path. A configured connection between two label switch routers (LSRs) in which label-switching techniques are used for packet forwarding; also a specific path through an Multiprotocol Label Switching (MPLS) network.

LSR --label switch router. A Multiprotocol Label Switching (MPLS) node that can forward native Layer 3 packets. The LSR forwards a packet based on the value of a label attached to the packet.

MIB --Management Information Base. A database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP). The value of a MIB object can be changed or retrieved by the use of SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS --Multiprotocol Label Switching. A switching method for the forwarding of IP traffic through the use of a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

MPLS label distribution--A constraint-based routing algorithm for routing label-switched path (LSP) tunnels.

NMS --network management station. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks. In the context of Simple Network Management Protocol (SNMP), an NMS is a device that performs SNMP queries to the SNMP agent of a managed device to retrieve or modify information.

notification --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant network event has occurred. See also trap.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature of the packet streams they want to receive by specifying such items as bandwidth, jitter, and maximum burst.

RTR --Response Time Reporter. A tool that allows you to monitor network performance, network resources, and applications by measuring response times and availability.

SNMP --Simple Network Management Protocol. A network management protocol used almost exclusively in TCP/IP networks. SNMP enables a user to monitor and control network devices, manage configurations, collect statistics, monitor performance, and ensure network security.

SNMP communities--Authentication scheme that enables an intelligent network device to validate SNMP requests.

SNMPv2c --Version 2c of the Simple Network Management Protocol. SNMPv2c supports centralized as well as distributed network management strategies and includes improvements in the Structure of Management Information (SMI), protocol operations, management architecture, and security.

SNMPv3 --Version 3 of the Simple Network Management Protocol. Interoperable standards-based protocol for network management. SNMPv3 provides secure access to devices by a combination of authenticating and encrypting packets over the network.

TLV --Type-Length-Value. A mechanism used by several routing protocols to carry a variety of attributes. Cisco Discovery Protocol (CDP), Label Discovery Protocol (LDP), and Border Gateway Protocol (BGP) are examples of protocols that use TLVs. BGP uses TLVs to carry attributes such as Network Layer Reachability Information (NLRI), Multiple Exit Discriminator (MED), and local preference.

trap --A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant network event has occurred. Traps (notifications) are less reliable than inform requests, because the receiver of the trap does not send an acknowledgment of receipt; furthermore, the sender of the trap cannot determine if the trap was received. See also notification.

VCC --virtual channel connection. A logical circuit, made up of virtual channel links (VCLs), that carries data between two endpoints in an ATM network. Sometimes called a virtual circuit connection.

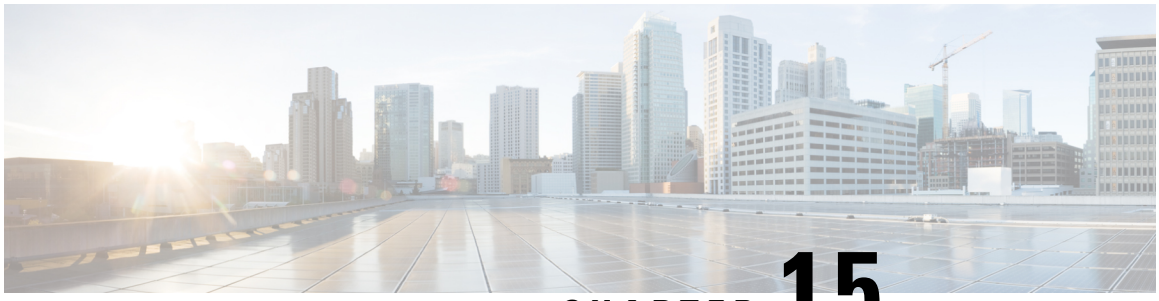
VCI --virtual channel identifier. A 16-bit field in the header of an ATM cell. The VCI, together with the virtual path identifier (VPI), is used to identify the next network virtual channel link (VCL) as the cell passes through a series of ATM switches on its way to its final destination.

VCL --virtual channel link. The logical connection that exists between two adjacent switches in an ATM network.

VPI --virtual path identifier. An 8-bit field in the header of an ATM cell. The VPI, together with the virtual channel identifier (VCI), is used to identify the next network virtual channel link (VCL) as the cell passes through a series of ATM switches on its way to its final destination.

VPN --Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

VRF --VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.



CHAPTER 15

MPLS VPN--MIB Support

This document describes the Simple Network Management Protocol (SNMP) agent support in Cisco software for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) management, as implemented in the draft *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2* (*draft-ietf-ppvpn-mpls-vpn-mib-05.txt*). This document also describes the cMplsNumVrfRouteMaxThreshCleared notification, which is implemented as part of the proprietary MIB CISCO-IETF-PPVNP-MPLS-VPN-MIB.

Prerequisites for MPLS VPN--MIB Support

- SNMP is installed and enabled on the label switching routers.
- MPLS is enabled on the label switching routers.
- Multiprotocol Border Gateway Protocol (BGP) is enabled on the label switching routers.
- Cisco Express Forwarding is enabled on the label switching routers.

Restrictions for MPLS VPN--MIB Support

- Configuration of the MIB using the **snmp set**command is not supported, except for trap-related objects, such as mplsVpnNotificationEnable and mplsVpnVrfSecIllegalLabelRcvThresh.
- The mplsVpnVrfBgpNbrPrefixTable is not supported.

Information About MPLS VPN--MIB Support

MPLS VPN Overview

The MPLS VPN technology allows service providers to offer intranet and extranet VPN services that directly connect their customers' remote offices to a public network with the same security and service levels that a private network offers. Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF is created for each VPN defined on a router and contains most of the information needed to manage and monitor MPLS VPNs: an IP routing table, a derived Cisco Express Forwarding table, a set of interfaces that use this forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

MPLS VPN MIB Overview

The Provider-Provisioned VPN (PPVPN)-MPLS-VPN MIB provides access to MPLS VRF information, and interfaces included in the VRF, and other configuration and monitoring information.

The PPVPN-MPLS-VPN MIB provides the following benefits:

- A standards-based SNMP interface for retrieving information about critical MPLS VPN events.
- VRF information to assist in the management and monitoring of MPLS VPNs.
- Information, in conjunction with the Interfaces MIB, about interfaces assigned to VRFs.
- Performance statistics for all VRFs on a router.
- The generation and queueing of notifications that call attention to major changes in the operational status of MPLS VPN enabled interfaces; the forwarding of notification messages to a designated network management system (NMS) for evaluation and action by network administrators.
- Advanced warning when VPN routing tables are approaching or exceed their capacity.
- Warnings about the reception of illegal labels on a VRF-enabled interface. Such receptions may indicate misconfiguration or an attempt to violate security.

This document also describes the CISCO-IETF-PPVPN-MPLS-VPN-MIB, which contains the cMplsNumVrfRouteMaxThreshCleared notification.

MPLS VPN MIB and the IETF

SNMP agent code operating with the PPVPN-MPLS-VPN MIB enables a standardized, SNMP-based approach to managing MPLS VPNs in Cisco software.

The PPVPN-MPLS-VPN MIB is based on the Internet Engineering Task Force draft MIB specification *draft-ietf-ppvpn-mpls-vpn-mib-05.txt* , which includes objects describing features that support MPLS VPN events. This IETF draft MIB, which undergoes revisions from time to time, is becoming a standard. Accordingly, the Cisco implementation of the PPVPN-MPLS-VPN MIB is expected to track the evolution of the IETF draft MIB, and may change accordingly.

Some slight differences between the IETF draft MIB and the actual implementation of MPLS VPNs within Cisco software require some minor translations between the PPVPN-MPLS-VPN MIB and the internal data

structures of Cisco software. These translations are accomplished by means of the SNMP agent code. Also, while running as a low priority process, the SNMP agent provides a management interface to Cisco software. SNMP adds little overhead on the normal functions of the device.

The SNMP objects defined in the PPVPN-MPLS-VPN MIB can be viewed by any standard SNMP utility. The network administrator can retrieve information in the PPVPN-MPLS-VPN MIB using standard SNMP get and getnext operations for SNMP v1, v2, and v3.

All PPVPN-MPLS-VPN MIB objects are based on the IETF draft MIB; thus, no Cisco-specific SNMP application is required to support the functions and operations pertaining to the PPVPN-MPLS-VPN MIB features.

Capabilities Supported by PPVPN-MPLS-VPN MIB

The PPVPN-MPLS-VPN MIB provides you with the ability to do the following:

- Gather routing and forwarding information for MPLS VPNs on a router.
- Expose information in the VRF routing table.
- Gather information on BGP configuration related to VPNs and VRF interfaces and statistics.
- Emit notification messages that signal changes when critical MPLS VPN events occur.
- Enable, disable, and configure notification messages for MPLS VPN events by using extensions to existing SNMP command-line interface (CLI) commands.
- Specify the IP address of NMS in the operating environment to which notification messages are sent.
- Write notification configurations into nonvolatile memory.

Functional Structure of the PPVPN-MPLS-VPN MIB

The SNMP agent code supporting the PPVPN-MPLS-VPN MIB follows the existing model for such code in Cisco software and is, in part, generated by the Cisco software tool set, based on the MIB source code.

The SNMP agent code, which has a layered structure that is common to MIB support code in Cisco software, consists of four layers:

- Platform-independent layer--This layer is generated primarily by the MIB development Cisco software tool set and incorporates platform- and implementation-independent functions. The Cisco MIB development tool set creates a standard set of files associated with a MIB.
- Application interface layer--The functions, names, and template code for MIB objects in this layer are also generated by the MIB development Cisco software tool set.
- Application-specific layer--This layer provides an interface between the application interface layer and the API and data structures layer below and performs tasks needed to retrieve required information from Cisco software, such as searching through data structures.
- API and data structures layer--This layer contains the data structures or APIs within Cisco software that are retrieved or called in order to set or retrieve SNMP management information.

Supported Objects in PPVPN-MPLS-VPN MIB

The PPVPN-MPLS-VPN MIB contains numerous tables and object definitions that provide read-only SNMP management support for the MPLS VPN feature in Cisco IOS software. The PPVPN-MPLS-VPN MIB conforms to Abstract Syntax Notation One (ASN.1), thus reflecting an idealized MPLS VPN database.

Using any standard SNMP network management application, you can retrieve and display information from the PPVPN-MPLS-VPN MIB using GET operations; similarly, you can traverse information in the MIB database for display using GETNEXT operations.

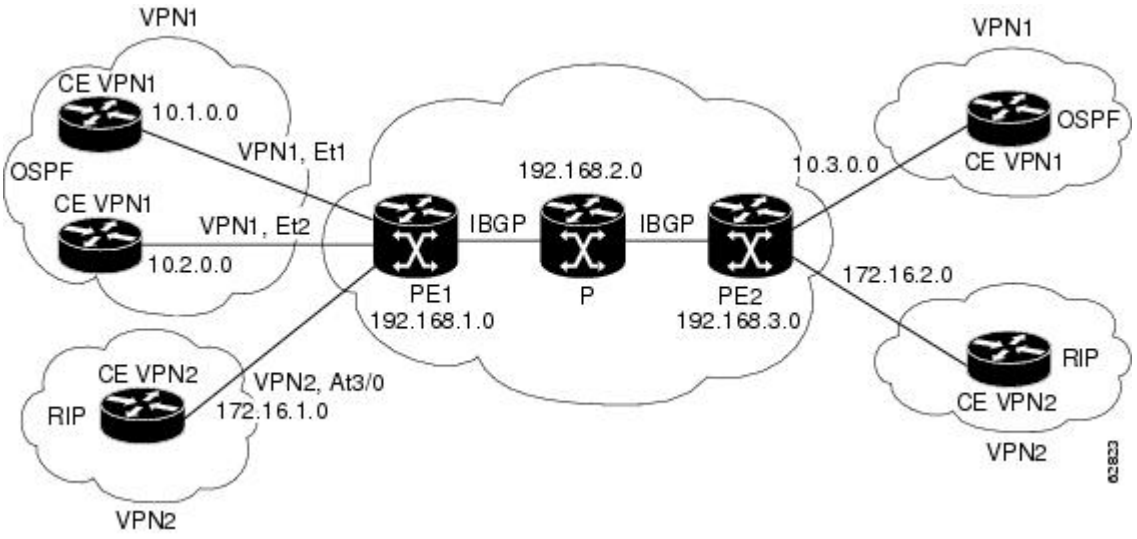
The PPVPN-MPLS-VPN MIB tables and objects are described briefly in the following sections:

The figure below shows a simple MPLS VPN configuration. This configuration includes two customer MPLS VPNs, labeled VPN1 and VPN2, and a simple provider network that consists of two provider edge (PE) routers, labeled PE1 and PE2, and a provider core router labeled P. The figure below shows the following sample configuration:

- VRF names--VPN1 and VPN2
- Interfaces associated with VRFs--Et1, Et2, and At3/0
- Routing protocols--Open Shortest Path First. Link-state (OSPF), Routing Information Protocol (RIP), and internal Border Gateway Protocol (IBGP)
- Routes associated with VPN1--10.1.0.0, 10.2.0.0, and 10.3.0.0
- Routes associated with VPN2--172.16.1.0 and 172.16.2.0
- Routes associated with the provider network--192.168.1.0, 192.168.2.0, and 192.168.3.0

This configuration is used in this document to explain MPLS VPN events that are monitored and managed by the PPVPN-MPLS-VPN MIB.

Figure 33: Sample MPLS VPN Configuration



Scalar Objects

The table below shows the supported PPVPN-MPLS-VPN MIB scalar objects.

Table 47: PPVPN-MPLS-VPN MIB Scalar Objects

MIB Object	Function
mplsVpnConfiguredVrfs	The number of VRFs configured on the router, including VRFs recently deleted.
mplsVpnActiveVrfs	The number of VRFs that are active on the router. An active VRF is assigned to at least one interface that is in the operationally up state.
mplsVpnConnectedInterfaces	The total number of interfaces assigned to any VRF.
mplsVpnNotificationEnable	A value that indicates whether all the PPVPN-MPLS-VPN MIB notifications are enabled: • Setting this object to true enables all notifications defined in the PPVPN-MPLS-VPN MIB. • Setting this object to false disables all notifications defined in the MIB. This is one of the few objects that is writable.
mplsVpnVrfConfMaxPossibleRoutes	A number that indicates the amount of routes that this router is capable of storing. This value cannot be determined because it is based on the amount of available memory in the system. Therefore, this object is set to zero (0).

MIB Tables

The PPVPN-MPLS-VPN MIB implementation supports the following tables described in this section:

mplsVpnVrfTable

Each VRF is referenced by its VRF name (mplsVpnVrfName). The table below lists the MIB objects and their functions for this table.

Table 48: PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfTable

MIB Object	Function
mplsVpnVrfName	The name associated with this VRF. When this object is used as an index to a table, the first octet is the string length, and subsequent octets are the ASCII codes of each character. For example, “vpn1” is represented as 4.118.112.110.49.

mplsVpnVrfDescription	The description of the VRF. This is specified with the following configuration command:
	`Router(config)# ip vrf`
	` vrf-name`
	`Router(config-vrf)# description`
	` vrf-description`

mplsVpnVrfRouteDistinguisher	The route distinguisher for this VRF. This is specified with the following configuration command:
	`Router(config)# ip vrf`
	`vrf-name`
	`Router(config-vrf)# rd`
	` route-distinguisher`

MIB Object	Function
mplsVpnVrfCreationTime	The value of the sysUpTime when this VRF entry was created.
mplsVpnVrfOperStatus	The operational status of this VRF. A VRF is up (1) when at least one interface associated with the VRF is up. A VRF is down (2) when: • No interfaces exist whose ifOperStatus = up (1). • No interfaces are associated with this VRF.
mplsVpnVrfActiveInterfaces	The number of interfaces assigned to this VRF that are operationally up.
mplsVpnVrfAssociatedInterfaces	The number of interfaces assigned to this VRF, independent of the operational status.

mplsVpnVrfConfMidRouteThreshold	The middle route threshold. If the amount of routes in the VRF crosses this threshold, an mplsNumVrfRouteMidThreshExceeded notification is sent (if notifications are enabled and configured). You can set this value in configuration mode as a percentage of the maximum with the **maximum routes** *limit* {*warn-threshold* | **warn-only**} command, as follows:
	`Router(config)# ip vrf vpn1`
	`Router(config-vrf)# maximum routes 1000 50`
	The middle or warn threshold is set for VRF vpn1 as 50 percent of the maximum route threshold.
	The following command sets a middle threshold of 1000 routes. An mplsNumVrfRouteMidThreshExceeded notification is sent when this threshold is exceeded. However, additional routes are still allowed because a maximum route threshold is not set with this command.
	`Router(config-vrf)# maximum routes 1000 warn-only`

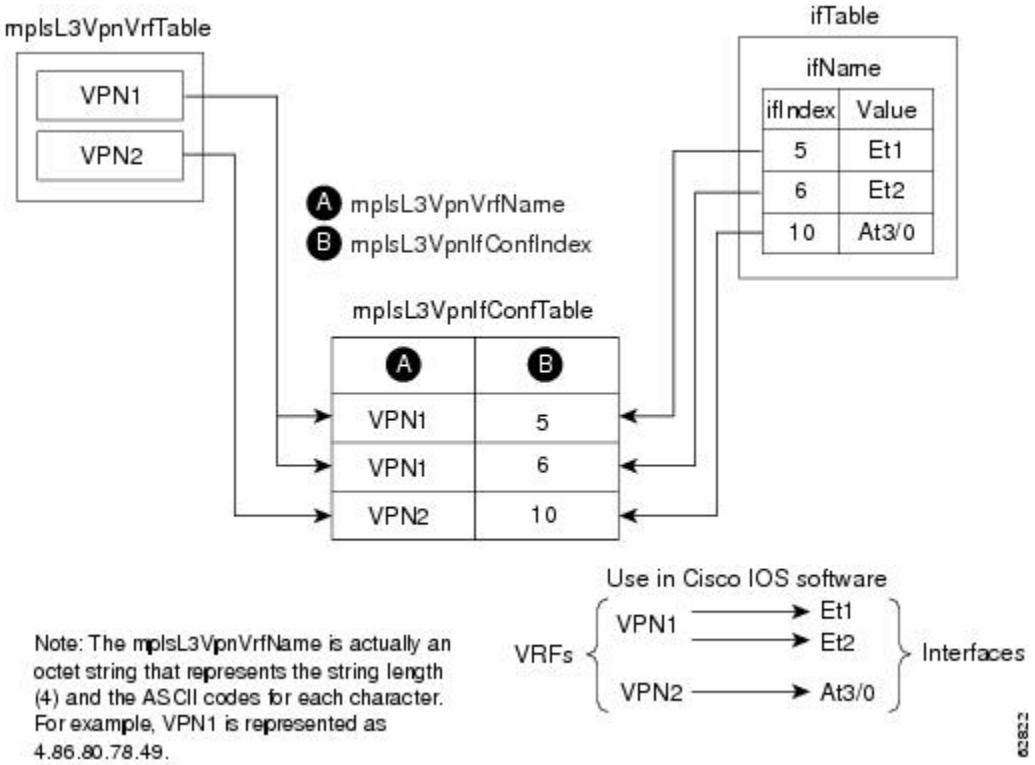
mplsVpnVrfConfHighRouteThreshold	The maximum route threshold. If the number of routes in the VRF crosses this threshold, an mplsNumVrfRouteMaxThreshExceeded notification is sent (if notifications are enabled and configured). You can set this value in configuration mode with the **maximum routes** *limit* {*warn-threshold* | **warn-only**} command as follows:
	`Router(config)# ip vrf vpn2`
	`Router(config-vrf)# maximum routes 1000 75`
	The maximum route threshold is set for 1000 routes for VRF vpn2 with a middle or warn threshold of 75 percent of this threshold.

mplsVpnVrfConfMaxRoutes	This value is the same as the mplsVpnVrfConfHighRouteThreshold.
mplsVpnVrfConfLastChanged	The value of sysUpTime when the configuration of the VRF changes or interfaces are assigned or unassigned from the VRF. **Note** This object is updated only when values in this table change.
mplsVpnVrfConfRowStatus	Read-only implementation. This object normally reads “active (1),” but may read “notInService (2),” if a VRF was recently deleted.
mplsVpnVrfConfStorageType	Read-only implementation. This object always reads “volatile (2).”

mplsVpnInterfaceConfTable

In Cisco software, a VRF is associated with one MPLS VPN. Zero or more interfaces can be associated with a VRF. A VRF uses an interface that is defined in the ifTable of the Interfaces Group of MIB II (IFMIB). The IFMIB defines objects for managing interfaces. The ifTable of this MIB contains information on each interface in the network. The mplsVpnInterfaceConfTable associates a VRF from the mplsVpnVrfTable with a forwarding interface from the ifTable. The figure below shows the relationship between VRFs and interfaces defined in the ifTable and the mplsVpnInterfaceConfTable.

Figure 34: VRFs, the Interfaces MIB, and the mplsVpnInterfaceConfTable



Entries in the VPN interface configuration table (mplsVpnInterfaceConfTable) represent the interfaces that are assigned to each VRF. The information available in this table is also displayed with the **show ip vrf** command.

The mplsVpnInterfaceConfTable shows how interfaces are assigned to VRFs. A label switch router (LSR) creates an entry in this table for every interface capable of supporting MPLS VPNs.

The mplsVpnInterfaceConfTable is indexed by the following:

- mplsVpnVrfName--The VRF name
- mplsVpnInterfaceConfIndex--An identifier that is the same as the ifIndex from the Interface MIB of the interface assigned to the VRF

The table below lists the MIB objects and their functions for this table.

Table 49: PPVPN-MPLS-VPN MIB Objects for the mplsVpnInterfaceConfTable

MIB Object	Function
mplsVpnInterfaceConfIndex	Provides the interface MIB ifIndex of this interface that is assigned to a VRF.
mplsVpnInterfaceLabelEdgeType	Indicates whether the interface is a provider edge interface (1) or a customer edge interface (2). This value is always providerEdge (1) because in Cisco IOS, customerEdge interfaces are not assigned to VRFs and do not appear in this table.
mplsVpnInterfaceVpnClassification	Specifies what type of VPN this interface is providing: carrier supporting carrier (CsC) (1), enterprise (2), or InterProvider (3). This value is set to enterprise (2) if MPLS is not enabled and to carrier supporting carrier (1) if MPLS is enabled on this interface.
mplsVpnInterfaceVpnRouteDistProtocol	Indicates the route distribution protocols that are being used to redistribute routes with BGP on this interface: BGP (2), OSPF (3), or RIP (4). In Cisco software, router processes are defined and redistributed on a per-VRF basis, not per-interface. Therefore, all interfaces assigned to the same VRF have the same value for this object.
mplsVpnInterfaceConfStorageType	Read-only implementation. This object always reads “volatile (2).”
mplsVpnInterfaceConfRowStatus	Read-only implementation. This object normally reads “active (1),” but may read “notInService (2),” if a VRF was recently deleted.

mplsVpnVrfRouteTargetTable

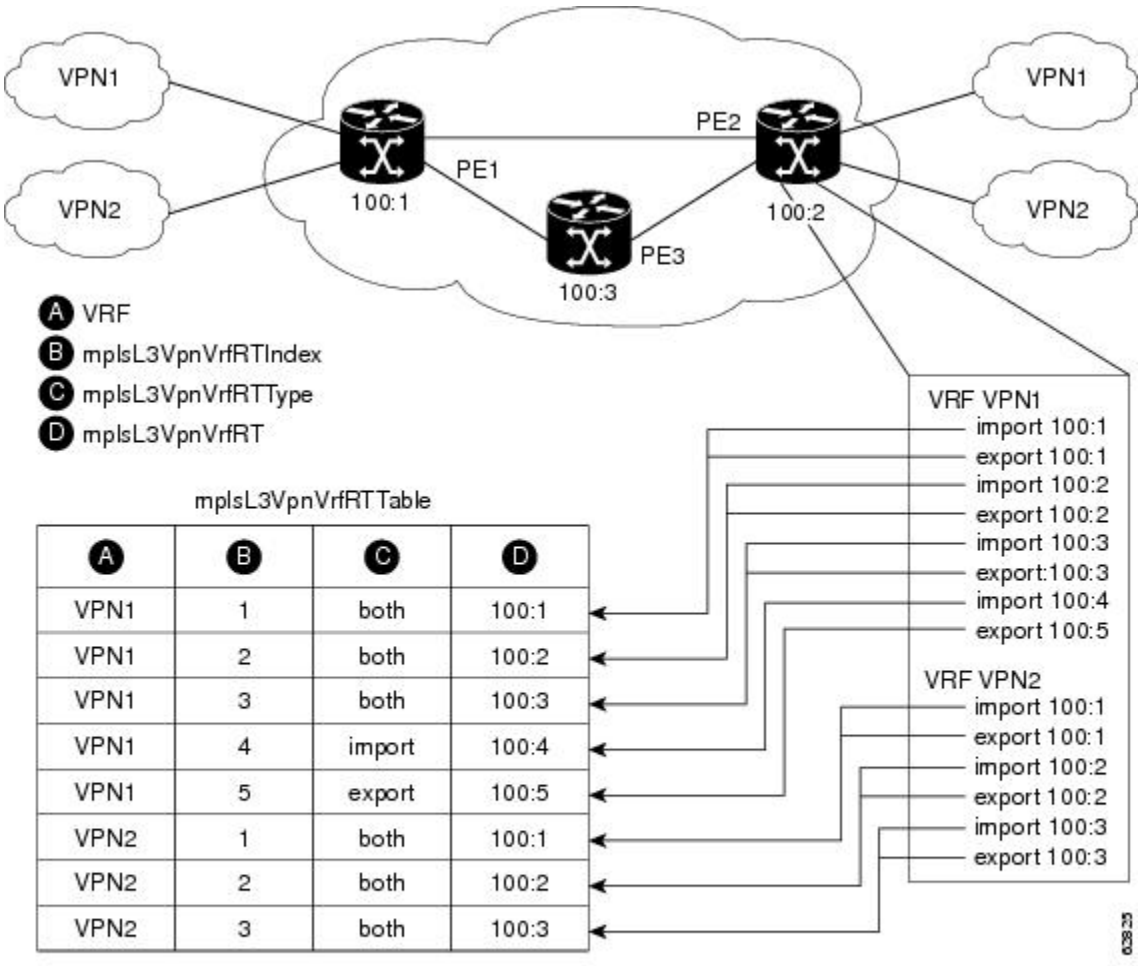
The route target table (mplsVpnVrfRouteTargetTable) describes the route target communities that are defined for a particular VRF. An LSR creates an entry in this table for each target configured for a VRF supporting an MPLS VPN instance.

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by BGP extended communities. Distribution of VPN routing information works as follows:

- When a VPN route learned from a customer edge (CE) router is injected into BGP, a list of VPN route target extended community attributes is associated with it. Typically the list of route target community values is set from an export list of route targets associated with the VRF from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes a route must have for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target communities A, B, and C, then any VPN route that carries any of those route target extended communities--A, B, or C--is imported into the VRF.

The figure below shows a sample configuration and its relationship to an mplsVpnVrfRouteTargetTable. A route target table exists on each PE router. Routers with route distinguishers (RDs) 100:1, 100:2, and 100:3 are shown in the sample configuration. Routers with RDs 100:4 and 100:5 are not shown in the figure, but are included in the route targets for PE2 and in the mplsVpnVrfRouteTargetTable.

Figure 35: Sample Configuration and the mplsVpnVrfRouteTargetTable



mplsL3VpnVrfRTTable

A	B	C	D
VPN1	1	both	100:1
VPN1	2	both	100:2
VPN1	3	both	100:3
VPN1	4	import	100:4
VPN1	5	export	100:5
VPN2	1	both	100:1
VPN2	2	both	100:2
VPN2	3	both	100:3

Note: The mplsL3VpnVrfName is actually an octet string that represents the string length (4) and the ASCII codes for each character. For example, VPN1 is represented as 4.86.80.78.49.

The mplsVpnVrfRouteTargetTable shows the import and export route targets for each VRF. The table is indexed by the following:

- mplsVpnVrfName--The VRF name
- mplsVpnVrfRouteTargetIndex--The route target entry identifier
- mplsVpnVrfRouteTargetType--A value specifying whether the entry is an import route target, export route target, or is defined as both

The table below lists the MIB objects and their functions for this table.

Table 50: PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfRouteTargetTable

MIB Object	Function
mplsVpnVrfRouteTargetIndex	A value that defines each route target's position in the table.
mplsVpnVrfRouteTargetType	Determines which type of route target the entry represents: import (1), export (2), or both (3).
mplsVpnVrfRouteTarget	Determines the route distinguisher for this target.
mplsVpnVrfRouteTargetDescr	Description of the route target. This object is not supported. Therefore, the object is the same as mplsVpnVrfRouteTarget.
mplsVpnVrfRouteTargetRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)," if a VRF was recently deleted.

mplsVpnVrfBgpNbrAddrTable

The BGP neighbor address table (mplsVpnVrfBgpNbrAddrTable) represents the MPLS external Border Gateway Protocol (eBGP) neighbors that are defined for a particular VRF. An LSR creates an entry for every BGP neighbor that is defined in the VRF's address-family.

The mplsVpnVrfBgpNbrAddrTable is indexed by the following:

- mplsVpnVrfName--The VRF name
- mplsVpnInterfaceConfIndex--An identifier that is the same as the ifIndex from the Interface MIB of the interface assigned to the VRF
- mplsVpnVrfBgpNbrIndex--The IP address of the neighbor

The table below lists the MIB objects and their functions for this table.

Table 51: PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfBgpNbrAddrTable

MIB Object	Function
mplsVpnVrfBgpNbrIndex	The IPv4 address of the eBGP neighbor.
mplsVpnVrfBgpNbrRole	The role of this eBGP neighbor: customer edge (1) or provider edge (2). If the object mplsVpnInterfaceVpnClassification is CSC, then this value is provider edge (2); otherwise, this value is customer edge (1).
mplsVpnVrfBgpNbrType	Address type of this eBGP neighbor. The MIB supports only IPv4 (1). Therefore, this object returns "ipv4 (1)."
mplsVpnVrfBgpNbrAddr	IP address of the eBGP neighbor.
mplsVpnVrfBgpNbrRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)" if a VRF was recently deleted.
mplsVpnVrfBgpNbrStorageType	Read-only implementation. This object always reads "volatile (2)."

mplsVpnVrfSecTable

The VRF security table (mplsVpnVrfSecTable) provides information about security for each VRF. An LSR creates an entry in this table for every VRF capable of supporting MPLS VPN.

The mplsVpnVrfSecTable *augments* the mplsVpnVrfTable and has the same indexing.

The table below lists the MIB objects and their functions for this table.

Table 52: PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfSecTable

MIB Object	Function
mplsVpnVrfSecIllegalLabelViolations	The number of illegally received labels on a VRF interface. Only illegal labels are counted by this object, therefore the object only applies to a VRF interface that is MPLS enabled (CSC situation). This counter is incremented whenever a label is received that is above or below the valid label range, not in the global label forwarding table, or is received on the wrong VRF (that is, table IDs for the receiving interface and appropriate VRF label forwarding table do not match).
mplsVpnVrfSecIllegalLabelRcvThresh	Notification threshold for illegal labels received on this VRF. When the number of illegal labels received on this interface crosses this threshold, an mplsNumVrfSecIllegalLabelThreshExceeded notification is sent (if the notification is enabled and configured). This object is one of the few in this MIB agent that supports the SNMP SET operation, which allows you to change this value.

mplsVpnVrfPerfTable

The VRF performance table (mplsVpnVrfPerfTable) provides statistical performance information for each VRF. An LSR creates an entry in this table for every VRF capable of supporting MPLS VPN.

The mplsVpnVrfPerfTable *augments* the mplsVpnVrfTable and has the same indexing.

The table below lists the MIB objects and their functions for this table.

Table 53: PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfPerfTable

MIB Objects	Functions
mplsVpnVrfPerfRoutesAdded	The number of routes added to this VRF over the course of its lifetime.
mplsVpnVrfPerfRoutesDeleted	The number of routes removed from this VRF.
mplsVpnVrfPerfCurrNumRoutes	The number of routes currently defined within this VRF.

mplsVpnVrfRouteTable

The VRF routing table (mplsVpnVrfRouteTable) provides the IP routing table information for each VRF. The information available in this table can also be accessed with the **show ip route vrf** *vrf-name* command. For example, for PE1 in the figure above:

- With the **show ip route vrf vpn1** command, you would see results like the following:

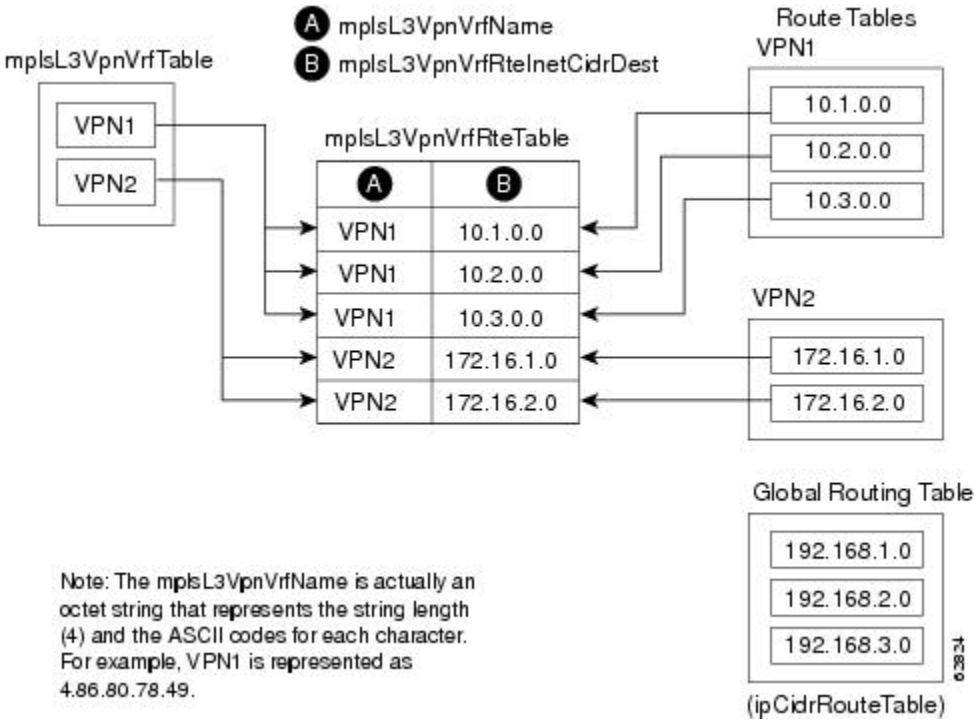
```
Router# show ip route vrf vpn1
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
!
Gateway of last resort is not set
!
     10.0.0.0/32 is subnetted, 3 subnets
B       10.3.0.0 [200/0] via 192.168.2.1, 04:36:33
C       10.1.0.0/16 is directly connected, FastEthernet1
C       10.2.0.0/16 [200/0] directly connected FastEthernet2, 04:36:33
```

- With the **show ip route vrf vpn2** command, you would see results like the following:

```
Router# show ip route vrf vpn2
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
!
Gateway of last resort is not set
!
     172.16.0.0/32 is subnetted, 2 subnets
B       172.16.2.0 [200/0] via 192.168.2.1, 04:36:33
C       172.16.1.0 is directly connected, ATM 3/0
```

The figure below shows the relationship of the routing tables, the VRFs, and the mplsVpnVrfRouteTable. You can display information about the VPN1 and VPN2 route tables using the **show ip route vrf** *vrf-name* command. The global route table is the same as ipCidrRouteTable in the IP-FORWARD-MIB. You can display information about the global route table with the **show ip route** command.

Figure 36: Route Table, VRFs, and the mplsVpnVrfRouteTable



An LSR creates an entry in this table for every route that is configured, either dynamically or statically, within the context of a specific VRF capable of supporting MPLS VPN.

The mplsVpnVrfRouteTable is indexed by the following:

- mplsVpnVrfName--The VRF name, which provides the VRF routing context
- mplsVpnVrfRouteDest--The IP destination address
- mplsVpnVrfRouteMask--The IP destination mask
- mplsVpnVrfRouteTos--The IP header ToS bits
- mplsVpnVrfRouteNextHop--The IP address of the next hop for each route entry

Note The ToS bits are not supported and, therefore, are always 0.

The table below lists the MIB objects and their functions for the mplsVpnVrfRouteTable. This table represents VRF-specific routes. The global routing table is the ipCidrRouteTable in the IP-FORWARD-MIB.

Table 54: PPVPN-MPLS-VPN MIB Objects for the mplsVpnVrfRouteTable

MIB Object	Function
mplsVpnVrfRouteDest	The destination IP address defined for this route.

MIB Object	Function
mplsVpnVrfRouteDestAddrType	The address type of the IP destination address (mplsVpnVrfRouteDest). This MIB implementation supports only IPv4 (1). Therefore, this object has a value of "ipv4 (1)."
mplsVpnVrfRouteMask	The destination IP address mask defined for this route.
mplsVpnVrfRouteMaskAddrType	The address type of the destination IP address mask. This MIB implementation supports only IPv4 (1). Therefore, this object has a value of "ipv4 (1)."
mplsVpnVrfRouteTos	The ToS bits from the IP header for this route. Cisco software supports only ToS bits of zero. Therefore, the object is always 0.
mplsVpnVrfRouteNextHop	The next hop IP address defined for this route.
mplsVpnVrfRouteNextHopAddrType	The address type of the next hop IP address. This MIB implementation only supports only IPv4 (1). Therefore, this object has a value of "ipv4 (1)."
mplsVpnVrfRouteIfIndex	The interface MIB ifIndex for the interface through which this route is forwarded. The object is 0 if no interface is defined for the route.
mplsVpnVrfRouteType	Defines if this route is a local or remotely defined route.
mplsVpnVrfRouteProto	The routing protocol that was responsible for adding this route to the VRF.
mplsVpnVrfRouteAge	The number of seconds since this route was last updated.
mplsVpnVrfRouteInfo	A pointer to more information from other MIBs. This object is not supported and always returns "nullOID (0.0)."
mplsVpnVrfRouteNextHopAS	The autonomous system number of the next hop for this route. This object is not supported and is always 0.
mplsVpnVrfRouteMetric1	The primary routing metric used for this route.
mplsVpnVrfRouteMetric2 mplsVpnVrfRouteMetric3 mplsVpnVrfRouteMetric4 mplsVpnVrfRouteMetric5	Alternate routing metrics used for this route. These objects are supported only for Cisco Interior Gateway Routing Protocol (IGRP) and Cisco Enhanced Interior Gateway Routing Protocol (EIGRP). These objects display the bandwidth metrics used for the route. Otherwise, these values are set to -1.
mplsVpnVrfRouteRowStatus	Read-only implementation. This object normally reads "active (1)," but may read "notInService (2)," if a VRF was recently deleted.
mplsVpnVrfRouteStorageType	Read-only implementation. This object always reads "volatile (2)."

PPVPN-MPLS-VPN MIB Notifications

This section provides the following information about supported PPVPN-MPLS-VPN MIB notifications:

PPVPN-MPLS-VPN MIB Notification Events

The following notifications of the PPVPN-MPLS-VPN MIB are supported:

- mplsVrfIfUp--Sent to an NMS when an interface comes up and is assigned a VRF instance.

- mplsVrfIfDown--Generated and sent to the NMS when a VRF is removed from an interface or the interface transitions from an operationally "up" state to a "down" state.
- mplsNumVrfRouteMidThreshExceeded--Generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the following commands:

```
Router(config)# ip vrf vrf-name
Router(config-vrf)#  maximum routes limit warn-threshold (% of max)
```

The *warn-threshold* argument is a percentage of the maximum routes specified by the *limit* argument. You can also configure a middle threshold with the following command, in which the *limit* argument represents the warning threshold:

```
Router(config-vrf)# maximum routes limit warn-threshold (% of max)
```

This notification is sent to the NMS only at the time the threshold is exceeded. (See the figure below for a comparison of the warning and maximum thresholds.) Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.

- MplsNumVrfRouteMaxThreshExceeded--Generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the *limit* argument of the **maximum routes**c ommands:

```
Router(config)# ip vrf vrf-name
Router(config-vrfmaximum routes limit warn-threshold (% of max)
```

A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another MplsNumVrfRouteMaxThreshExceeded notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. (See the figure below for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *limit warn-threshold* command. Prior to implementation of the PPVPN-MPLS-VPN MIB, you were not notified when this threshold (or the warning threshold) was reached.

- mplsNumVrfSecIllegalLabelThreshExceeded--Generated and sent when the number of illegal labels received on a VRF interface exceeds the threshold *mplsVpnVrfSecIllegalLabelRcvThresh* . This threshold is defined with a value of 0. Therefore, a notification is sent when the first illegal label is received on a VRF. Labels are considered illegal if they are outside of the valid label range, do not have a Label Forwarding Information Base (LFIB) entry, or the table ID of the message does not match the table ID for the label in the LFIB.

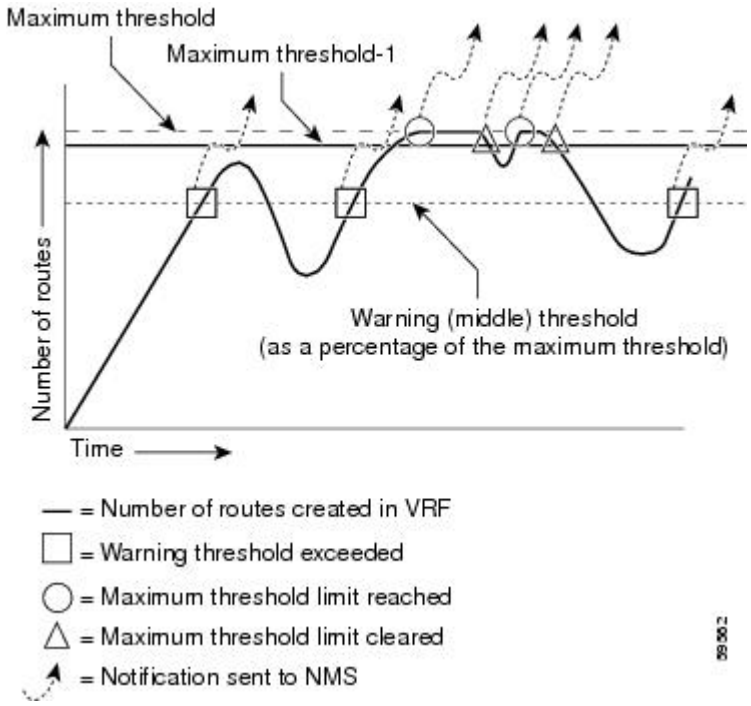
CISCO-IETF-PPVPN-MPLS-VPN MIB Notification Events

The following notification of the CISCO-IETF-PPVPN-MPLS-VPN MIB is supported in Cisco software:

- cMplsNumVrfRouteMaxThreshCleared--Generated and sent when the number of routes on a VRF attempts to exceed the maximum number of routes and then drops below the maximum number of routes. If you attempt to create a route on a VRF that already contains the maximum number of routes, the mplsNumVrfRouteMaxThreshExceeded notification is sent (if enabled). When you remove routes from

the VRF so that the number of routes falls below the set limit, the cMplsNumVrfRouteMaxThreshCleared notification is sent. You can clear all routes from the VRF by using the **clear ip route vrf** command. (See the figure below to see when the cMplsNumVrfRouteMaxThreshCleared notification is sent.)

Figure 37: Comparison of Warning and Maximum Thresholds



Notification Specification

In an SNMPv1 notification, each VPN notification has a generic type identifier and an enterprise-specific type identifier for identifying the notification type.

- The generic type for all VPN notifications is “enterpriseSpecific” because this is not one of the generic notification types defined for SNMP.
- The enterprise-specific type is identified as follows:
  - 1 for *mplsVrfIfUp*
  - 2 for *mplsVrfIfDown*
  - 3 for *mplsNumVrfRouteMidThreshExceeded*
  - 4 for *mplsNumVrfRouteMaxThreshExceeded*
  - 5 for *mplsNumVrfSecIllegalLabelThreshExceeded*
  - 6 for cMplsNumVrfRouteMaxThreshCleared

In SNMPv2, the notification type is identified by an SnmpTrapOID varbind (variable binding consisting of an object identifier [OID] type and value) included within the notification message.

Each notification also contains two additional objects from the PPVPN-MPLS-VPN MIB. These objects provide additional information about the event, as follows:

- The VRF interface up/down notifications provide additional variables--*mplsVpnInterfaceConfIndex* and *mplsVpnVrfName*-- in the notification. These variables describe the SNMP interface index and the VRF name, respectively.
- The mid and max threshold notifications include the *mplsVpnVrfName* variable (VRF name) and the *mplsVpnVrfPerfCurrNumRoutes* variable that indicates the current number of routes within the VRF.
- The illegal label notification includes the *mplsVpnVrfName* variable (VRF name) and the *mplsVpnVrfSecIllegalLabelViolations* variable that maintains the current count of illegal labels on a VPN.

Monitoring the PPVPN-MPLS-VPN MIB Notifications

When PPVPN-MPLS-VPN MIB notifications are enabled (see the **snmp-server enable traps mpls vpn** command in the Cisco IOS Multiprotocol Label Switching Command Reference), notification messages relating to specific MPLS VPN events within Cisco software are generated and sent to a specified NMS in the network. Any utility that supports SNMPv1 or SNMPv2 notifications can receive notification messages.

To monitor PPVPN-MPLS-VPN MIB notification messages, log in to an NMS that supports a utility that displays SNMP notifications, and start the display utility.

Unsupported Objects in PPVPN-MPLS-VPN MIB

The following objects from the mplsVpnVrfBgpPathAttrTable are not supported with SNMP management for MPLS VPN features in Cisco software:

- mplsVpnVrfBgpPathAttrPeer
- mplsVpnVrfBgpPathAttrIpAddrPrefixLen
- mplsVpnVrfBgpPathAttrIpAddrPrefix
- mplsVpnVrfBgpPathAttrOrigin
- mplsVpnVrfBgpPathAttrASPathSegment
- mplsVpnVrfBgpPathAttrNextHop
- mplsVpnVrfBgpPathAttrMultiExitDisc
- mplsVpnVrfBgpPathAttrLocalPref
- mplsVpnVrfBgpPathAttrAtomicAggregate
- mplsVpnVrfBgpPathAttrAggregatorAS
- mplsVpnVrfBgpPathAttrAggregatorAddr
- mplsVpnVrfBgpPathAttrCalcLocalPref
- mplsVpnVrfBgpPathAttrBest
- mplsVpnVrfBgpPathAttrUnknown

How to Configure MPLS VPN--MIB Support

Configuring the SNMP Community

An SNMP community string defines the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the router.

Perform this task to configure an SNMP community.

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*]
5. **do copy running-config startup-config**
6. **exit**
7. **show running-config** [*options*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** [*options*] **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. • If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*] **Example:** `Router(config)# snmp-server community comaccess ro`	Sets up the community access string to permit access to the SNMP protocol. • The *string* argument acts like a password and permits access to the SNMP protocol. • The **view** *view-nameview-name* keyword argument pair specifies the name of a previously defined view.

	Command or Action	Purpose
		The view defines the objects available to the community. • The **ro** keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects. • The **rw** keyword specifies read/write access. Authorized management stations are able to both retrieve and modify MIB objects. • The *acl-number* argument is an integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent.
Step 5	**do copy running-config startup-config** **Example:** `Router(config)# do copy running-config startup-config`	Saves the modified configuration to NVRAM as the startup configuration file. • The **do** command allows you to perform EXEC level commands in configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 7	**show running-config** [*options*] **Example:** `Router# show-running config | include smnp-server`	(Optional) Displays the configuration information currently on the router, the configuration for a specific interface, or map-class information. • Use the **show running-config** command to check that the **snmp-server** statements appear in the output.

Configuring the Router to Send SNMP Traps

Perform this task to configure the router to sendm SNMP traps to a host.

The **snmp-server host** command specifies which hosts receive traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

For a host to receive a trap, an **snmp-server host** command must be configured for that host, and, generally, the trap must be enabled globally through the **snmp-server enable traps** command.

Note Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend you define this string using the **snmp-server community** command before using the **snmp-server host** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
4. **snmp-server enable traps mpls vpn** [**illegal-label**] [**max-thresh-cleared**] [**max-threshold**] [**mid-threshold**] [**vrf-down**] [**vrf-up**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrf** *vrf-name*]
	Example:
	`Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn`
		Specifies the recipient of an SNMP notification operation.
		• The*host-addr* argument specifies the name or Internet address of the host (the targeted recipient).
		• The **traps** keyword sends SNMP traps to this host. This is the default.
		• The **informs** keyword sends SNMP informs to this host.
		• The **version** keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, because it allows packet encryption with the **priv** keyword. If you use the **version** keyword, you must specify one of the following:
		  • **1**--SNMPv1. This option is not available with informs.
		  • **2c** --SNMPv2C.
		  • **3**--SNMPv3. The following three optional keywords can follow the **version 3**keyword (**auth**, **noauth**, **priv**).
		• The *community-string* argument is a password-like community string sent with the notification operation.
		• The **udp-port** *port* keyword and argument pair names the User Datagram Protocol (UDP) port of the host to use. The default is 162.

	Command or Action	Purpose
		• The *notification-type* argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent. • The **vrf** *vrf-name* keyword and argument pair specifies the VRF table that should be used to send SNMP notifications.
Step 4	**snmp-server enable traps mpls vpn** [**illegal-label**] [**max-thresh-cleared**] [**max-threshold**] [**mid-threshold**] [**vrf-down**] [**vrf-up**] **Example:** `Router(config)# snmp-server enable traps mpls vpn vrf-down vrf-up`	Enables the router to send MPLS VPN-specific SNMP notifications (traps and informs). • The **illegal-label** keyword enables a notification for any illegal labels received on a VRF interface. Labels are illegal if they are outside the legal range, do not have an LFIB entry, or do not match table IDs for the label. • The **max-thresh-cleared** keyword enables a notification when the number of routes falls below the limit after the maximum route limit was attempted. • The **max-threshold** keyword enables a notification that a route creation attempt was unsuccessful because the maximum route limit was reached. Another MplsNumVrfRouteMaxThreshExceeded notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. The max-threshold value is determined by the **maximum routes** command in VRF configuration mode. • The **mid-threshold** keyword enables a notification of a warning that the number of routes created has crossed the warning threshold. This warning is sent only at the time the warning threshold is exceeded. • The **vrf-down** keyword enables a notification for the removal of a VRF from an interface or the transition of an interface to the down state. • The **vrf-up** keyword enables a notification for the assignment VRF to an interface that is operational or for the transition of a VRF interface to the operationally up state.
Step 5	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.

Configuring Threshold Values for MPLS VPN--SNMP Notifications

Perform this task to configure the following threshold values for MPLS VPN--SNMP notifications:

- The mplsNumVrfRouteMidThreshExceeded notification event is generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the **maximum routes** command in VRF configuration mode. This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.
- The mplsNumVrfRouteMaxThreshExceeded notification event is generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the **maximum routes** command in VRF configuration mode. A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another MplsNumVrfRouteMaxThreshExceeded notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again.

See the figure above for an example of how this notification works and for a comparison of the maximum and warning thresholds.

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *limit warn-threshold* command. Prior to the implementation of the PPVPN-MPLS-VPN MIB, you were not notified when this threshold (or the warning threshold) was reached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **maximum routes** *limit* {*warn-threshold* | **warn-only**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:**	Configures a VRF routing table and enters VRF configuration mode.

	Command or Action	Purpose
	`Router(config)# ip vrf vpn1`	• The *vrf-name* argument specifies the name assigned to a VRF.
Step 4	**maximum routes** *limit* {*warn-threshold* | **warn-only**} **Example:** `Router(config-vrf)# maximum routes 10000 80`	Limits the maximum number of routes in a VRF to prevent a PE router from importing too many routes. • The *limit* argument specifies the maximum number of routes allowed in a VRF. The range is from 1 to 4,294,967,295. • The *warn-threshold* argument generates a warning when the number of routes set by the *warn-threshold* argument is reached and rejects routes that exceed the maximum number set in the *limit* argument. The warning threshold is a percentage from 1 to 100 of the maximum number of routes specified in the *limit* argument. • The **warn-only** keyword specifies that a system logging error message is issued when the maximum number of routes allowed for a VRF exceeds the limit threshold. However, additional routes are still allowed.
Step 5	**end** **Example:** `Router(config-vrf)# end`	(Optional) Exits to privileged EXEC mode.

Configuration Examples for MPLS VPN--MIB Support

Example Configuring the SNMP Community

The following example shows enabling a simple SNMP community group. This configuration permits any SNMP client to access all PPVPN-MPLS-VPN MIB objects with read-only access using the community string comaccess.

```
Router# configure terminal
Router(config)# snmp-server community comaccess ro
```

Verify that the SNMP master agent is enabled for the MPLS VPN--MIB Support feature:

```
Router# show running-config | include snmp-server
Building configuration...
.
snmp-server community comaccess RO
```

Note If you do not see any "snmp-server" statements, SNMP is not enabled on the router.

Example Configuring the Router to Send SNMP Traps

The following example shows you how to enable the router to send MPLS VPN notifications to host 172.20.2.160 using the comaccess community string if a VRF transitions from an up or down state:

```
Router# configure terminal
Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn
Router(config)# snmp-server enable traps mpls vpn vrf-down vrf-up
```

Example Configuring Threshold Values for MPLS VPN--SNMP Notifications

The following example shows how to set a maximum threshold of 10,000 routes and a warning threshold that is 80 percent of the maximum threshold for a VRF named vpn1 on a router:

```
Router(config)# ip vrf vpn1
Router(config-vrf)# maximum routes 10000 80
```

The following example shows how to set a warning threshold of 10,000 routes for a VRF named vpn2 on a router. An error message is generated; however, additional routes are still allowed because a maximum route threshold is not set with this command.

```
Router(config)# ip vrf vpn2
Router(config-vrf)# maximum routes 10000 warn-only
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
MPLS VPN configuration tasks	Configuring MPLS Layer 3 VPNs
A description of SNMP agent support in Cisco software for the MPLS Traffic Engineering MIB (MPLS TE MIB)	MPLS Traffic Engineering (TE) MIB
Overview and configuration tasks for the MPLS distribution protocol	MPLS Label Distribution Protocol

Standards

Standard	Title
draft-ietf-ppvpn-mpls-vpn-mib-05	*MPLS/BGP Virtual Private Network Management Information Base Using SMIv2*

MIBs

MIB	MIBs Link
• MPLS-VPN-MIB • CISCO-IETF-PPVPN-MPLS-VPN-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2233	*The Interfaces Group MIB using SMIv2*
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN--MIB Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 55: Feature Information for MPLS VPN--MIB Support

Feature Name	Releases	Feature Information
MPLS VPN--MIB Support	Cisco IOS XE Release 2.1	The following command was introduced or modified: **snmp-server enable traps mpls vpn**

Glossary

6VPE router—Provider edge router that provides BGP-MPLS IPv6 VPN service over an IPv4-based MPLS core. It is a IPv6 VPN PE, dual-stack router that implements 6PE concepts on the core-facing interfaces.

autonomous system—A collection of networks that share the same routing protocol and that are under the same system administration.

ASN.1 —Abstract Syntax Notation One. The data types independent of particular computer structures and representation techniques. Described by ISO International Standard 8824.

BGP —Border Gateway Protocol. The exterior Border Gateway Protocol used to exchange routing information between routers in separate autonomous systems. BGP uses TCP. Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

BGP prefixes—A route announcement using the BGP. A prefix is composed of a path of autonomous system numbers, indicating which networks the packet must pass through, and the IP block that is being routed. A BGP prefix would look something like: 701 1239 42 206.24.14.0/24. (The /24 part is referred to as a CIDR mask.) The /24 indicates that there are 24 ones in the netmask for this block starting from the left side. A /24 corresponds to the natural mask 255.255.255.0.

CE router—customer edge router. A router on the border between a VPN provider and a VPN customer that belongs to the customer.

CIDR —classless interdomain routing. A technique supported by BGP4 and based on route aggregation. CIDR allows routers to group routes to reduce the quantity of routing information carried by the core routers. With CIDR, several IP networks appear to networks outside the group as a single, larger entity. With CIDR, IP addresses and their subnet masks are written as four octets, separated by periods, followed by a forward slash and a two-digit number that represents the subnet mask.

Cisco Express Forwarding—An advanced Layer 3 IP switching technology. Cisco Express Forwarding optimizes network performance and scalability for networks with large and dynamic traffic patterns.

community —In SNMP, a logical group of managed devices and NMSs in the same administrative domain.

community name—*See* community string.

community string—A text string that acts as a password and is used to authenticate messages sent between a managed station and a router containing an SNMP agent. The community string is sent in every packet between the manager and the client. Also called a community name.

IETF —Internet Engineering Task Force. A task force consisting of over 80 working groups responsible for developing Internet standards. The IETF operates under the auspices of ISOC. *See also* ISOC.

informs —A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, and a trap notification does not.

ISOC —Internet Society. An international nonprofit organization, founded in 1992, that coordinates the evolution and use of the Internet. In addition, ISOC delegates authority to other groups related to the Internet, such as the IAB. ISOC is headquartered in Reston, Virginia (United States).

label —A short, fixed-length data construct that tells switching nodes how to forward data (packets or cells).

LDP —Label Distribution Protocol. A standard protocol between MPLS-enabled routers that is used for the negotiation of the labels (addresses) used to forward packets.

LFIB —Label Forwarding Information Base. In the Cisco Label Switching system, the data structure for storing information about incoming and outgoing tags (labels) and associated equivalent packets suitable for labeling.

LSR —label switch router. A device that forwards MPLS packets based on the value of a fixed-length label encapsulated in each packet.

MIB —Management Information Base. A database of network management information that is used and maintained by a network management protocol such as SNMP or CMIP. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS —Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

MPLS interface—An interface on which MPLS traffic is enabled.

MPLS VPN—Multiprotocol Label Switching Virtual Private Network. An IP network infrastructure delivering private network services over a public infrastructure using a Layer 3 backbone. Using MPLS VPNs in a Cisco IOS network provides the capability to deploy and administer scalable Layer 3 VPN backbone services including applications, data hosting network commerce, and telephony services to business customers.

For an MPLS VPN solution, an MPLS VPN is a set of provider edge routers that are connected by means of a common “backbone” network to supply private IP interconnectivity between two or more customer sites for a given customer. Each VPN has a set of provisioning templates and policies and can span multiple provider administrative domains (PADs).

NMS —network management system. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

notification —A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS software has occurred. *See also* trap.

PE router—provider edge router. A router on the border between a VPN provider and a VPN customer that belongs to the provider.

QoS —quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RIB —Routing Information Base. Also called the routing table.

RT —route target. An extended community attribute that identifies a group of routers and, in each router of that group, a subset of forwarding tables maintained by the router that can be populated with a BGP route carrying that extended community attribute. The RT is a 64-bit value by which Cisco IOS software discriminates routes for route updates in VRFs.

SNMP —Simple Network Management Protocol. The network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security. *Seealso* SNMP2.

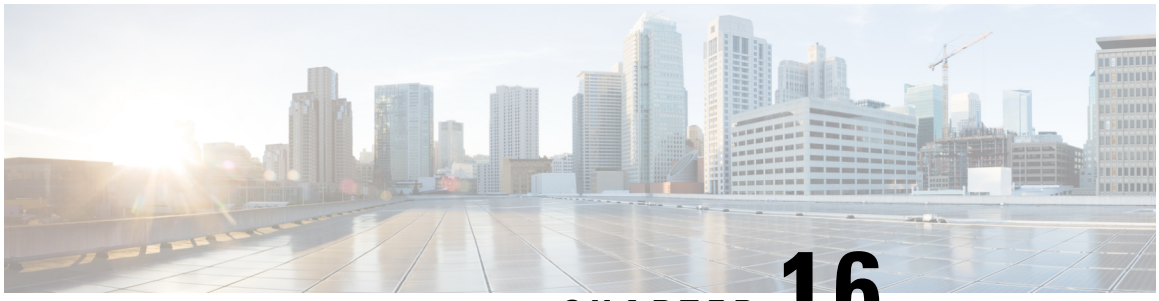
SNMP2 —SNMP Version 2. Version 2 of the popular network management protocol. SNMP2 supports centralized and distributed network management strategies, and includes improvements in the Structure of Management Information (SMI), protocol operations, management architecture, and security. *See also* SNMP.

trap —A message sent by an SNMP agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps (notifications) are less reliable than inform requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received. *See also* notification.

VPN —Virtual Private Network. A group of sites that, as the result of a set of administrative policies, are able to communicate with each other over a shared backbone network. A VPN is a secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone. *Seealso* MPLS VPN.

VPN ID —A mechanism that identifies a VPN based on RFC 2685. A VPN ID consists of an Organizational Unique Identifier (OUI), a three-octet hex number assigned by the IEEE Registration Authority, and a VPN index, a four-octet hex number, which identifies the VPN within the company.

VRF —VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.



CHAPTER 16

MPLS VPN SNMP Notifications

This document describes the Simple Network Management Protocol (SNMP) agent support in Cisco IOS for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) event notifications as implemented in the notifications section of the draft *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2* (*draft-ietf-ppvpn-mpls-vpn-mib-03.txt*).

The MPLS VPN technology allows service providers to offer intranet and extranet VPN services that directly connect their customers' remote offices to a public network with the same security and service levels that a private network offers. The Provider-Provisioned VPN (PPVPN)-MPLS-VPN MIB notifications provide SNMP notification for critical MPLS VPN events.

The MPLS VPN SNMP Notifications feature provides the following benefits:

- A standards-based SNMP interface for retrieving information about critical MPLS VPN events.
- The generation and queuing of notifications that call attention to major changes in the operational status of MPLS VPN enabled interfaces; the forwarding of notification messages to a designated NMS for evaluation and action by network administrators.
- Advanced warning when VPN routing tables are approaching or exceed their capacity.
- Warnings about the reception of illegal labels on a VRF enabled interface. Such receptions may indicate misconfiguration or an attempt to violate security.

Prerequisites for MPLS VPN SNMP Notifications

The MPLS VPN SNMP Notifications feature requires the following:

- SNMP is installed and enabled on the label switching routers.
- Multiprotocol Label Switching (MPLS) is enabled on the label switching routers.

- Multiprotocol Border Gateway Protocol (BGP) is enabled on the label switching routers.
- Cisco Express Forwarding is enabled on the label switching routers.

Restrictions for MPLS VPN SNMP Notifications

- The MPLS-VPN-MIB agent is not implemented in this release.
- Configuration of the MIB using the SNMP SET command is not supported in this release.
- The retrieval of MPLS-VPN-MIB objects using SNMP GET is not supported in this release.

Information About MPLS VPN SNMP Notifications

Cisco Implementation of MPLS VPN MIB

SNMP agent code operating with the notifications of the MPLS VPN SNMP Notifications feature enables a standardized, SNMP-based approach to monitoring the MPLS VPN MIB notifications that aid in the management of Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) in Cisco software.

The MPLS VPN SNMP Notifications feature is based on the IETF draft specification *draft-ietf-ppvpn-mpls-vpn-mib-02.txt*, which includes notification objects that support MPLS VPN notification events. This IETF draft MIB, which undergoes revisions from time to time, is being evolved toward becoming a standard. Accordingly, the Cisco implementation of features of the MPLS VPN MIB is expected to track the evolution of the IETF draft MIB, and may change accordingly.

Some slight differences between the IETF draft MIB and the actual implementation of MPLS VPNs within Cisco software require some minor translations between the MPLS VPN MIB and the internal data structures of Cisco software. These translations are accomplished by means of the SNMP agent code. Also, while running as a low priority process, the SNMP agent provides a management interface to Cisco software. SNMP adds little overhead on the normal functions of the device.

The SNMP objects defined in the MPLS VPN MIB notifications can be viewed by any standard SNMP utility. The network administrator can retrieve information in the MPLS VPN MIB using standard SNMP **get** and **getnext** operations for SNMP v1, v2, and v3.

All MPLS VPN MIB objects are based on the IETF draft MIB; thus, no specific Cisco SNMP application is required to support the functions and operations pertaining to the MPLS VPN SNMP Notifications feature.

This section contains the following information about the Cisco implementation of the MPLS VPN MIB:

Capabilities Supported by MPLS VPN SNMP Notifications

The following functionality is supported in this release for the MPLS VPN SNMP Notifications feature. This feature provides you with the ability to do the following:

- Create and send notification messages that signal changes when critical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) events occur.
- Enable, disable, and configure notification messages for MPLS VPN events by using extensions to existing SNMP CLI commands.

- Specify the IP address of a network management system (NMS) in the operating environment to which notification messages are sent.
- Write notification configurations into nonvolatile memory.

Notification Generation Events for the MPLS VPN MIB

The following notifications of the MPLS VPN MIB are implemented for this release:

- **mplsVrfIfUp**—Sent to an NMS when an interface comes up and is assigned a VPN routing/forwarding table instance (VRF).
- **mplsVrfIfDown**—Generated and sent to the NMS when a VRF is removed from an interface or the interface transitions from an operationally "up" state to a "down" state.

Note For the mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the **snmp-server traps atm subif** command or the **snmp-server traps frame-relay subif** command on the subinterfaces, respectively.

- **mplsNumVrfRouteMidThreshExceeded**—Generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the following commands:

```
Router(config)# ip vrf vrf-name
Router(config-vrf)# maximum routes max-thresh
mid-thresh (% of max)
```

This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS. (See the figure below for a comparison of the warning and maximum thresholds.)

- **mplsNumVrfRouteMaxThreshExceeded**—Generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the following CLI commands:

```
Router(config)# ip vrf vrf-name
Router(config-vrf)# maximum routes max-thresh
mid-thresh (% of max)
```

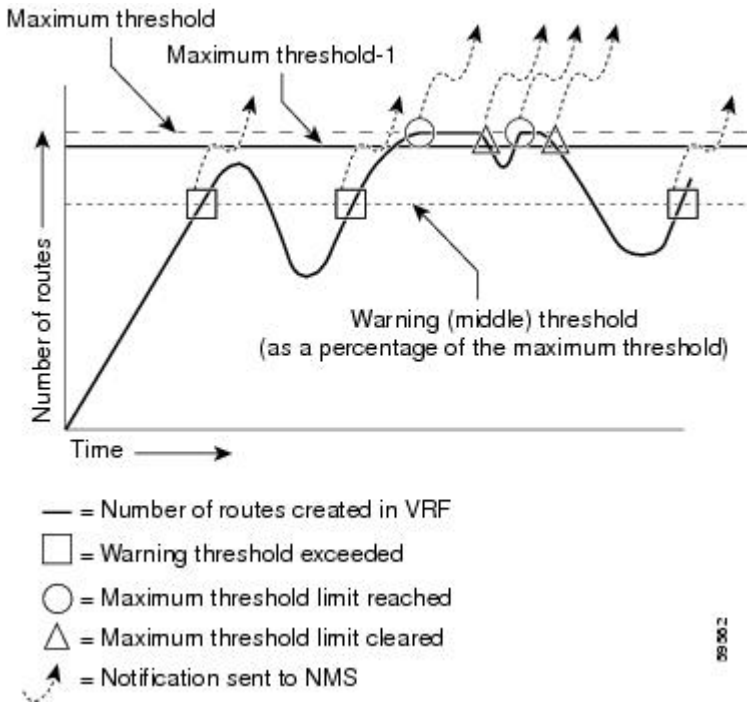
A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. (See the figure below for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command. Prior to this implementation of the MPLS-VPN-MIB, you were not notified when this threshold (or the warning threshold) was reached.

- **mplsNumVrfSecIllegalLabelThreshExceeded**—Generated and sent when the amount of illegal labels received on a VRF interface exceeds the threshold *mplsVpnVrfSecIllegalLabelRcvThresh*. This threshold

is defined with a value of 0. Therefore, a notification is sent when the first illegal label is received on a VRF. Labels are considered illegal if they are outside of the valid label range, do not have a Label Forwarding Information Base (LFIB) entry, or the table ID of the message does not match the table ID for the label in the LFIB.

Figure 38: Comparison of Warning and Maximum Thresholds



Notification Specification for MPLS-VPN-MIB

In an SNMPv1 notification, each VPN notification has a generic type identifier and an enterprise-specific type identifier for identifying the notification type.

- The generic type for all VPN notifications is “enterpriseSpecific” as this is not one of the generic notification types defined for SNMP.
- The enterprise-specific type is identified as follows:
  - 1 for *mplsVrfIfUp*
  - 2 for *mplsVrfIfDown*
  - 3 for *mplsNumVrfRouteMidThreshExceeded*
  - 4 for *mplsNumVrfRouteMaxThreshExceeded*
  - 5 for *mplsNumVrfSecIllegalLabelThreshExceeded*

In SNMPv2, the notification type is identified by an **SnmpTrapOID** varbind (variable binding consisting of an object identifier (OID) type and value) included within the notification message.

Each notification also contains two additional objects from the MPLS-VPN-MIB. These objects provide additional information about the event, as follows:

- The VRF interface up/down notifications provide additional variables--*mplsVpnInterfaceConfIndex* and *mplsVpnVrfName*-- in the notification. These variables describe the SNMP interface index and the VRF name, respectively.
- The mid and max threshold notifications include the *mplsVpnVrfName* variable (VRF name) as well as the *mplsVpnVrfPerfCurrNumRoutes* variable that indicates the current number of routes within the VRF.
- The illegal label notification includes the *mplsVpnVrfName* variable (VRF name) and the *mplsVpnVrfSecIllegalLabelViolations* variable that maintains the current count of illegal labels on a VPN.

Monitoring the MPLS VPN SNMP Notifications

When MPLS-VPN-MIB notifications are enabled, notification messages relating to specific Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) events within Cisco software are generated and sent to a specified network management system (NMS) in the network. Any utility that supports SNMPv1 or SNMPv2 notifications can receive notification messages.

To monitor MPLS-VPN-MIB notification messages, log in to an NMS that supports a utility that displays SNMP notifications, and start the display utility.

How to Configure the MPLS VPN SNMP Notifications

Configuring an SNMP Community

An SNMP community string defines the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the device.

Perform this task to configure an SNMP community.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*]
5. **do copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:**	Displays the running configuration to determine if an SNMP agent is already running.

	Command or Action	Purpose
	`Device# show running-config`	If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*] **Example:** `Device(config)# snmp-server community comaccess ro`	Sets up the community access string to permit access to the Simple Network Management Protocol (SNMP). • The *string* argument acts like a password and permits access to the SNMP protocol. • The **view***view-name* keyword and argument specifies the name of a previously defined view. The view defines the objects available to the community. • The **ro** keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects. • The **rw** keyword specifies read-write access. Authorized management stations are able to both retrieve and modify MIB objects. • The *acl-number* argument is an integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent.
Step 5	**do copy running-config startup-config** **Example:** `Device(config)# do copy running-config startup-config`	Saves the modified configuration to nonvolatile memory (NVRAM) as the startup configuration file. (The **do** command allows you to perform Exec level commands in configuration mode.)

Configuring the Device to Send SNMP Traps

Perform this task to configure the device to send traps to a host.

The **snmp-server host** command specifies which hosts receive traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

Note Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
4. Do one of the following:
   - **snmp-server enable traps atm** [**pvc** | **subif**]
   - **snmp-server enable traps frame-relay** [**subif**]
5. **snmp-server enable traps mpls vpn**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
	Example:
	`Device(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn`
		Specifies the recipient of an SNMP notification operation.
		• The *host-addr* argument specifies the name or Internet address of the host (the targeted recipient).
		• The **traps** keyword sends SNMP traps to this host. This is the default.
		• The **informs** keyword sends SNMP informs to this host.
		• The **version** keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, as it allows packet encryption with the **priv** keyword. If you use the **version** keyword, you must specify one of the following:
		  • **1**—SNMPv1. This option is not available with informs.
		  • **2c**—SNMPv2C.
		  • **3**—SNMPv3. The following three optional keywords can follow the **version 3** keyword (**auth**, **noauth**, **priv**).
		• The *community-string* argument is a password-like community string sent with the notification operation.

	Command or Action	Purpose
		• The **udp-port** *port* keyword and argument names the UDP port of the host to use. The default is 162. • The *notification-type* argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent. MPLS VPN notifications are specified with the **mpls-vpn** keyword.

Step 4	Do one of the following:
	• **snmp-server enable traps atm** [**pvc** | **subif**]
	• **snmp-server enable traps frame-relay** [**subif**]
	Example:
	`Device(config)# snmp-server enable traps atm subif`
	Example:
	`Device(config)# snmp-server enable traps frame-relay subif`
		(For ATM subinterfaces only) Enables the sending of ATM SNMP notifications.
		• The **pvc** keyword enables SNMP ATM permanent virtual circuit (PVC) traps.
		• The **subif** keyword enables SNMP ATM subinterface traps.
		or
		(For Frame Relay subinterfaces only) Enables Frame Relay DLCI link status SNMP notifications.
		• The **subif** keyword enables SNMP Frame Relay subinterface traps.
		Note For mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the appropriate **snmp-server enable traps** command with the **subif** keyword.

Step 5	**snmp-server enable traps mpls vpn** **Example:** `Device(config)# snmp-server enable traps mpls vpn vrf-up vrf-down`	Enables the device to send MPLS VPN SNMP notifications.
Step 6	**end** **Example:** `Device(config)# end`	(Optional) Returns to user EXEC mode.

Configuring Threshold Values for MPLS VPN SNMP Notifications

Perform this task to configure threshold values for MPLS VPN SNMP notifications.

The **mplsNumVrfRouteMidThreshExceeded** notification event is generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the **maximum routes** command in VRF configuration mode. This notification is sent to the NMS only at the time the threshold is

exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.

The **mplsNumVrfRouteMaxThreshExceeded** notification event is generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the **maximum routes** command in VRF configuration mode. A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again.

(See the figure above for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command. Prior to this implementation of the MPLS-VPN-MIB, you were not notified when this threshold (or the warning threshold) was reached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **maximum routes** *limit* {*warn-threshold* | **warning-only**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vpn1`	Configures a VRF routing table. • The *vrf-name* argument specifies the name assigned to a VRF.
Step 4	**maximum routes** *limit* {*warn-threshold*	**warning-only**} **Example:** `Device(config-vrf)# maximum routes 10000 80`	Limits the maximum number of routes in a VRF to prevent a PE device from importing too many routes. • The *limit* argument specifies the maximum number of routes allowed in a VRF. You may select from 1 to 4,294,967,295 routes to be allowed in a VRF.

	Command or Action	Purpose
		• The *warn-threshold* argument specifies when the threshold limit is reached and routes are rejected. The threshold limit is a percentage of the *limit* specified, from 1 to 100 percent. • The **warning-only** keyword specifies that a SYSLOG error message is issued when the maximum number of routes allowed for a VRF exceeds the threshold. However, additional routes are still allowed.
Step 5	**end** **Example:** `Device(config-vrf)# end`	(Optional) Returns to privileged EXEC mode.

Configuration Examples for MPLS VPN SNMP Notifications

Example: Configuring the Community

The following example shows enabling a simple SNMP community group. This configuration permits any SNMP client to access all MPLS-VPN-MIB objects with read-only access using the community string comaccess.

```
Device# configure terminal
Device(config)# snmp-server community comaccess ro
```

Verify that the SNMP master agent is enabled for the MPLS VPN SNMP Notifications feature:

```
Device# show running-config | include snmp-server
Building configuration...
....
snmp-server community comaccess RO
....
```

Note If you do not see any "snmp-server" statements, SNMP has not been enabled on the device.

Example: Configuring the Device to Send SNMP Traps

The following example shows you how to enable the device to send MPLS VPN notifications to host 172.20.2.160 using the comaccess community string if a VRF transitions from a down state to an up state or from an up state to a down state.

```
Device# configure terminal
Device(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn
Device(config)# snmp-server enable traps mpls vpn vrf-up vrf-down
```

Example: Configuring Threshold Values for MPLS VPN SNMP Notifications

The following example shows how to set a maximum threshold or 10000 routes and a warning threshold that is 80 percent of the maximum threshold for a VRF named vpn1 on a device:

```
Device(config)# ip vrf vpn1
Device(config)# maximum routes 10000 80
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS Virtual Private Network (VPN) configuration tasks	"MPLS Virtual Private Networks" module in the *MPLS Layer 3 VPNs Configuration Guide*

MIBs

MIBs	MIBs Link
MPLS/BGP Virtual Private Network Management Information Base Using SMIv2 (*draft-ietf-ppvpn-mpls-vpn-mib-03.txt*) MPLS-VPN-MIB.my	To obtain lists of supported MIBs by platform and Cisco software release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2233	*The Interfaces Group MIB using SMIv2*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN SNMP Notifications

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 56: Feature Information for MPLS VPN SNMP Notifications

Feature Name	Releases	Feature Information
MPLS VPN SNMP Notifications	12.0(21)ST 12.0(22)S 12.2(13)T	The MPLS VPN SNMP Notifications feature provides Simple Network Management Protocol (SNMP) agent support in Cisco software for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) event notifications. In Cisco IOS Release 12.0(21)ST, this feature was introduced. In Cisco IOS Release 12.0(22)S, this feature was integrated. In Cisco IOS Release 12.2(13)T, this feature was integrated. Supported platforms: • Cisco IOS 12.0 S and ST Releases: Cisco 7500 series, Cisco 12000 series. • Cisco IOS 12.2 T Releases: Cisco 3620, Cisco 3640, Cisco 7200 series, Cisco 7500 series, Cisco MGX 8850-RPM. **Note** In Cisco IOS Releases 12.0(21)ST and 12.0(22)S, the PPVPN MPLS-VPN-MIB notifications are described in the *MPLS VPN--SNMP MIB Support* feature module. The following commands were introduced or modified: **snmp-server enable traps mpls vpn**, **snmp-server host**.

Glossary

ASN.1—Abstract Syntax Notation One. OSI language for describing data types independent of particular computer structures and representation techniques. Described by ISO International Standard 8824.

BGP—Border Gateway Protocol. The exterior Border Gateway Protocol used to exchange routing information between routers in separate autonomous systems. BGP uses Transmission Control Protocol (TCP). Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

CEF—Cisco Express Forwarding. An advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns.

CE device—customer edge router. A router on the border between a VPN provider and a VPN customer that belongs to the customer.

community—In SNMP, a logical group of managed devices and NMSs in the same administrative domain.

community name—*See* community string.

community string—Text string that acts as a password and is used to authenticate messages sent between a managed station and a router containing an SNMP agent. The community string is sent in every packet between the manager and the client. Also called a community name.

IETF—Internet Engineering Task Force. Task force consisting of over 80 working groups responsible for developing Internet standards. The IETF operates under the auspices of ISOC. *See also* ISOC.

informs—A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, and a trap notification does not.

ISOC—Internet Society. International nonprofit organization, founded in 1992, that coordinates the evolution and use of the Internet. In addition, ISOC delegates authority to other groups related to the Internet, such as the IAB. ISOC is headquartered in Reston, Virginia (United States).

label—A short, fixed-length data construct that tells switching nodes how to forward data (packets or cells).

label distribution protocol—*See* LDP.

label forwarding information base—*See* LFIB.

label switch router—*See* LSR.

LDP —label distribution protocol. A standard protocol between MPLS-enabled routers that is used for the negotiation of the labels (addresses) used to forward packets.

LFIB —label forwarding information base. In the Cisco Label Switching system, the data structure for storing information about incoming and outgoing tags (labels) and associated equivalent packets suitable for labeling.

LSR —label switch router. A device that forwards MPLS packets based on the value of a fixed-length label encapsulated in each packet.

MIB —Management Information Base. Database of network management information that is used and maintained by a network management protocol such as SNMP or CMIP. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS —Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

MPLS interface—An interface on which MPLS traffic is enabled.

MPLS VPN—Multiprotocol Label Switching Virtual Private Network. Using MPLS VPNs in a Cisco network provide the capability to deploy and administer scalable Layer 3 VPN backbone services including applications, data hosting network commerce, and telephony services, to business customers. A VPN is a secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.

For an MPLS VPN Solution, an MPLS VPN is a set of PEs that are connected by means of a common "backbone" network to supply private IP interconnectivity between two or more customer sites for a given customer. Each VPN has a set of provisioning templates and policies and can span multiple provider administrative domains (PADs).

Multiprotocol label Switching—*See* MPLS.

notification —A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco software has occurred. *See also* trap.

NMS —network management system. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

PE device—provider edge router. A router on the border between a VPN provider and a VPN customer that belongs to the provider.

PPVPN —Provider-Provisioned VPN. The name of the IETF working group that is developing the PPVPN-MPLS-VPN MIB (MPLS-VPN-MIB).

QoS —quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP —Resource Reservation Protocol. Protocol for reserving network resources to provide Quality of Service guarantees to application flows.

Simple Network Management Protocol—*See* SNMP.

SNMP —Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security. See *also* SNMP2.

SNMP2 —SNMP Version 2. Version 2 of the popular network management protocol. SNMP2 supports centralized as well as distributed network management strategies, and includes improvements in the SMI, protocol operations, management architecture, and security. *See also* SNMP.

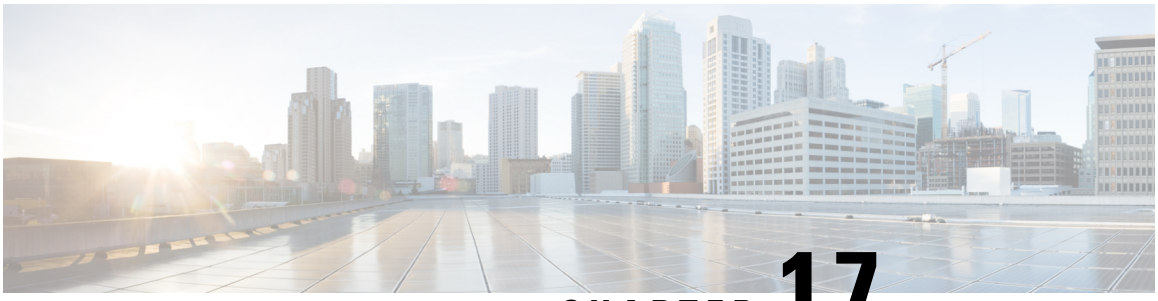
traffic engineering—The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

trap —A message sent by an SNMP agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps (notifications) are less reliable than inform requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received. *See also* notification.

VPN —Virtual Private Network. A group of sites that, as the result of a set of administrative policies, are able to communicate with each other over a shared backbone network. *See* MPLS VPN.

VRF —VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what

goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE device.



CHAPTER 17

Pseudowire Emulation Edge-to-Edge MIBs

The Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services feature provides Simple Network Management Protocol (SNMP) support within an Any Transport over Multiprotocol Label Switching (AToM) infrastructure emulating Ethernet, Frame Relay, and ATM services over packet switched networks (PSNs). The Pseudowire Emulation Edge-to-Edge (PWE3) MIBs are the following:

- CISCO-IETF-PW-MIB (PW-MIB)
- CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB)
- CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB)
- CISCO-IETF-PW-FR-MIB (PW-FR-MIB)
- CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB)

Prerequisites for Pseudowire Emulation Edge-to-Edge MIBs

- SNMP must be enabled on the label switch routers (LSRs).
- MPLS must be enabled on the LSRs.
- Pseudowires must be configured with Ethernet, Frame Relay, or ATM access circuits. (For more detailed information, see the “Any Transport over MPLS” module.

Restrictions for Pseudowire Emulation Edge-to-Edge MIBs

The PWE3 MIBs are limited to read-only (RO) permission for MIB objects except for the cpwVcUp and cpwVcDown notification enable object, cpwVcUpDownNotifEnable, which has been extended to be writable by the SNMP agent.

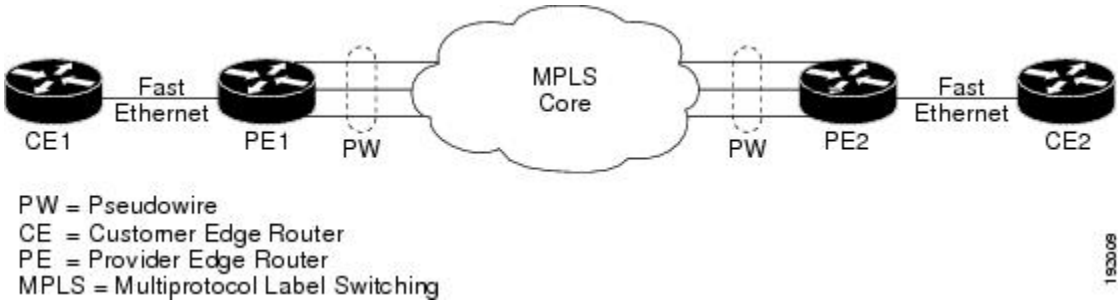
- The following tables in the PW-MIB are not supported:
  - cpwVcPerfCurrentTable
  - cpwVcPerfIntervalTable
- The following objects in the PW-MPLS-MIB are not supported:
  - cpwVcMplsOutboundIndexNext
  - cpwVcMplsInboundIndexNext
- The following tables in the PW-ENET-MIB are not supported:
  - cpwVcEnetMplsPriMappingTable
  - cpwVcEnetStatsTable
- The following table in the PW-FR-MIB is not supported:
  - cpwVcFrPMTable
- The PW-ATM-MIB does not support a high-capacity cell counter per virtual path (VP) or cells per port.
- The PW-ATM-MIB virtual path identifier (VPI)/virtual channel identifier (VCI) value for port mode cell relay is 0.
- The PW-ATM-MIB VP cell relay VCI value is 0.
- The PW-ATM-MIB VP does not support ATM adaptation layer 5 (AAL5); therefore, all packet counters are invalid.

Note This feature is not supported over Ethernet, Frame Relay, and ATM in all releases. For more detailed information, see the "Feature Information for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet Frame Relay and ATM Services" section.

Information About Pseudowire Emulation Edge-to-Edge MIBs

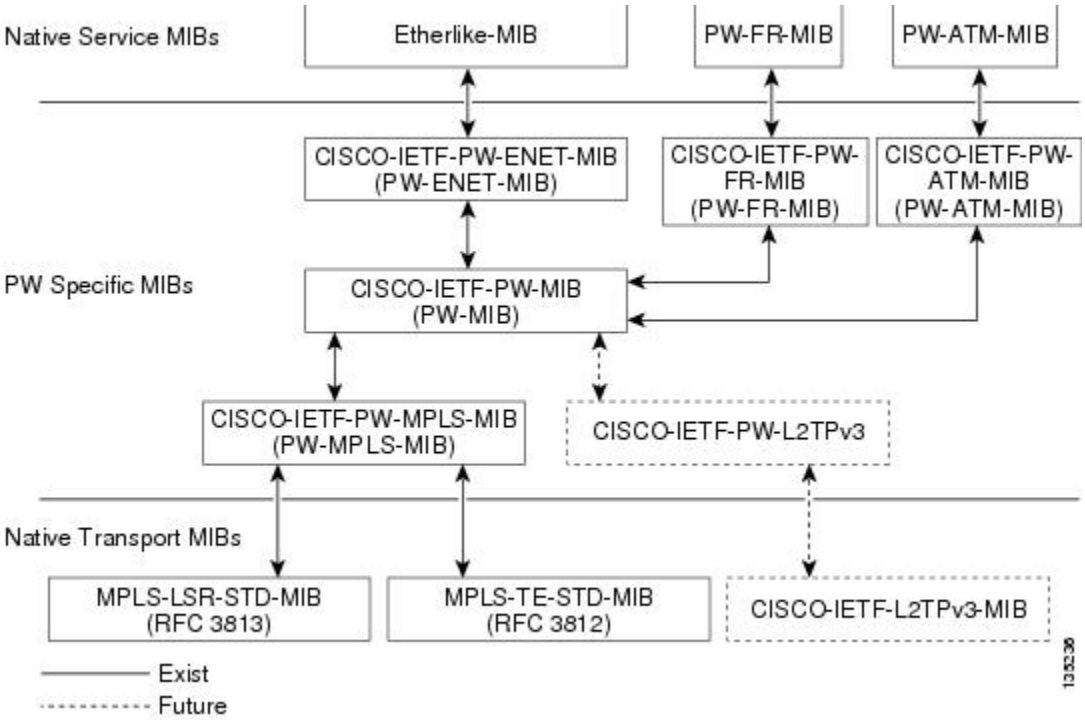
The Function of a Pseudowire in the PWE3 MIBs

A pseudowire is a point-to-point connection between pairs of provider edge (PE) routers (as shown in the figure below). Its primary function is to emulate services like Ethernet over an underlying core MPLS network through encapsulation into a common MPLS format. By encapsulating services into a common MPLS format, a pseudowire allows carriers to converge their services to an MPLS network.



PWE3 MIBs Architecture

The PWE3 MIBs architecture shown in the figure below categorizes three groups of MIBs that, when used together, provide the complete emulated service; the native transport, which carries the service across the core network; and the relationship between the two.



The architecture is modular in that once deployed, new emulated service MIB modules or additional transport MIB modules “plug in” to or extend the existing infrastructure rather than require a new and unique one. This allows you to build management applications without the concern of a new service requiring the deployment of a completely different management strategy. Because the architecture is a generalized association mechanism between existing service and transport MIB modules, native MIB modules work in the absence of the associated PWE3-specific MIBs. The advantage is that if a PWE3-specific MIB has not yet been deployed in Cisco software, which associates a service or transport with pseudowires, these MIB modules can still be queried. However, the only drawback is that the associations with the pseudowires are absent.

Components and Functions of the PWE3 MIBs

The PWE3 MIBs have the following components and functions:

- PW-MIB (the pseudowire MIB)

This MIB binds the PW-MPLS-MIB and the PW-ENET-MIB together, and provides status of the pseudowire emulation and statistics and configuration information. The PW-MIB also defines the notifications for pseudowire fault and event monitoring.

- PW-MPLS-MIB (the pseudowire MPLS-MIB)

This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation MPLS services, such as MPLS-traffic engineering (TE)-PSN and MPLS-non-TE-PSN.

This MIB shows the following:

-
  - Cross-connect (XC) indexes for virtual circuits (VCs) that are Label Distribution Protocol (LDP)-signaled and have a preferred path that is not set to an MPLS TE tunnel.
  - Tunnel indexes for VCs with a preferred path set to a TE tunnel and an output interface that is a TE tunnel.

- PW-ENET-MIB (the pseudowire Ethernet services MIB)

This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation Ethernet services.

- PW-FR-MIB (the pseudowire Frame Relay services MIB)

This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation Frame Relay services.

This MIB uses a Frame Relay over pseudowire (FRoPW) connection that consists of two segments: the Frame Relay segment and the pseudowire segment. The PW-FR-MIB provides hooks to those segments. The PW MIB contains information about the pseudowire segment, and the PW-FR-MIB contains information about the Frame Relay segment.

The PW-FR-MIB is defined at the Pseudowire Service Emulation Layer and resides on top of the generic PW-MIB as shown in the figure above. Therefore, the PW-FR-MIB is highly dependent on the existence and the service provided by the PW-MIB. In addition, an existing PW-FR connection entry must associate with an existing VC entry in the PW-MIB.

The PW-FR-MIB and the generic PW-MIB are logically tied by the PW VC Index, which is an internal index defined to support the PW-MIB. Each PW VC index uniquely maps into an existing VC entry in the PW-MIB and the PW-FR-MIB.

- PW-ATM-MIB (the pseudowire ATM services MIB)

This MIB contains managed objects that can be used by a network manager to monitor pseudowire emulation ATM services.

This MIB uses an ATM over pseudowire (ATMoPW) connection that consists of two segments: the ATM segment and the pseudowire segment. The PW-ATM-MIB provides hooks to those segments. The PW MIB contains information about the pseudowire segment, and the PW-ATM-MIB contains information about the ATM segment called the attachment circuit.

The PW-ATM-MIB is defined at the Pseudowire Service Emulation Layer and resides on top of the generic PW-MIB as shown in the figure above. Therefore, the PW-ATM-MIB is highly dependent on the existence and the service provided by the PW-MIB. In addition, an existing PW-ATM connection entry must associate with an existing VC entry in the PW-MIB.

The PW-ATM-MIB and the generic PW-MIB are logically tied by the PW VC Index, which is an internal index defined to support the PW-MIB. Each PW VC index uniquely maps into an existing VC entry in the PW-MIB and the PW-ATM-MIB.

Tables in the PW-MIB

The PW-MIB consists of the following tables:

- cpwVcTable -- Contains high-level generic parameters related to VC creation. This table is implemented as read only and is indexed by the cpwVcIndex, which uniquely identifies a singular connection. A row in this table represents an emulated virtual connection. This table is used for all VC types.
- cpwVcPerfTotalTable -- Provides per-VC performance information from the VC start time. This table is indexed by the cpwVcIndex.
- cpwVcIdMappingTable -- Provides reverse mapping of the existing VCs based on VC type and VC ID ordering. This table is typically useful for element manager software (EMS) ordered query of existing VCs. This table is indexed by cpwVcIdMappingVcType, cpwVcIdMappingVcID, cpwVcIdMappingPeerAddrType, and cpwVcIdMappingPeerAddr. This table is implemented as read only.
- cpwVcPeerMappingTable -- Provides reverse mapping of the existing VCs based on VC type and VC ID ordering. This table is typically useful for EMS ordered query of existing VCs. This table is indexed by cpwVcPeerMappingPeerAddrType, cpwVcPeerMappingPeerAddr, cpwVcPeerMappingVcType, and cpwVcPeerMappingVcID. This table is implemented as read only.

cpwVcTable

The table below lists the cpwVcTable objects and their descriptions.

Table 57: cpwVcTable Objects and Descriptions

Objects	Description
cpwVcType	Indicates the service to be carried over this VC. This is circuit type information.
cpwVcOwner	Set by the operator to indicate the protocol responsible for establishing this VC. Values are the following: • manual(1)--Used when no maintenance protocol (PW signaling) is needed to set up the VC, such as configuration of entries in the VC tables including VC labels, and so forth. • maintenanceProtocol(2)--Used for standard signaling of the VC for the specific PSN; for example, LDP for MPLS PSN as specified in draft-martini-l2circuit-trans-mpls or the Layer 2 Tunneling Protocol (L2TP). • other(3)--Used for all other types of signaling.

Objects	Description
cpwVcPsnType	Set by the operator to indicate the PSN type on which this VC is carried. Based on this object, the relevant PSN table entries are created in the PSN-specific MIB modules. For example, if mpls(1) is defined, the agent creates an entry in the cpwVcMplsTable, which further defines the MPLS PSN configuration.
cpwVcSetUpPriority	Defines the relative setup priority of the VC in a lowest-to-highest manner, where 0 is the highest priority. This value is significant if there are competing resources between VCs and the implementation supports this feature. Because this is not implemented in AToM, the value of 0 is used.
cpwVcHoldingPriority	Defines the relative holding priority of the VC in a lowest-to-highest manner, where 0 is the highest priority. This value is significant if there are competing resources between VCs and the implementation supports this feature. Because this is not implemented in AToM, the value of 0 is used.
cpwVcInboundMode	Enables greater security for implementations that use per-platform VC label space. Modes are the following: • strict(1) • loose(2) In strict mode, packets coming from the PSN are accepted only from tunnels that are associated to the same VC via the inbound tunnel table in the case of MPLS, or as identified by the source IP address in the case of L2TP or IP PSN. The entries in the inbound tunnel table are either explicitly configured or implicitly known by the maintenance protocol used for VC setup. If such association is not known, not configured, or not desired, loose mode should be configured, and the node should accept the packet based on the VC label only, regardless of the outer tunnel used to carry the VC.
cpwVcPeerAddrType	Denotes the address type of the peer node maintenance protocol (signaling) address if the PW maintenance protocol is used for the VC creation. It should be set to unknown if the PW maintenance protocol is not used; for example, cpwVcOwner is set to manual.
cpwVcPeerAddr	Contains the value of the peer node address of the PW maintenance protocol entity. This object should contain a value of 0 if not relevant (manual configuration of the VC).
cpwVcID	Use in the outgoing VC ID field within the VC forward equivalence class (FEC) element with LDP signaling or the PW ID attribute-value (AV) pair for the L2TP.
cpwVcLocalGroupID	Use in the Group ID field sent to the peer PW within the maintenance protocol for VC setup; 0 if not used.
cpwVcControlWord	Defines if the control word is sent with each packet by the local node.

Objects	Description
cpwVcLocalIfMtu	If not = 0, the optional IfMtu object in the maintenance protocol is sent with this value, representing the locally supported maximum transmission unit (MTU) size over the interface (or the virtual interface) associated with the VC.
cpwVcLocalIfString	Each VC is associated to an interface (or a virtual interface) in the ifTable of the node as part of the service configuration. This object defines if the maintenance protocol sends the interface's name as it appears in the ifTable in the name object as part of the maintenance protocol. If this object is set to false, the optional element is not sent.
cpwVcRemoteGroupID	Obtained from the Group ID field as received via the maintenance protocol used for VC setup; 0 if not used. The value of 0xFFFF is used if the object is not defined by the VC maintenance protocol.
cpwVcRemoteControlWord	If the maintenance protocol is used for VC establishment, this parameter indicates the received status of the control word usage; that is, if packets are received with the control word or not. The value of notYetKnown is used while the maintenance protocol has not yet received the indication from the remote node. In a manual configuration of the VC, this parameter indicates to the local node the expected encapsulation for the received packets.
cpwVcRemoteIfMtu	The remote interface MTU as received from the remote node via the maintenance protocol. This object should be 0 if this parameter is not available or not used.
cpwVcRemoteIfString	Indicates the interface description string as received by the maintenance protocol; it must be a NULL string if not applicable or not known yet.
cpwVcOutboundVcLabel	The VC label used in the outbound direction toward the PSN. This object may be set up manually if the owner is manual; otherwise, it is automatic. Examples; for MPLS PSN, the label represents the 20 bits of the VC tag; for L2TP, it represents the 32 bits of the session ID. If the label is not yet known (signaling in process), the object should return a value of 0xFFFF.
cpwVcInboundVcLabel	The VC label used in the inbound direction for packets received from the PSN. This object may be set up manually if the owner is manual; otherwise, it is automatic. Examples; for MPLS PSN, the label represents the 20 bits of VC tag; for L2TP, the label represents the 32 bits of the session ID. If the label is not yet known (signaling in process), the object should return a value of 0xFFFF.
cpwVcName	The canonical name assigned to the VC.
cpwVcDescr	A textual string containing information about the VC. If there is no description, this object contains a 0 length string.
cpwVcCreateTime	System time when this VC was created.
cpwVcUpTime	Number of consecutive ticks that this VC has been up in both directions together. (Up is observed in cpwVcOperStatus.)
cpwVcAdminStatus	The desired operational status of this VC.

Objects	Description
cpwVcOperStatus	Indicates the actual combined operational status of this VC. This object is up if both cpwVcInboundOperStatus and cpwVcOutboundOperStatus are in the up state. For all other values, if the VCs in both directions are of the same value, this object reflects that value; otherwise, it is set to the more severe status of the two. The order of severity from most severe to less severe is as follows: unknown, notPresent, down, lowerLayerDown, dormant, testing, and up. The operator can consult the direction of OperStatus for fault isolation.
cpwVcInboundOperStatus	Indicates the actual operational status of this VC in the inbound direction. Values are the following: • up--The VC is established and ready to pass packets. • down--PW signaling has not yet finished or indications available at the service level show that the VC is not passing packets. • testing--AdminStatus at the VC level is set to test. • dormant--The VC is not available because the required resources are occupied by higher priority VCs. • notPresent--Some component needed for the setup of the VC is missing. • lowerLayerDown--The underlying PSN is not in OperStatus up.
cpwVcOutboundOperStatus	Indicates the actual operational status of this VC in the outbound direction. Values are the following: • up--The VC is established and ready to pass packets. • down--PW signaling has not yet finished or indications available at the service level show that the VC is not passing packets. • testing--AdminStatus at the VC level is set to test. • dormant--The VC is not available because the required resources are occupied by higher priority VCs. • notPresent--Some component needed for the setup of the VC is missing. • lowerLayerDown--The underlying PSN is not in OperStatus up.
cpwVcTimeElapsed	The number of seconds, including partial seconds, that have elapsed since the beginning of the current measurement period. If, for some reason, such as an adjustment in the system's time-of-day clock, and the current interval exceeds the maximum value, the agent returns the maximum value. Because cpwVcPerfIntervalTable is not implemented, this is 0.

Objects	Description
cpwVcValidIntervals	The number of previous 15-minute intervals for which data was collected. An agent with PW capability must be capable of supporting at least *x*intervals. The minimum value of *x*is 4; the default of *x*is 32, and the maximum value of *x*is 96. The value is *x*unless the measurement was (re)started within the last *x**15 minutes, in which case the value will be the number of complete 15-minute intervals; for example, in the case where the agent is a proxy, some intervals may be unavailable. In this case, this interval is the maximum interval number for which data is available. This interval is set to 0.
cpwVcRowStatus	A read-only implementation that is always active(1). It is used for creating, modifying, and deleting.
cpwVcStorageType	The storage type for this object is a read-only implementation that is always volatile(2).

cpwVcPerfTotalTable

The table below lists the cpwVcPerfTotalTable objects and their descriptions.

Table 58: cpwVcPerfTotalTable Objects and Descriptions

Objects	Description
cpwVcPerfTotalInHCPackets	High-capacity counter for the number of packets received by the VC from the PSN.
cpwVcPerfTotalInHCBytes	High-capacity counter for the number of bytes received by the VC from the PSN.
cpwVcPerfTotalOutHCPackets	High-capacity counter for the number of packets forwarded by the VC to the PSN.
cpwVcPerfTotalOutHCBytes	High-capacity counter for the number of bytes forwarded by the VC (to the PSN).
cpwVcPerfTotalDiscontinuityTime	The value of sysUpTime on the most recent occasion when one or more of this object's counters suffered a discontinuity. The relevant counters are the specific instances of any Counter32 or Counter64. If no such discontinuities have occurred since the last reinitialization of the local management subsystem, this object contains a 0 value.

cpwVcIdMappingTable

The table below lists the cpwVcIdMappingTable objects and their descriptions.

Table 59: cpwVcIdMappingTable Objects and Descriptions

Objects	Description
cpwVcIdMappingVcType	The VC type (indicates the service) of this VC.

Objects	Description
cpwVcIdMappingVcID	The VC ID of this VC; 0 if the VC is configured manually.
cpwVcIdMappingPeerAddrType	IP address type of the peer node.
cpwVcIdMappingPeerAddr	IP address of the peer node.
cpwVcIdMappingVcIndex	The value that represents the VC in the cpwVcTable.

cpwVcPeerMappingTable

The table below lists the cpwVcPeerMappingTable objects and their descriptions.

Table 60: cpwVcPeerMappingTable Objects and Descriptions

Objects	Description
cpwVcPeerMappingPeerAddrType	IP address type of the peer node.
cpwVcPeerMappingPeerAddr	IP address of the peer node.
cpwVcPeerMappingVcType	The VC type (indicates the service) of this VC.
cpwVcPeerMappingVcID	The VC ID of this VC; 0 if the VC is configured manually.
cpwVcPeerMappingVcIndex	The value that represents the VC in the cpwVcTable.

Tables in the PW-MPLS-MIB

The PW-MPLS-MIB consists of the following tables:

- cpwVcMplsTable -- Specifies information for the VC to be carried over an MPLS PSN. This table is indexed on cpwVcIndex.
- cpwVcMplsOutboundTable -- Associates VCs using an MPLS PSN with the outbound MPLS tunnels toward the PSN or the physical interface in the case of the VC only. A row in this table represents a link between PW VCs that require MPLS tunnels and an MPLS tunnel toward the PSN. This table is indexed by the cpwVcIndex and an additional index that is not supported; consequently, its value is 1. The operator creates at least one entry in this table for each PW VC that requires an MPLS PSN. The VC-only case and the cpwVcMplsOutboundIndex is not supported.
- cpwVcMplsInboundTable -- Associates VCs using an MPLS PSN with the inbound MPLS tunnels for packets coming from the PSN, if such association is desired mainly for security reasons. A row in this table represents a link between PW VCs that require MPLS tunnels and an MPLS tunnel for packets arriving from the PSN. This table is indexed by the set of indexes used to identify the VC, cpwVcIndex, and an additional index that is not supported; consequently, its value is 1. An entry is created in this table either automatically by the local agent or manually by the operator when strict mode is required. This table points to the appropriate MPLS MIB. For MPLS-TE, the four variables relevant to the indexing of an MPLS TE tunnel are set. The VC-only case and the cpwVcMplsInboundIndex are not supported.
- cpwVcMplsNonTeMappingTable -- Maps an inbound or outbound tunnel to a VC in non-TE applications. A row in this table represents the association between a PW VC and its non-TE MPLS outer tunnel. An

application can use this table to retrieve the PW carried over a specific non-TE MPLS outer tunnel quickly. This table is indexed by the xconnect index for the MPLS non-TE tunnel and the direction of the VC in the specific entry. The same table is used in both inbound and outbound directions, but in a different row for each direction. If the inbound association is not known, no rows should exist for it. Rows are created by the local agent when all the association data is available for display.

- cpwVcMplsTeMappingTable -- Maps an inbound or outbound tunnel to a VC in MPLS-TE applications. A row in this table represents the association between a PW VC and its MPLS-TE outer tunnel. An application can use this table to retrieve the PW carried over a specific TE MPLS outer tunnel quickly. This table is indexed by the four indexes of a TE tunnel, the direction of the VC specific entry, and the VcIndex. The same table is used in both inbound and outbound directions, but a different row for each direction. If the inbound association is not known, no rows should exist for it. Rows are created by the local agent when all the association data is available for display. This table shows mappings between pseudowires and the xconnect index for non-TE outer tunnel or index.

cpwVcMplsTable

The table below lists the cpwVcMplsTable objects and their descriptions.

Table 61: cpwVcMplsTable Objects and Descriptions

Objects	Description
cpwVcMplsMplsType	Set by the operator to indicate the outer tunnel types, if they exist. Values are the following: • mplsTe(0)--Used when the outer tunnel is set up by MPLS-TE. • mplsNonTe(1)--Used when the outer tunnel is set up by LDP or manually.
cpwVcMplsExpBitsMode	Set by the operator to indicate the way the VC shim label EXP bits are to be determined. The value is the following: • outerTunnel(1)--Used when there is an outer tunnel and cpwVcMplsMplsType is mplsTe or mplsNonTe.
cpwVcMplsExpBits	Set by the operator to indicate the MPLS EXP bits to be used on the VC shim label if cpwVcMplsExpBitsMode is specified; value = 0.
cpwVcMplsTtl	Set by the operator to indicate the VC time-to-live (TTL) bits to be used on the VC shim label; value = 0.
cpwVcMplsLocalLdpID	The local LDP identifier of the LDP entity creating this VC in the local node. Because the VC labels are always set from the per-platform label space, the last two octets in the LDP ID must be 0s.
cpwVcMplsLocalLdpEntityID	The local LDP entity index of the LDP entity to be used for this VC on the local node; this should be set to all 0s when this object is not used.
cpwVcMplsPeerLdpID	The peer LDP identifier as identified by the LDP session; this should be zero if not relevant or not known yet.

Objects	Description
cpwVcMplsStorageType	The storage type for this object is a read-only implementation that is always volatile(2).

cpwVcMplsOutboundTable

The table below lists the cpwVcMplsOutboundTable objects and their descriptions.

Table 62: cpwVcMplsOutboundTable Objects and Descriptions

Objects	Description
cpwVcMplsOutboundIndex	An arbitrary index for enabling multiple rows per VC in this table. The next available free index can be retrieved using cpwVcMplsOutboundIndexNext. The value = 1, because this object is not supported.
cpwVcMplsOutboundLsrXcIndex	Set by the operator. If the outer label is defined in the MPL-LSR-MIB, that is, set by LDP or manually, this object points to the xconnect index of the outer tunnel. Otherwise, this object is set to 0.
cpwVcMplsOutboundTunnelIndex	Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to 0.
cpwVcMplsOutboundTunnelInstance	Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to 0.
cpwVcMplsOutboundTunnelLclLSR	Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to NULL.
cpwVcMplsOutboundTunnelPeerLSR	Part of the set of indexes for an outbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to NULL.
cpwVcMplsOutboundIfIndex	For a VC only with no outer tunnel, this object holds the ifIndex of the outbound port. The value = 0.
cpwVcMplsOutboundRowStatus	A read-only implementation that is always active(1). It is used for creating, modifying, and deleting.
cpwVcMplsOutboundStorageType	The storage type for this object is a read-only implementation that is always volatile(2).

cpwVcMplsInboundTable

The table below lists the cpwVcMplsInboundTable objects and their descriptions.

Table 63: cpwVcMplsInboundTable Objects and Descriptions

Objects	Description
cpwVcMplsInboundIndex	An arbitrary index for enabling multiple rows per VC in this table. The next available free index can be retrieved using cpwVcMplsInboundIndexNext. the value = 1, because this object is not supported.
cpwVcMplsInboundLsrXcIndex	If the outer label is defined in the MPL-LSR-MIB; that is, set by LDP or manually, this object points to the xconnect index of the outer tunnel. The xconnect index represents the pseudowire in the inbound direction retrieving 0 if information for this object is not known.
cpwVcMplsInboundTunnelIndex	Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; value = 0. This object does not support TE tunnels at the ingress router.
cpwVcMplsInboundTunnelInstance	Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; value = 0. This object does not support TE tunnels at the ingress router.
cpwVcMplsInboundTunnelLclLSR	Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, set to NULL. This object does not support TE tunnels at the ingress router.
cpwVcMplsInboundTunnelPeerLSR	Part of the set of indexes for an inbound tunnel, specifically an MPLS-TE outer tunnel; otherwise, this object is set to NULL. This object does not support TE tunnels at the ingress router.
cpwVcMplsInboundIfIndex	In the case of a VC only (no outer tunnel), this object holds the ifIndex of the inbound port. The value = 0.
cpwVcMplsInboundRowStatus	A read-only implementation that is always active(1). It is used for creating, modifying, and deleting.
cpwVcMplsInboundStorageType	The storage type for this object is a read-only implementation that is always volatile(2).

cpwVcMplsNonTeMappingTable

The table below lists the cpwVcMplsNonTeMappingTable objects and their descriptions.

Table 64: cpwVcMplsNonTeMappingTable Objects and Descriptions

Objects	Description
cpwVcMplsNonTeMappingTunnelDirection	Identifies if the row represents an outbound or inbound mapping.
cpwVcMplsNonTeMappingXcTunnelIndex	XC index in the MPLS-LSR-MIB of the pseudowire LDP-generated XC entry.

Objects	Description
cpwVcMplsNonTeMappingIfIndex	Identifies the port on which the VC is carried for VC only; the value = 0.
cpwVcMplsNonTeMappingVcIndex	Represents the VC in the cpwVcTable.

cpwVcMplsTeMappingTable

The table below lists the cpwVcMplsTeMappingTable objects and their descriptions.

Table 65: cpwVcMplsTeMappingTable Objects and Descriptions

Objects	Description
cpwVcMplsTeMappingTunnelDirection	Identifies if the row represents an outbound mapping.
cpwVcMplsTeMappingTunnelIndex	Index for the conceptual row identifying an MPLS-TE tunnel.
cpwVcMplsTeMappingTunnelInstance	Identifies an instance of an MPLS-TE tunnel.
cpwVcMplsTeMappingTunnelPeerLsrID	Identifies a peer LSR when the outer tunnel is MPLS-TE based.
cpwVcMplsTeMappingTunnelLocalLsrID	Identifies the local LSR.
cpwVcMplsTeMappingVcIndex	Represents the VC in the cpwVcTable.

Tables in the PW-ENET-MIB

The PW-ENET-MIB consists of the following table:

- cpwVcEnetTable -- Provides Ethernet port mapping and VLAN configuration for each Ethernet emulated virtual connection. This table is indexed on cpwVcIndex, which uniquely identifies a singular connection. The second level index for this table is cpwVcEnetPwVlan, which indicates VLANs on this VC. This table is used only for Ethernet VC types--ethernetVLAN, ethernet, or ethernet virtual private LAN service (VPLS), and is implemented as read-only.

cpwVcEnetTable

The table below lists the cpwVcEnetTable objects and their descriptions.

Table 66: cpwVcEnetTable Objects and Descriptions

Objects	Description
cpwVcEnetPwVlan	The VLAN value for frames on a VC. This is one of the indexes to the table so multiple VLAN values can be configured for a PW VC. This value is 4096 to indicate untagged frames; that is, if the cpwVcEnetVlanMode value is removeVlan. This value is the VLAN value of the access circuit if the cpwVcEnetVlanMode value is noChange. The value of 4097 is used if the object is not applicable; for example, when mapping all packets from an Ethernet port to the VC.

Objects	Description
cpwVcEnetVlanMode	Indicates the way the VLAN field is handled between the access circuit and the PW VC. The possible values for this field are as follows: • noChange--Indicates that the VC contains the original user VLAN, as specified in cpwVcEnetPortVlan. • changeVlan--Indicates that the VLAN field on the VC may be different from the VLAN field on the user's port. • removeVlan--Indicates that the encapsulation on the VC does not include the original VLAN field.
cpwVcEnetPortVlan	Defines the VLAN value on the physical port (or VPLS virtual port) if a change is required to the VLAN value between the VC and the physical or virtual port. It is equal to cpwVcEnetPwVlan if the cpwVcEnetVlanMode value is noChange. A value of 4096 indicates that no VLAN is associated with the VC; that is, assigning Default VLAN to untagged frames. If all traffic from the VC is being forwarded to the port, then this value is 4097 indicating it is not relevant.
cpwVcEnetPortIfIndex	The ifIndex value of the Ethernet port associated with this PW VC for point-to-point Ethernet service. For VPLS, this value is an ifIndex value for a virtual interface for the VPLS instance.
cpwVcEnetVcIfIndex	Models the VC as a virtual interface in the ifTable. This value is always 0 to indicate no virtual interface is created.
cpwVcEnetRowStatus	A read-only implementation that is always active(1). It is used for creating, modifying, and deleting.
cpwVcEnetStorageType	The storage type for this object is a read-only implementation that is always volatile(2).

Tables in the PW-FR-MIB

The PW-FR-MIB consists of the following table:

- cpwVcFrTable -- Contains entries that represent an FRoPW connection operating in one-to-one mapping mode in which there is a one-to-one correspondence between a Frame Relay VC and a pair of unidirectional pseudowires.

cpwVcFrTable

The table below lists the cpwVcFrTable objects and their descriptions.

Table 67: cpwVcFrTable Objects and Descriptions

Objects	Description
cpwVcFrIfIndex	Returns the interface ifIndex of the Frame Relay (FR) segment of the FRoPW connection.

Objects	Description
cpwVcFrDlci	Returns the data-link connection identifier (DLCI) of the Frame Relay segment of an FRoPW connection.
cpwVcFrAdminStatus	Returns the administrative status of an FRoPW connection.
cpwVcFrOperStatus	Returns the combined operational status of an FRoPW connection.
cpwVcFrPw2FrOperStatus	Returns the operational status of the PW-to-FR direction in an FRoPW connection.
cpwVcFrRowStatus	A read-only implementation that is always active(1). It is used for creating, modifying, and deleting.
cpwVcFrStorageType	The storage type for this object is a read-only implementation that is always volatile(2).

Tables in the PW-ATM-MIB

The PW-ATM-MIB consists of the following tables:

- cpwVcAtmTable -- Specifies information for an ATM VC to be carried over the PSN.
- cpwVcAtmPerfTable -- Specifies performance-related attributes for an ATM VC.

cpwVcAtmTable

The table below lists the cpwVcAtmTable objects and their descriptions.

Table 68: cpwVcAtmTable Objects and Descriptions

Objects	Description
cpwAtmIf	Specifies the ATM interface that sends and receives cells from the ATM network.
cpwAtmVpi	Specifies the VPI value of the ATM VC.
cpwAtmVci	Specifies the VCI value of the ATM VC.
cpwAtmClpQosMapping	Indicates the presence of cell loss priority (CLP) bits determining the value in quality of service (QoS) fields of the encapsulating protocol. The value could be used only for outbound traffic, which means traffic going out to the PSN.
cpwAtmRowStatus	A read-only implementation that is always active(1). It is used for creating, modifying, and deleting.
cpwAtmOamCellSupported	Indicates whether operation, administration, and maintenance (OAM) cells are transported on this VC.
cpwAtmQosScalingFactor	Represents the scaling factor to be applied to ATM QoS rates when calculating QoS rates for the PSN domain.

Objects	Description
cpwAtmCellPacking	Identifies if the VC is configured to do cell packing.
cpwAtmMncp	Identifies the number of cells that need to be packed.
cpwAtmEncap	Provides information on whether MPLS or Layer 2 Tunneling Protocol Version 3 (L2TPv3) is used as the transport.
cpwAtmPeerMncp	Represents the maximum number of cells that can be packed in one packet for a peer interface.
cpwAtmMcptTimeout	Represents the maximum cell packing timeout (MCPT) value used.

cpwVcAtmPerfTable

The table below lists the cpwVcAtmPerfTable objects and their descriptions.

Table 69: cpwVcAtmPerfTable Objects and Descriptions

Objects	Description
cpwAtmCellsReceived	Obtains information on the number of cells that were received and sent to the PSN.
cpwAtmCellsSent	Provides information on the number of cells sent to the ATM network.
cpwAtmCellsRejected	Indicates the number of cells that were rejected by this VC because of policing.
cpwAtmCellsTagged	Indicates the number of cells that were tagged.
cpwAtmHCCellsReceived	Provides the high-capacity counter for the number of cells received by this VC.
cpwAtmHCCellsRejected	Provides the high-capacity counter for the number of cells rejected by this VC.
cpwAtmHCCellsTagged	Provides the high-capacity counter for number of cells that were tagged.
cpwAtmAvgCellsPacked	Provides the average number of cells that were packed.
cpwAtmPktsReceived	Indicates the number of ATM AAL5 packets that are actually sent into the ATM network as packets when the VC is configured to do AAL5 over PW.
cpwAtmPktsSent	Gets the number of packets that are reconstructed from the cells, assigns a VC label, and sends the packets into the PSN.
cpwAtmPktsRejected	Indicates the number of packets that were rejected because of policing.

Objects in the PWE3 MIBs

The PWE3 MIBs represent an ASN.1 notation reflecting specific components of the pseudowire services. The MIBs enable a network management application using SNMP to get this information for display. The MIBs support the standard GETNEXT and GETBULK functionality, but do not support configuration capabilities (via SET) in the current implementation.

Scalar Objects in the PWE3 MIBs

The PWE3 MIBs contain the following supported scalar object:

- cpwVcUpDownNotifEnable--This object reflects the configuration of the cpwVcUp and cpwVcDown notifications. If either of the notifications is configured via the command-line interface (CLI), then this object has a value of true(1). If this object is set via SNMP to true(1), then it enables the emission of both the cpwVcUp and cpwVcDown notifications; if the object is set via SNMP to false(2), these notifications are not emitted.

Note cpwVcUpDownNotifEnable can be set only if RW is configured for the **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*] command.

The PWE3 MIBs contain the following unsupported scalar objects:

- cpwVcIndexNext--Indicates the next cpwVcIndex value to use when you add rows to the cpwVcTable.
- cpwVcNotifRate--Indicates the rate at which cpwVcUp/Down notifications can be issued from the device.
- cpwVcMplsOutboundIndexNext--Contains an appropriate value to be used for cpwVcMplsOutboundIndex when you create entries in the cpwVcMplsOutboundTable. The value 0 indicates that no unassigned entries are available. To obtain the cpwVcMplsOutboundIndex value for a new entry, the manager issues a management protocol retrieval operation to obtain the current value of this object. After each retrieval, the software agent should modify the value to the next unassigned index; however, the software agent *must not* assume such retrieval will be done for each row created.
- cpwVcMplsInboundIndexNext--Contains an appropriate value to be used for cpwVcMplsInboundIndex when you create entries in the cpwVcMplsInboundTable. The value 0 indicates that no unassigned entries are available. To obtain the cpwVcMplsInboundIndex value for a new entry, the manager issues a management protocol retrieval operation to obtain the current value of this object. After each retrieval, the software agent should modify the value to the next unassigned index; however, the agent *must not* assume such retrieval will be done for each row created.

Notifications in the PWE3 MIBs

The cpwVcUp and cpwVcDown notifications in the PW-MIB indicate when the operStatus values for a range of PW VCs have changed state.

The definition of these objects in the PW-MIB indicates that events of the same type, either up or down, must be able to be correlated into ranges. The implementation of these notifications does not do any of this correlation. A notification is generated for each individual VC that has an operational state change if that notification is enabled. A notification does not signal an operational state change for a group of VCs as described in the MIB.

Benefits of the PWE3 MIBs

The PWE3 MIBs provide the ability to manage pseudowire emulation edge-to-edge by providing MPLS-related information about the service and a mechanism to monitor the Ethernet, Frame Relay, or ATM access circuits.

How to Configure Pseudowire Emulation Edge-to-Edge MIBs

Enabling the SNMP Agent for the PWE3 MIBs

SUMMARY STEPS

1. **enable**
2. **show running-config** [**interface** | **map-class**]
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
5. **end**
6. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** [**interface**	**map-class**] **Example:** `Router# show running-config`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired. • The optional **interface** keyword displays interface-specific configuration information. • The optional **map-class** keyword displays dialer or Frame Relay map-class information.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro**	**rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Router(config)# snmp-server community public ro`	Sets up the community access string to permit access to SNMP for the MIBs. • The *string* argument consists of 1 to 32 alphanumeric characters and functions much like a password, permitting access to SNMP. Blank spaces are not permitted in the community string.

	Command or Action	Purpose
		• The optional **view** *view-name* keyword argument combination specifies a previously defined view. The view defines the objects available to the SNMP community. • The optional **ro** keyword configures read-only (ro) access to the objects in the MIBs. • The optional **rw** keyword specifies read-write access. Authorized management stations can both retrieve and modify MIB objects. • The optional **ipv6** *nacl* keyword argument combination specifies an IPv6 named access list. • The optional *access-list-number* argument is an integer from 1 to 99 that specifies a standard access list of IP addresses or a string (not to exceed 64 characters) that is the name of a standard access list of IP addresses allowed access to the SNMP agent. Alternatively, it is an integer from 1300 to 1999 that specifies a list of IP addresses in the expanded range of standard access list numbers that are allowed to use the community string to gain access to the SNMP agent.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.

Configuring the Pseudowire Class

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You configure the connection, called a pseudowire, between the routers.

Note In simple configurations, this task is optional. You do not need to specify a pseudowire class if you specify the tunneling method as part of the **xconnect**command.

The pseudowire-class configuration group specifies the following characteristics of the tunneling mechanism:

- Encapsulation type
- Control protocol

- Payload-specific options

You must specify the **encapsulation mpls** command as part of the pseudowire class or as part of the **xconnect** command for the AToM VCs to work properly. If you omit the **encapsulation mpls** command as part of the **xconnect**command, you receive the following error:

```
% Incomplete command.
```

Once you specify the **encapsulation mpls** command, you cannot remove it using the **no encapsulation mpls**command. Nor can you change the command's setting using the **encapsulation l2tpv3** command. Those methods result in the following error message:

```
Encapsulation changes are not allowed on an existing pw-class.
```

To remove the command, you must delete the pseudowire with the **no pseudowire-class** command. To change the type of encapsulation, remove the pseudowire with the **no pseudowire-class** command and reestablish the pseudowire and specify the new encapsulation type.

Note There are many options that you can configure. For detailed information, see the "Any Transport over MPLS" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *name* **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:**	Specifies the tunneling encapsulation. For AToM, the encapsulation type is mpls.

	Command or Action	Purpose
	`Router(config-pw)# encapsulation mpls`	

What to Do Next

Perform a MIB walk using your SNMP management tool on cpwVcMIB, cpwVcMplsMIB, cpwVcEnetMIB, cpwVcFrMIB, and cpwVcAtmMIB to verify that the PW-MIB, the PW-MPLS-MIB, the PW-ENET-MIB, the PW-FR-MIB, and the PW-ATM-MIB objects, respectively, are populated correctly.

Note SNMPv1 and SNMPv2c are supported.

Configuration Examples for the Pseudowire Emulation Edge-to-Edge MIBs

PWE3 MIBs Example

In the following example, the configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public* .

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# snmp-server community public ro
```

Note There is no explicit way to configure the PWE3 MIBs. However, for information on AToM configuration tasks and examples, see the "Any Transport over MPLS" module.

There are notifications specific to the PWE3 MIBs. For detailed information on the commands used to configure them, see the "Additional References" section.

Additional References

Related Documents

Related Topic	Document Title
Description of commands associated with MPLS and MPLS applications	*Multiprotocol Label Switching Command Reference*
AToM and MPLS	“Any Transport over MPLS” module

Related Topic	Document Title
Pseudowire-related Internet drafts	• *An Architecture for Multi-Segment Pseudo Wire Emulation Edge-to-Edge*, Internet draft, December 2007 [draft-ietf-pwe3-ms-arch-03.txt] • Definitions for Textual Conventions and OBJECT-IDENTITIES for Pseudo-Wires Management, Internet draft, August 10, 2007 [draft-ietf-pwe3-pw-tc-mib-09.txt] • Ethernet Pseudo Wire (PW) Management Information Base, Internet draft, August 30, 2007 [draft-pwe3-enet-mib-10.txt] • *Managed Objects for ATM over Packet Switched Network (PSN)* , Internet draft, August 8, 2007 [draft-ietf-pwe3-pw-atm-mib-02.txt] • Pseudo Wire (PW) Management Information Base, Internet draft, May 31, 2007 [draft-ietf-pwe3-pw-mib-11.txt] • Pseudo Wire (PW) over MPLS PSN Management Information Base, Internet draft, August 11, 2007 [draft-ietf-pwe3-pw-mpls-mib-11.txt] **Note** For information on using SNMP MIB features, see the appropriate documentation for your network management system.

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
SNMP-VACM-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1156	*Management Information Base for Network Management of TCP/IP-based Internets*
RFC 1157	*A Simple Network Management Protocol (SNMP)*
RFC 1213	*Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*
RFC 1315	*Management Information Base for Frame Relay DTEs*

RFC	Title
RFC 3815	*Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)*
RFC 3916	*Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*
RFC 4619	*Encapsulation Methods for Transport of Frame Relay over Multiprotocol Label Switching (MPLS) Networks*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Pseudowire Emulation Edge-to-Edge MIBs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 70: Feature Information for Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services

Feature Name	Releases	Feature Information
Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services	Cisco IOS Release XE 2.3	The Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services feature provides Simple Network Management Protocol (SNMP) support within an Any Transport over Multiprotocol Label Switching (AToM) infrastructure emulating Ethernet, Frame Relay, and ATM services over packet switched networks (PSNs). In Cisco IOS Release XE 2.3, this feature was integrated into the releases as the Pseudowire Emulation Edge-to-Edge (PWE3) MIBs providing SNMP support within an Any Transport over Multiprotocol Label Switching (AToM) infrastructure emulating Ethernet, Frame Relay, and ATM services over packet switched networks (PSNs).

Glossary

AAL —ATM adaptation layer. AAL defines the conversion of user information into cells. AAL1 and AAL2 handle isochronous traffic, such as voice and video; AAL3/4 and AAL5 pertain to data communications through the segmentation and reassembly of packets.

ATM —asynchronous transfer mode. A cell-based data transfer technique in which channel demand determines packet allocation. This is an international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as E3, SONET, and T3.

CE router—customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

DLCI —data-link connection identifier. A unique number assigned to a PVC endpoint in a Frame Relay network. Identifies a particular PVC endpoint within an access channel in a Frame Relay network and has local significance only to that channel.

encapsulation —Wrapping of data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when bridging occurs in dissimilar networks, the entire frame from one network is simply placed in the header used by the data link layer protocol of the other network.

EoMPLS —Ethernet over multiprotocol label switching (MPLS). A tunneling mechanism that allows a service provider to tunnel customer Layer 2 traffic through a Layer 3 MPLS network. EoMPLS is a point-to-point solution only. EoMPLS is also known as Layer 2 tunneling.

Frame Relay—The industry standard, switched data link layer protocol that handles multiple virtual circuits using High-Level Data Link Control (HDLC) encapsulation between connected devices. Frame Relay is more efficient than X.25, the protocol for which it is generally considered a replacement.

IETF —internet engineering task force. A task force (consisting of more than 80 working groups) that is developing standards for the Internet and the IP suite of protocols.

LDP —label distribution protocol. The protocol that supports MPLS hop-by-hop forwarding and the distribution of bindings between labels and network prefixes. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LSP —label switched path. A configured connection between two label switch routers (LSRs) in which label-switching techniques are used for packet forwarding; also a specific path through an MPLS network.

LSR —label switch router. A Multiprotocol Label Switching (MPLS) node that can forward native Layer 3 packets. The LSR forwards a packet based on the value of a label attached to the packet.

MIB—management information base. A database of network management information that is used and maintained by a network management protocol such as simple network management protocol (SNMP). The value of a MIB object can be changed or retrieved by using SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS—multiprotocol label switching. A switching method for the forwarding of IP traffic through the use of a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

MTU —maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

NMS—network management system. System responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. An NMS communicates with agents to help keep track of network statistics and resources.

notification —A message sent by a Simple Network Management Protocol (SNMP) agent to a network management station, console, or terminal to indicate that a significant network event has occurred. See also trap.

OSPF —Open Shortest Path First. A link-state hierarchical Interior Gateway Protocol routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

PE router—provider edge router. A router that is part of a service provider's network and is connected to a customer edge (CE) router.

primary tunnel—A tunnel whose label-switched path (LSP) may be fast rerouted if there is a failure. Backup tunnels cannot be primary tunnels.

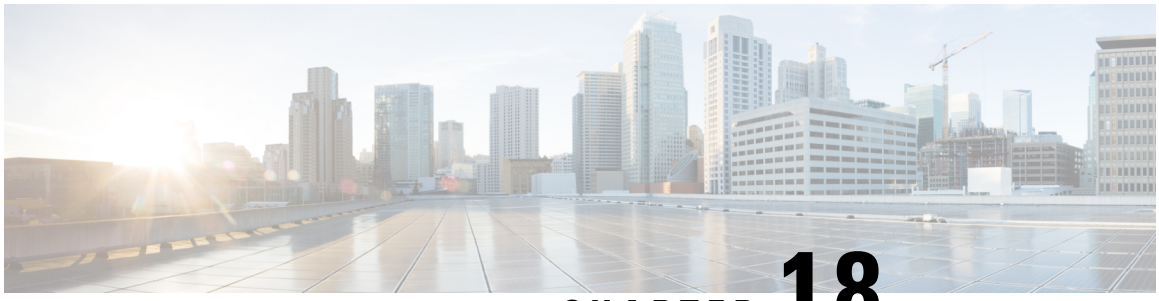
pseudowire —PW. A mechanism that carries the elements of an emulated service from one provider edge (PE) to one or more PEs over a packet switched network (PSN).

SNMP—simple network management protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices, and for managing configurations, statistics collection, performance, and security.

trap—A message sent by an SNMP agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps are less reliable than notification requests because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received.

tunnel —A secure communication path between two peers, such as routers.

VC —virtual circuit. A logical circuit created to ensure reliable communication between two network devices. A virtual circuit can be either permanent (PVC) or switched (SVC).



CHAPTER 18

MPLS Traffic Engineering--Fast Reroute MIB

The MPLS Traffic Engineering--Fast Reroute MIB provides Simple Network Management Protocol (SNMP)-based network management of the Multiprotocol Label Switching (MPLS) Fast Reroute (FRR) feature in Cisco software.

The Fast Reroute MIB has the following features:

- Notifications can be created and queued.
- Command-line interface (CLI) commands enable notifications, and specify the IP address to where the notifications will be sent.
- The configuration of the notifications can be written into nonvolatile memory.

The MIB includes objects describing features within MPLS FRR, and it includes the following tables:

- cmplsFrrConstTable
- cmplsFrrLogTable
- cmplsFrrFacRouteDBTable

The MIB also includes scalar objects (that is, objects that are not in a table). For more information, see theMPLS Traffic Engineering--Fast Reroute MIB, on page 369 .

Prerequisites for the MPLS Traffic Engineering--Fast Reroute MIB

- The network must support the Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF) protocol.

- The SNMP is installed and enabled on the label switch routers (LSRs).
- MPLS is enabled globally on each LSR.
- Cisco Express Forwarding is enabled on the LSRs.
- Traffic engineering (TE) tunnels are enabled.
- MPLS FRR is enabled on one of the TE tunnels.
- The Resource Reservation Protocol (RSVP) is enabled.

Restrictions for the MPLS Traffic Engineering--Fast Reroute MIB

- The implementation of the FRR MIB is limited to read-only (RO) permission for MIB objects.
- The following tables are not implemented:
  - mplsFrrOne2OnePlrTable
  - mplsFrrDetourTable.

Information About the MPLS Traffic Engineering--Fast Reroute MIB

Feature Design of the MPLS Traffic Engineering--Fast Reroute MIB

The FRR MIB enables standard, SNMP-based network management of FRR in Cisco software. This capability requires that SNMP agent code executes on a designated network management station (NMS) in the network. The NMS serves as the medium for user interaction with the network management objects in the MIB.

The FRR MIB is based on the Internet Engineering Task Force (IETF) draft MIB specification *draft-ietf-mpls-fastreroute-mib-02.txt* . The IETF draft MIB, which undergoes revisions periodically, is evolving toward becoming a standard. The Cisco implementation of the FRR MIB is expected to track the evolution of the IETF draft MIB, and may change accordingly.

Slight differences between the IETF draft MIB and the implementation of FRR within Cisco software require some minor translations between the FRR MIB objects and the internal data structures of Cisco software. These translations are accomplished by the SNMP agent, which runs in the background on the NMS workstation as a low priority process and provides a management interface to Cisco software.

You can use an SNMP agent to access FRR MIB objects using standard SNMP GET operations. All the objects in the FRR MIB follow the conventions defined in the IETF draft MIB.

Functional Structure of the MPLS Traffic Engineering--Fast Reroute MIB

The SNMP agent code supporting the FRR MIB follows the existing model for such code in Cisco software and is, in part, generated by the Cisco tool set, based on the MIB source code. The basis for the generated code is the Cisco version of the FRR MIB CISCO-ietf-frr-mib.

The SNMP agent code, which has a layered structure that is common to MIB support code in Cisco software, consists of the following layers:

- Platform-independent layer--This layer is generated primarily by the MIB development Cisco tool set and incorporates platform- and implementation-independent functions. These functions handle SNMP standard functionality in the context of the specific MIB. This layer handles indexes and range or enumeration value checks for GET, GET-NEXT, and SET SNMP operations. A function is generated for each SNMP table or group of objects. This layer calls into the next layer.
- Application interface layer--The Cisco tool set generates the function names and template code for MIB objects.
- Application-specific layer--This layer provides the mechanism for retrieving relevant data from the managed application layer. It includes an entry point function for each table. This function calls two other functions; one that searches the TE tunnel database that RSVP maintains for the relevant data according to the indexes, and another function that fills the data into the structure.
- Managed application layer--This layer includes all the structures and mechanisms, and is managed by the MIB.

System Flow of SNMP Protocol Requests and Response Messages

All SNMP protocol requests and response messages are ultimately handled by the SNMP master agent. When such a message is received on a router, the master agent parses the requests and identifies the MIB to which the request refers. The master agent then queries the subagent responsible for the MIB with a GET, GET-NEXT, or SET request. The FRR MIB subagent retrieves the appropriate data, and returns it to the master agent. The master agent is then responsible for returning an SNMP response to the NMS. All queries occur within the IP SNMP Cisco software process, which runs as a low priority task.

FRR MIB Scalar Objects

Scalar objects are objects that are not in tables. A scalar object has one instance (that is, one occurrence).

The table below describes the FRR MIB scalar objects.

Table 71: Scalar Objects

MIB Object	Function
cmplsFrrDetourIncoming	Number of detour link-state packets (LSPs) entering the device. This object returns 0 because cmplsFrrConstProtectionMethod is set to facilityBackup(1).
cmplsFrrDetourOutgoing	Number of detour LSPs leaving the device. This object returns 0 because cmplsFrrConstProtectionMethod is set to facilityBackup(1).
cmplsFrrDetourOriginating	Number of detour LSPs originating from the device. This object returns 0 because cmplsFrrConstProtectionMethod is set to facilityBackup(1).

MIB Object	Function
cmplsFrrSwitchover	Number of tunnels that are being backed up because cmplsFrrConstProtectionMethod is set to facilityBackup(1).
cmplsFrrNumOfConfIfs	Number of MPLS interfaces FRR configured for protection; 0 indicates that LSPs traversing any interface can be protected.
cmplsFrrActProtectedIfs	Number of interfaces FRR is protecting because cmplsFrrConstProtectionMethod is set to facilityBackup(1).
cmplsFrrConfProtectingTuns	Number of backup Fast Reroute-protected tunnels configured because cmplsFrrConstProtectionMethod is set to facilityBackup(1).
cmplsFrrActProtectedTuns	Number of tunnels protected by the Fast Reroute feature. This object returns 0 because cmplsFrrConstProtectionMethod is set to facilityBackup(1).
cmplsFrrActProtectedLSPs	Number of LSPs that FRR is protecting. If cmplsFrrConstProtectionMethod is set to facilityBackup(1), this object returns 0.
cmplsFrrConstProtectionMethod	This object always returns facilityBackup(1) because Cisco software supports only the facility backup protection method.
cmplsFrrNotifsEnabled	A value that indicates whether FRR notifications defined in this MIB are enabled or disabled. This object returns True(1) for enabled, or False(2) for disabled. The default is that notifications are disabled.
cmplsFrrLogTableMaxEntries	Maximum number of entries allowed in the FRR log table.
cmplsFrrLogTableCurrEntries	Current number of entries in the FRR log table. This object always returns 0.
cmplsFrrNotifMaxRate	Maximum interval rate between FRR MIB notifications. This object always returns 0.

FRR MIB Notification Generation Events

Notifications are issued after particular FRR events occur.

When you enable FRR MIB notification functionality by issuing the **snmp-server enable traps mpls fast-reroute** command, FRR events generate notification messages that are sent to a designated NMS in the network to signal the occurrence of specific events in Cisco software.

The FRR MIB objects involved in FRR status transitions and event notifications include cmplsFrrProtected. This message is sent to an NMS if there is a major TE tunnel change (that is, fast rerouting of TE tunnels).

FRR MIB Notification Specification

Notifications are issued after particular FRR events occur.

Each FRR notification has a generic type identifier and an enterprise-specific type identifier for identifying the notification type. The generic type for all FRR notifications is “enterprise Specific” because this is not one of the generic notification types defined for SNMP. The enterprise-specific type is 1 for cmplsFrrProtected.

Each notification contains the following objects from the FRR MIB so that the FRR tunnel can be easily identified:

- cmplsFrrConstNumProtectingTunOnIf
- cmplsFrrConstNumProtectedTunOnIf
- cmplsFrrConstBandwidth

Upon being invoked, the appropriate FRR interface indexes have already been retrieved by existing FRR code. The FRR interfaces are then used to fill in data for the three objects included in the notification.

FRR MIB Notification Monitoring

Notifications are issued after particular FRR events occur.

When FRR MIB notifications are enabled (see the **snmp-server enable traps** command), notification messages relating to specific FRR events within Cisco software are generated and sent to a specified NMS in the network. Any utility that supports SNMPv1 or SNPv2 notifications can receive notification messages.

To monitor FRR MIB notifications, log in to an NMS that supports a utility that displays SNMP notifications, and start the display utility.

MIB Tables in the MPLS Traffic Engineering--Fast Reroute MIB

The FRR MIB consists of the following tables:

The tables access various data structures to obtain information regarding detours, the FRR database, and logging.

cmplsFrrConstTable

cmplsFrrConstTable displays the configuration of an FRR-enabled tunnel and the characteristics of its accompanying backup tunnels. For each protected tunnel, there can be multiple backup tunnels.

The table is indexed by the following:

- cmplsFrrConstIfIndex
- cmplsFrrConstTunnelIndex
- cmplsFrrConstTunnelInstance

The table below describes the MIB objects for cmplsFrrConstTable.

Table 72: cmplsFrrConstTable Objects

MIB Object	Function
cmplsFrrConstIfIndex	Uniquely identifies an interface on which FRR is configured. If an index has a value of 0, the configuration applies to all interfaces on the device on which the FRR feature can operate.
cmplsFrrConstTunnelIndex	Tunnel for which FRR is requested.
cmplsFrrConstTunnelInstance	Tunnel for which FRR is requested. The value always is 0 because only tunnel heads are represented, and tunnel heads have an instance value of 0.

MIB Object	Function
cmplsFrrConstSetupPrio	Setup priority of the backup tunnel.
cmplsFrrConstHoldingPrio	Holding priority of the backup tunnel.
cmplsFrrConstInclAnyAffinity	Attribute bits that must be set for the tunnel to traverse a link.
cmplsFrrConstInclAllAffinity	Attribute bits that must not be set for the tunnel to traverse a link.
cmplsFrrConstExclAllAffinity	A link satisfies the exclude-all constraint only if the link contains none of the administrative groups specified in the constraint.
cmplsFrrConstHopLimit	The maximum number of hops that the backup tunnel can traverse.
cmplsFrrConstBandwidth	The bandwidth of the backup tunnels for this tunnel, in thousands of bits per second (kbps).
cmplsFrrConstRowStatus	Creates, modifies, and deletes a row in this table.

cmplsFrrLogTable

cmplsFrrLogTable is indexed by the object cmplsFrrLogIndex. The index corresponds to a log entry in the FRR feature's **show mpls traffic-eng fast-reroute log reroutes** command. That **show** command stores up to 32 entries at a time. If entries are added, the oldest entry is overwritten with new log information.

cmplsFrrLogTable can store up to 32 entries at a time, overwriting older entries as newer ones are added. The index cmplsFrrLogIndex is incremented to give each log table entry of the MIB a unique index value. Therefore, it is possible to have indexes greater than 32 even though only 32 entries are displaying.

The table below describes the MIB objects for cmplsFrrLogTable.

Table 73: cmplsFrrLogTable Objects

MIB Object	Function
cmplsFrrLogIndex	Number of the FRR event.
cmplsFrrLogEventTime	Number of milliseconds that elapsed from bootstrap time to the time that the event occurred.
cmplsFrrLogInterface	Identifies the interface that was affected by this FRR event. The value can be set to 0 if mplsFrrConstProtectionMethod is set to oneToOneBackup(0).
cmplsFrrLogEventType	The type of FRR event that occurred. The object returns Protected or Other.
cmplsFrrLogEventDuration	Duration of the event, in milliseconds.
cmplsFrrLogEventReasonString	Implementation-specific explanation of the event. The object returns interface down event or interface other event.

cmplsFrrFacRouteDBTable

The following indexes specify which interface and tunnel are being protected by the FRR feature:

- cmplsFrrFacRouteProtectedIfIndex

- cmplsFrrFacRouteProtectedTunIndex

The following indexes specify the backup tunnel that provides protection to the protected tunnel:

- cmplsFrrFacRouteProtectedIfIndex
- cmplsFrrFacRouteProtectingTunIndex
- cmplsFrrFacRouteProtectedTunIndex
- cmplsFrrFacRouteProtectedTunInstance
- cmplsFrrFacRouteProtectedTunIngressLSRId
- cmplsFrrFacRouteProtectedTunEgressLSRId

This version of the MIB will attempt to leverage the work already done for the MPLS TE MIB because it contains similar lookup functions for TE tunnels.

The table below describes the MIB objects for cmplsFrrFacRouteDBTable.

Table 74: cmplsFrrFacRouteDBTable Objects

MIB Object	Function
cmplsFrrFacRouteProtectedIfIndex	Interface configured for FRR protection.
cmplsFrrFacRouteProtectingTunIndex	The tunnel number of the protecting (backup) tunnel.
cmplsFrrFacRouteProtectedTunIndex	The mplsTunnelEntry primary index for the tunnel head interface designated to protect the interface specified in mplsFrrFacRouteIfProtIdx (and all the tunnels using this interface).
cmplsFrrFacRouteProtectedTunInstance	An mplsTunnelEntry that is being protected by FRR. An instance uniquely identifies a tunnel.
cmplsFrrFacRouteProtectedTunIngressLSRId	Inbound label for the backup LSR.
cmplsFrrFacRouteProtectedTunEgressLSRId	Outbound label for the backup LSR.
cmplsFrrFacRouteProtectedTunStatus	State of the protected tunnel. Valid values are: • active--Tunnel label has been placed in the Label Forwarding Information Base (LFIB) and is ready to be applied to incoming packets. • ready--Tunnel's label entry has been created, but is not in the LFIB. • partial--Tunnel's label entry has not been fully created.
cmplsFrrFacRouteProtectingTunResvBw	Amount of bandwidth, in megabytes per second, that is reserved by the backup tunnel.
cmplsFrrFacRouteProtectingTunProtectionType	Type of protection: 0 designates link protection; 1 designates node protection.

How to Configure the MPLS Traffic Engineering--Fast Reroute MIB

Enabling the SNMP Agent for FRR MIB Notifications

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro**] [*access-list-number*]
5. **snmp-server enable traps mpls fast-reroute protected**
6. **end**
7. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router to determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify or change the information.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro**] [*access-list-number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) SNMP community strings for the FRR MIB.
Step 5	**snmp-server enable traps mpls fast-reroute protected** **Example:**	Enables Fast Reroute traps.

	Command or Action	Purpose
	`Router(config)# snmp-server enable traps mpls fast-reroute protected`	
Step 6	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 7	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.

Enabling Cisco Express Forwarding

SUMMARY STEPS

1. **enable**
2. **configure teminal**
3. **ip cef distributed**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure teminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling TE Tunnels

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **mpls traffic-eng tunnels**
5. **interface** *type slot/subslot/port*[**.***subinterface*]
6. **mpls traffic-eng tunnels**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Router(config)# ip cef`	Enables standard Cisco Express Forwarding operations.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config)# mpls traffic-eng tunnels`	Enables the MPLS TE tunnel feature on a device.
Step 5	**interface** *type slot/subslot/port*[**.***subinterface*] **Example:** `Router(config)# interface POS1/0/0`	Specifies the interface and enters interface configuration mode.
Step 6	**mpls traffic-eng tunnels** **Example:** `Router(config-if)# mpls traffic-eng tunnels`	Enables the MPLS TE tunnel feature on an interface.
Step 7	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

Enabling MPLS FRR on Each TE Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot/port*[**.***subinterface*]
4. **tunnel mode mpls traffic-eng**
5. **tunnel mpls traffic-eng fast-reroute**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/subslot/port*[**.***subinterface*] **Example:** `Router(config)# interface POS1/0/0`	Specifies the interface and enters interface configuration mode.
Step 4	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the mode of a tunnel to MPLS for traffic engineering.
Step 5	**tunnel mpls traffic-eng fast-reroute** **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute`	Enables Fast Reroute on the TE tunnel being protected.
Step 6	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

Enabling a Backup Tunnel on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *typeslot/subslot/port*[**.***subinterface*]
4. **mpls traffic-eng backup-path tunnel** *interface*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *typeslot/subslot/port*[**.***subinterface*] **Example:** `Router(config)# interface POS1/0/0`	Specifies the interface and enters interface configuration mode.
Step 4	**mpls traffic-eng backup-path tunnel** *interface* **Example:** `Router(config-if)# mpls traffic-eng backup-path tunnel1`	Enables a backup tunnel on a specified interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuration Examples for the MPLS Traffic Engineering--Fast Reroute MIB

Example Enabling an SNMP Agent on a Host NMS

```
enable
 show running-config
 configure terminal
 snmp-server community public ro
 snmp-server enable traps mpls fast-reroute protected
 end
 write memory
```

Example Enabling Cisco Express Forwarding

```
enable
 configure terminal
 ip cef
 end
```

Example Enabling TE Tunnels

```
enable
 configure terminal
 ip cef
 mpls traffic-eng tunnels
 interface FastEthernet1/0/0
 mpls traffic-eng tunnels
 end
```

Example Enabling MPLS FRR on Each TE Tunnel

```
enable
 configure terminal
 interface POS1/0/0
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng fast-reroute
 end
```

Example Enabling a Backup Tunnel on an Interface

```
enable
 configure terminal
 interface POS1/0/0
 mpls traffic-eng backup-path tunnel1
 end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS applications	*Multiprotocol Label Switching Command Reference*
SNMP agent support for the MPLS Traffic Engineering MIB	MPLS Traffic Engineering MIB
Fast Reroute	MPLS Traffic Engineering: Fast Reroute Link and Node Protection

Standards

Standard	Title
MPLS-FRR-MIB	*draft-ietf-mpls-fastreroute-mib-02.txt*

MIBs

MIB	MIBs Link
MPLS Traffic Engineering (TE) MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering--Fast Reroute MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 75: Feature Information for MPLS Traffic Engineering--Fast Reroute MIB

Feature Name	Releases	Feature Information
MPLS Embedded Management--MPLS Fast Reroute MIB (IETF draft v01)	Cisco IOS XE Release 2.3	The Fast Reroute MIB provides SNMP-based network management of the Multiprotocol Label Switching (MPLS) Fast Reroute (FRR) feature.

Glossary

FEC—Forwarding Equivalence Class. A set of packets that can be handled equivalently for forwarding purposes and are thus suitable for binding to a single label. Examples include the set of packets destined for one address prefix and any flow.

flow—Generally, a set of packets traveling between a pair of hosts, or a pair of transport protocol ports on a pair of hosts. For example, packets with the same source address, source port, destination address, and destination port might be considered a flow.

A flow is also a stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

fragmentation—Process of breaking a packet into smaller units when transmitting over a network medium that cannot support the original size of the packet.

ICMP— Internet Control Message Protocol. A network layer Internet protocol that reports errors and provides other information relevant to IP packet processing. It is documented in RFC 792.

LFIB—label forwarding information base. A data structure and way of managing forwarding in which destinations and incoming labels are associated with outgoing interfaces and labels.

localhost—A name that represents the host name of a device. The localhost uses the reserved loopback IP address 127.0.0.1.

LSP—label switched path. A connection between two devices that uses MPLS to carry the packets.

LSPV—Label Switched Path Verification. An LSP Ping subprocess that encodes and decodes MPLS echo requests and replies; interfaces with IP, MPLS, and AToM switching for sending and receiving MPLS echo requests and replies; and, at the MPLS echo request originator device, maintains a database of outstanding echo requests for which echo responses have not been received.

MPLS router alert label—An MPLS label of 1. An MPLS packet with a router alert label is redirected by the device to the Route Processor (PR) processing level for handling. This allows these packets to bypass any forwarding failures in hardware routing tables.

MRU—maximum receive unit. Maximum size, in bytes, of a labeled packet that can be forwarded through an LSP.

MTU—maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

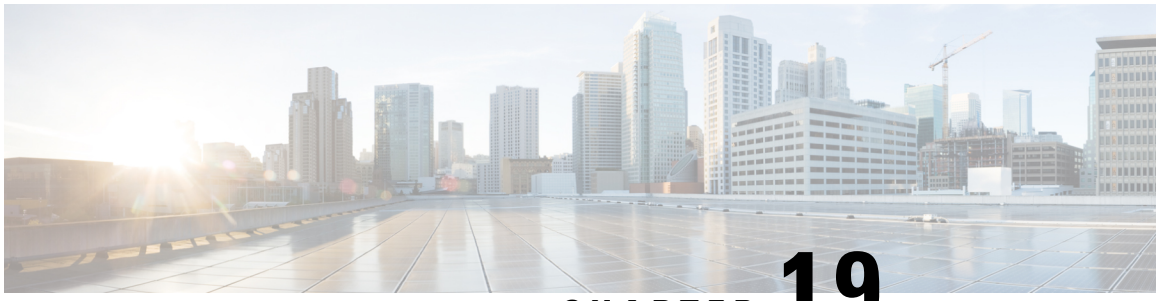
punt—Redirect packets with a router alert from the line card or interface to Route Processor (RP) level processing for handling.

PW—pseudowire. A mechanism that carries the essential elements of an emulated circuit from one provider edge (PE) device to another PE device over a packet-switched network.

RP—Route Processor. Processor module in the Cisco 7000 series routers that contains the CPU, system software, and most of the memory components that are used in the device. It is sometimes called a supervisory processor.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive. RSVP depends on IPv6. Is is also known as Resource Reservation Setup Protocol.

UDP—User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.



CHAPTER 19

MPLS Traffic Engineering MIB

The MPLS Traffic Engineering MIB enables Simple Network Management Protocol (SNMP) agent support in Cisco software for Multiprotocol Label Switching (MPLS) traffic engineering (TE) management, as implemented in the MPLS Traffic Engineering MIB (MPLS TE MIB). The SNMP agent code operating with the MPLS TE MIB enables a standardized, SNMP-based approach to be used in managing the MPLS TE features in Cisco software.

Restrictions for the MPLS Traffic Engineering MIB

- Supports read-only (RO) permission for MIB objects.
- Contains no configuration support by means of SET functions, except for the mplsTunnelTrapEnable object (which has been made writable). Accordingly, the MPLS TE MIB contains indexing support for the Interfaces MIB.
- Supports only SNMP GET, GETNEXT, and GETBULK retrieval functions, except in the case of the mplsTunnelTrapEnable object (which has been made writable by means of SET functions).
- Contains no support for Guaranteed Bandwidth Traffic Engineering (GBTE) or Auto Bandwidth features.

Information About the MPLS Traffic Engineering MIB

MPLS Traffic Engineering MIB Cisco Implementation

The MPLS TE MIB is based on the Internet Engineering Task Force (IETF) draft MIB entitled *draft-ietf-mpls-te-mib-05.txt* which includes objects describing features that support MPLS TE.

Slight differences between the IETF draft MIB and the implementation of the TE capabilities within Cisco software require some minor translations between the MPLS TE MIB and the internal data structures of Cisco software. These translations are made by the SNMP agent code that is installed and operating on various hosts within the network. This SNMP agent code, running in the background as a low priority process, provides a management interface to Cisco software.

The SNMP objects defined in the MPLS TE MIB can be displayed by any standard SNMP utility. All MPLS TE MIB objects are based on the IETF draft MIB; thus, no specific Cisco SNMP application is required to support the functions and operations pertaining to the MPLS TE MIB.

MPLS Traffic Engineering Overview

MPLS TE capabilities in Cisco software enable an MPLS backbone to replicate and expand upon the TE capabilities of Layer 2 ATM and Frame Relay networks.

TE capabilities are essential to effective management of service provider and Internet service provider (ISP) backbones. Such backbones must support high transmission capacities, and the networks incorporating backbones must be extremely resilient to link or node failures.

The MPLS TE facilities built into Cisco software provide a feature-rich, integrated approach to managing the large volumes of traffic that typically flow through WANs. The MPLS TE facilities are integrated into Layer 3 network services, thereby optimizing the routing of IP traffic in the face of constraints imposed by existing backbone transmission capacities and network topologies.

Capabilities Supported by the MPLS Traffic Engineering MIB

- The ability to generate and queue notification messages that signal changes in the operational status of MPLS TE tunnels.
- Extensions to existing SNMP commands that provide the ability to enable, disable, and configure notification messages for MPLS TE tunnels.
- The ability to specify the name or the IP address of a network management station (NMS) in the operating environment to which notification messages are to be sent.
- The ability to write notification configurations into nonvolatile memory.

Notification Generation Events

When MPLS TE notifications are enabled (see the **snmp-server enable traps mpls** command), notification messages relating to specific events within Cisco software are generated and sent to a specified NMS in the network.

For example, an mplsTunnelUp notification is sent to an NMS when an MPLS TE tunnel is configured and the tunnel transitions from an operationally “down” state to an “up” state.

Conversely, an mplsTunnelDown notification is generated and sent to an NMS when an MPLS TE tunnel transitions from an operationally “up” state to a “down” state.

An mplstunnelRerouted notification is sent to the NMS under the following conditions:

- The signaling path of an existing MPLS TE tunnel fails for some reason and a new path option is signaled and placed into effect (that is, the tunnel is rerouted).

- The signaling path of an existing MPLS TE tunnel is fully operational, but a better path option can be signaled and placed into effect (that is, the tunnel can be reoptimized). This reoptimazation can be triggered by:
  - A timer
  - The issuance of an **mpls traffic-eng reoptimize** command
  - A configuration change that requires the resignaling of a tunnel

The mplsTunnelReoptimized notification is not generated when an MPLS traffic engineering tunnel is reoptimized. However, an mplsTunnelReroute notification is generated. Thus, at the NMS, you cannot distinguish between a tunnel reoptimization event and tunnel reroute event.

Path options are configurable parameters that you can use to specify the order of priority for establishing a new tunnel path. For example, you can create a tunnel head configuration and define any one of many path options numbered 1 through n, with "1" being the highest priority option and "n" being an unlimited number of lower priority path options. Thus, there is no limit to the number of path options that you can specify in this manner.

Notification Implementation

When an MPLS TE tunnel interface (or any other device interface, such as an FastEthernet or Packet over SONET (POS) interface) transitions between an up and down state, an Interfaces MIB (ifMIB) link notification is generated. When such a notification occurs in an MPLS TE MIB environment, the interface is checked by software to determine if the notification is associated with an MPLS TE tunnel. If so, the interfaces MIB link notification is interlinked with the appropriate mplsTunnelUp or mplsTunnelDown notification to provide notification to the NMS regarding the operational event occurring on the tunnel interface. Hence, the generation of an Interfaces MIB link notification pertaining to an MPLS traffic engineering tunnel interface begets an appropriate mplsTunnelUp or mplsTunnelDown notification that is transmitted to the specified NMS.

An mplsTunnelRerouted notification is generated whenever the signaling path for an MPLS TE tunnel changes. However, software intelligence in the MPLS TE MIB prevents the reroute notification from being sent to the NMS when a TE tunnel transitions between an up or down state during an administrative or operational status check of the tunnel. Either an up or down notification or a reroute notification can be sent in this instance, but not both. This action prevents unnecessary traffic on the network.

Benefits of the MPLS Traffic Engineering MIB

- Provides a standards-based SNMP interface for retrieving information about MPLS TE.
- Provides information about the traffic flows on MPLS TE tunnels.
- Presents MPLS TE tunnel routes, including the configured route, the Interior Gateway Protocol (IGP) calculated route, and the actual route traversed.
- Provides information, in conjunction with the Interfaces MIB, about how a tunnel was rerouted in the event of a link failure.
- Provides information about the configured resources used for an MPLS TE tunnel.
- Supports the generation and queueing of notifications that call attention to major changes in the operational status of MPLS TE tunnels;
- Forwards notification messages to a designated NMS for evaluation or action by network administrators.

MPLS Traffic Engineering MIB Layer Structure

The SNMP agent code supporting the MPLS TE MIB follows the existing model for such code in Cisco software and is, in part, generated by the Cisco tool set, based on the MIB source code.

The SNMP agent code, which has a layered structure similar to that of the MIB support code in Cisco software, consists of four layers:

- Platform independent layer--This layer is generated primarily by the Cisco MIB development tool set and incorporates platform and implementation independent functions. The Cisco MIB development tool set creates a standard set of files associated with a MIB.
- Application interface layer--The functions, names, and template code for MIB objects in this layer are also generated by the Cisco MIB development tool set.
- Application specific layer--This layer provides an interface between the application interface layer and the application program interface (API) and data structures layer and performs tasks needed to retrieve required information from Cisco software, such as searching through data structures.
- API and data structures layer--This layer contains the data structures or APIs within Cisco software that are retrieved or called in order to set or retrieve SNMP management information.

Features and Technologies Related to the MPLS Traffic Engineering MIB

The MPLS TE MIB feature is used with the following features and technologies:

- Standards-based SNMP network management application
- MPLS
- MPLS TE

Supported Objects in the MPLS Traffic Engineering MIB

The MPLS TE MIB contains numerous tables and object definitions that provide read-only SNMP management support for the MPLS TE features in Cisco software. The MPLS TE MIB conforms to Abstract Syntax Notation One (ASN.1), thus reflecting an idealized MPLS TE database.

Using any standard SNMP network management application, you can retrieve and display information from the MPLS TE MIB by using GET operations; similarly, you can traverse information in the MIB database for display by using GETNEXT operations.

The MPLS TE MIB tables and objects supported in Cisco software follow. Important MIB tables (those highlighted in bold type) are described briefly in accompanying text.

- mplsTunnelConfigured--Total number of tunnel configurations that are defined on this node.
- mplsTunnelActive--Total number of label switched paths (LSPs) that are defined on this node.
- mplsTunnelTEDistProto--The IGP distribution protocol in use.
- mplsTunnelMaxHops--The maximum number of hops any given tunnel can utilize.
- mplsTunnelIndexNext--Unsupported; set to 0.

- mplsTunnelTable--Entries in this table with an instance of 0 and a source address of 0 represent tunnel head configurations. All other entries in this table represent instances of LSPs, both signaled and standby. If a tunnel instance is signaled, its operating status (operStatus) is set to “up” (1) and its instance corresponds to an active LSP.

Tunnel configurations exist only on the tunnel head where the tunnel interface is defined. LSPs traverse the network and involve tunnel heads, tunnel midpoints, and tunnel tails.

Pointers in the tunnel table refer to corresponding entries in other MIB tables. By using these pointers, you can find an entry in the mplsTunnelTable and follow a pointer to other tables for additional information. The pointers are the following: mplsTunnelResourcePointer, mplsTunnelHopTableIndex, mplsTunnelARHopTableIndex, and mplsTunnelCHopTableIndex.

The tunnel table is indexed by tunnel ID, tunnel instance, tunnel source address, and tunnel destination address. The description of each entry has an alphabetic suffix (a) for tunnel head configurations only, (b) for LSPs only, or (c) for both tunnel head configurations and LSPs , if appropriate, to indicate the applicability of the entry.

Following is a list and description of each entry.

-
  - mplsTunnelIndex--Same as tunnel ID (c).
  - mplsTunnelInstance--Tunnel instance of the LSP; 0 for head configurations (b).
  - mplsTunnelIngressLSRId--Source IP address of the LSP; 0 for head configurations (b).
  - mplsTunnelEgressLSRId--Destination IP address of the tunnel (c).
  - mplsTunnelName--Command name for the tunnel interfaces (a).
  - mplsTunnelDescr--Descriptive name for tunnel configurations and LSPs (c).
  - mplsTunnelIsIf--Indicator of whether the entry represents an interface (c).
  - mplsTunnelIfIndex--Index of the tunnel interface within the ifMIB (a).
  - mplsTunnelXCPointer--(For midpoints only - no tails) Pointer for the LSP within the mplsXCTable of the MPLS LSR MIB (b).
  - mplsTunnelSignallingProto--Signaling protocol used by tunnels (c).
  - mplsTunnelSetupPrio--Setup priority of the tunnel (c).
  - mplsTunnelHoldingPrio--Holding priority of the tunnel (c).
  - mplsTunnelSessionAttributes--Session attributes (c).
  - mplsTunnelOwner--Tunnel owner (c).
  - mplsTunnelLocalProtectInUse-- Not supported on midpoint node (c).
  - mplsTunnelResourcePointer--Pointer into the Resource Table (b).
  - mplsTunnelInstancePriority--Not implemented (b).
  - mplsTunnelHopTableIndex--Index into the Hop Table (a).
  - mplsTunnelARHopTableIndex--Index into the AR Hop Table (b).
  - mplsTunnelCHopTableIndex--Index into the C Hop Table (b).
  - mplsTunnelPrimaryTimeUp--Amount of time, in seconds, that the current path has been up (a).
  - mplsTunnelPathChanges--Number of times a tunnel has been resignalled (a).
  - mplsTunnelLastPathChange--Amount of time, in seconds, since the last path resignaling occurred (a).
  - mplsTunnelCreationTime--Time stamp when the tunnel was created (a).
  - mplsTunnelStateTransitions--Number of times the tunnel has changed state (a).
  - mplsTunnelIncludeAnyAffinity--Not implemented (a).
  - mplsTunnelIncludeAllAffinity--Attribute bits that must be set for the tunnel to traverse a link (a).

- mplsTunnelExcludeAllAffinity--Attribute bits that must *not* be set for the tunnel to traverse a link (a).
- mplsTunnelPathInUse--Path option number being used for the tunnel's path. If no path option is active, this object will be 0 (a).
- mplsTunnelRole--Role of the tunnel on the router; that is, head, midpoint, or tail (c).
- mplsTunneltotalUptime--Amount of time, in seconds, that the tunnel has been operationally up (a).
- mplsTunnelInstanceUptime--Not implemented (b).
- mplsTunnelAdminStatus--Administrative status of a tunnel (c).
- mplsTunnelOperStatus--Actual operating status of a tunnel (c).
- mplsTunnelRowStatus--This object is used in conjunction with configuring a new tunnel. This object will always be seen as "active" (a).
- mplsTunnelStorageType--Storage type of a tunnel entry (c).

- mplsTunnelHopListIndexNext--Next valid index to use as an index in the mplsTunnelHopTable.
- **mplsTunnelHopTable** --Entries in this table exist only for tunnel configurations and correspond to the path options defined for the tunnel. Two types of path options exist: *explicit* and *dynamic* . This table shows all hops listed in the explicit path options, while showing only the destination hop for dynamic path options. The tunnel hop table is indexed by tunnel ID, path option, and hop number.

Following is a list and description of each table entry.

- - mplsTunnelHopListIndex--Primary index into the table.
  - mplsTunnelHopIndex--Secondary index into the table.
  - mplsTunnelHopAddrType--Indicates if the address of this hop is the type IPv4 or IPv6.
  - mplsTunnelHopIpv4Addr--The IPv4 address of this hop.
  - mplsTunnelHopIpv4PrefixLen--The prefix length of the IPv4 address.
  - mplsTunnelHopIpv6Addr--The IPv6 address of this hop.
  - mplsTunnelHopIpv6PrefixLen--The prefix length of the IPv6 address.
  - mplsTunnelHopAsNumber--This object will contain 0 or the autonomous system number of the hop, depending on the value of mplsTunnelHopAddrType.
  - mplsTunnelHopLspId--This object will contain 0 or the LSPID of the tunnel, depending on the value of mplsTunnelHopAddrType.
  - mplsTunnelHopType--Denotes whether this tunnel hop is routed in a strict or loose fashion.
  - mplsTunnelHopRowStatus--This object is used in conjunction with the configuring of a new row in the table.
  - mplsTunnelHopStorageType--The storage type of this MIB object.

- mplsTunnelResourceIndexNext--This object contains the next appropriate value to be used for mplsTunnelResourceIndex when creating entries in the mplsTunnelResourceTable
- **mplsTunnelResourceTable** --Entries in this table correspond to the "Tspec" information displayed when you execute the **show mpls traff9c-eng tunnels** command. These entries exist only for LSPs.

The tunnel resource table is indexed by address and hop number. Following the mplsTunnelResourcePointer pointer from the tunnel table is the best way to retrieve information from this table.

Following is a list and description of each table entry.

- - mplsTunnelResourceIndex--The primary index into this table.
  - mplsTunnelResourceMaxRate--The maximum rate, in bits per second, supported by this tunnel.
  - mplsTunnelResourceMeanRate--The mean rate, in bits per second, supported by this tunnel.

- mplsTunnelResourceMaxBurstSize--The maximum burst size, in bytes, allowed by this tunnel.
- mplsTunnelResourceRowStatus--This object is used in conjunction with the configuration of a new row in the table.
- mplsTunnelResourceStorageType--The storage type of this MIB object.

- **mplsTunnelARHopTable** --Entries in this table correspond to the actual route taken by the tunnel, and whose route was successfully signaled by the network. The hops present in this table correspond to those present in the record route object (RRO) in Resource Reservation Protocol (RSVP). You can also display the information in this table by executing the **show mpls traff9c-eng tunnels** command.

The actual route hop table is indexed by address and hop number. Following the mplsTunnelARHopTableIndex pointer from the tunnel table is the best way to retrieve information from this table.

Following is a list and description of each table entry:

- - mplsTunnelARHopListIndex--The primary index into this table.
  - mplsTunnelARHopIndex--The secondary index into this table.
  - mplsTunnelARHopIpv4Addr--The IPv4 address of this hop.
  - mplsTunnelARHopIpv4PrefixLen--The prefix length of the IPv4 address.
  - mplsTunnelARHopIpv6Addr--The IPv6 address of this hop.
  - mplsTunnelARHopIpv6PrefixLen--The prefix length of the IPv6 address.
  - mplsTunnelARHopAsNumber--This object will contain 0 or the AS number of the hop, depending on the value of mplsTunnelARHopAddrType.
  - mplsTunnelARHopAddrType--The type of address for this MIB entry, either IPv4 or IPv6.
  - mplsTunnelARHopType--Denotes whether this tunnel hop is routed in a strict or loose manner.

- **mplsTunnelCHopTable** --Entries in this table correspond to the explicit route object (ERO) in RSVP, which is used to signal the LSP. The list of hops in this table will contain those hops that are computed by the constraint-based shortest path first (SPF) algorithm. In those cases where "loose" hops are specified for the tunnel, this table will contain the hops that are "filled-in" between the loose hops to complete the path. If you specify a complete explicit path, the computed hop table matches your specified path.

The computed hop table is indexed by address and hop number. Following the mplsTunnelCHopTableIndex pointer from the tunnel table is the best way to retrieve information from this table.

Following is a list and description of each table entry:

- - mplsTunnelCHopListIndex--The primary index into this table.
  - mplsTunnelCHopIndex--The secondary index into this table.
  - mplsTunnelCHopAddrType--Indicates if the address of this hop is the type IPv4 or IPv6.
  - mplsTunnelCHopIpv4Addr--The IPv4 address of this hop.
  - mplsTunnelCHopIpv4PrefixLen--The prefix length of the IPv4 address.
  - mplsTunnelCHopIpv6Addr--The IPv6 address of this hop.
  - mplsTunnelCHopIpv6PrefixLen--The prefix length of the IPv6 address.
  - mplsTunnelCHopAsNumber--This object will contain 0 or the autonomous system number of the hop, depending on the value of mplsTunnelHopAddrType.
  - mplsTunnelCHopType--Denotes whether this tunnel hop is routed in a strict or loose way.

- **mplsTunnelPerfTable** --The tunnel performance table, which augments the **mplsTunnelTable**, provides packet and byte counters for each tunnel. This table contains the following packet and byte counters:

  - mplsTunnelPerfPackets--This packet counter works only for tunnel heads.
  - mplsTunnelPerfHCPackets--This packet counter works only for tunnel heads.

- mplsTunnelPerfErrors--This packet counter works only for tunnel heads.
- mplsTunnelPerfBytes--This byte counter works for tunnel heads and tunnel midpoints, but not for tunnel tails.
- mplsTunnelPerfHCBytes--This byte counter works for tunnel heads and tunnel midpoints, but not for tunnel tails.

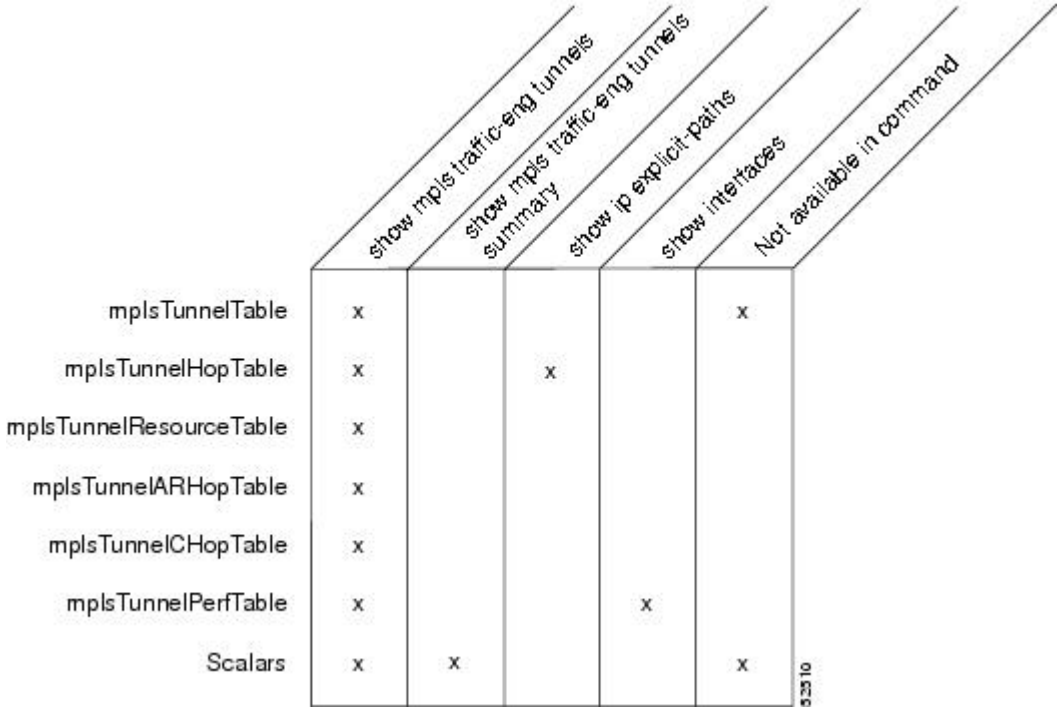
- mplsTunnelTrapEnable--The object type *mplsTunnelTrapEnable* is enhanced to be writable. Accordingly, if this object type is set to "TRUE," the following notifications are enabled, thus giving you the ability to monitor changes in the operational status of MPLS TE tunnels:
  - mplsTunnelUp
  - mplsTunnelDown
  - mplsTunnelRerouted

If the *mplsTunnelTrapEnable* object is set to "FALSE," such operational status notifications are not generated. These notification functions are based on the definitions (mplsTeNotifications) contained in the IETF draft document entitled *draft-ietf-mpls-te-mib-05.txt* .

CLI Access to MPLS Traffic Engineering MIB Information

The figure below shows commands that you can use to retrieve information from specific tables in the MPLS TE MIB. As noted in this figure, some information in the MPLS TE MIB is not retrievable by commands.

Figure 39: Commands for Retrieving MPLS TE MIB Information



Retrieving Information from the MPLS Traffic Engineering MIB

This section describes how to efficiently retrieve information about TE tunnels. Such information can be useful in large networks that contain many TE tunnels.

Traverse across a single column of the mplsTunnelTable, such as mplsTunnelName. This action provides the indexes of every tunnel configuration, and any LSPs involving the host router. Using these indexes, you can perform a GET operation to retrieve information from any column and row of the mplsTunnelTable.

The mplsTunnelTable provides pointers to other tables for each tunnel. The column mplsTunnelResourcePointer, for example, provides an object ID (OID) that you can use to access resource allocation information in the mplsTunnelResourceTable. The columns mplsTunnelHopTableIndex, mplsTunnelARHopTableIndex, and mplsTunnelCHopTableIndex provide the primary index into the mplsTunnelHopTable, mplsTunnelARHopTable, and mplsTunnelCHopTable, respectively. By traversing the MPLS TE MIB in this manner using a hop table column and primary index, you can retrieve information pertaining to the hops of that tunnel configuration.

Because tunnels are treated as interfaces, the tunnel table column (mplsTunnelIfIndex) provides an index into the Interfaces MIB that you can use to retrieve interface-specific information about a tunnel.

How to Configure the MPLS Traffic Engineering MIB

Enabling the SNMP Agent to Help Manage Various MPLS TE Tunnel Characteristics of Tunnels on the Local Router

The SNMP agent for the MPLS TE MIB is disabled by default. To enable the SNMP agent for the MPLS TE MIB, perform the following task.

SUMMARY STEPS

1. **telnet** *host*
2. **enable**
3. **show running-config**
4. **configure terminal**
5. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
6. **snmp-server enable traps** [*identification-type*] [*notification-option*]
7. **exit**
8. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**telnet** *host* **Example:** `Router> telnet 192.172.172.172`	Telnets to the router identified by the specified IP address (represented as *xxx.xxx.xxx.xxx*).
Step 2	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 3	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. • If no SNMP information is displayed, go to Step 4 . If any SNMP information is displayed, you can modify the information or change it as needed.
Step 4	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 5	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Router(config)# snmp-server community comaccess ro 4`	Enables the read-only (RO) community string.
Step 6	**snmp-server enable traps** [*identification-type*] [*notification-option*] **Example:** `Router(config)# snmp-server enable traps`	Enables an LSR to send SNMP notifications or informs to an SNMP host. **Note** This command is optional. After SNMP is enabled, all MIBs (not just the TE MIB) are available for the user to query.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**write memory** **Example:** `Router# write memory`	Writes the modified configuration to NVRAM, permanently saving the settings.

Verifying the Status of the SNMP Agent

To verify that the SNMP agent has been enabled on a host network device, perform the following steps.

SUMMARY STEPS

1. **telnet** *host*
2. **enable**
3. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**telnet** *host* **Example:** `Router# telnet 192.172.172.172`	Telnets to the target device identified by the specified IP address (represented as *xxx.xxx.xxx.xxx*).
Step 2	**enable** **Example:** `Router# enable`	Enables SNMP on the target device.
Step 3	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration on the target device and is used to examine the output for displayed SNMP information.

Examples

The follows example displays the running configuration on the target device and its SNMP information.

```
Router# show running-config
.
.
.
snmp-server community public ro
snmp-server community private ro
```

Any **snmp-server** statement that appears in the output and takes the form shown here verifies that SNMP has been enabled on that device.

Configuration Examples for the MPLS Traffic Engineering MIB

Example Enabling the SNMP Agent to Help Manage MPLS TE Characteristics of Tunnels on the Local Router

The following example shows how to enable an SNMP agent on a host network device:

```
Router# configure terminal
Router(config)# snmp-server community private
```

The following example shows how to enable SNMPv1 and SNMPv2C. The configuration permits any SNMP agent to access all MPLS TE MIB objects with read-only permissions using the community string public.

```
Router(config)# snmp-server community public
```

The following example shows how to allow read-only access to all MPLS TE MIB objects relating to members of access list 4 that specify the comaccess community string. No other SNMP agents will have access to any MPLS TE MIB objects.

```
Router(config)# snmp-server community comaccess ro 4
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Information about MPLS TE and enhancements	MPLS Traffic Engineering and Enhancements
MPLS TE commands	*Multiprotocol Label Switching Command Reference*
SNMP commands	*Network Management Command Reference*
SNMP configuration	"Configuring SNMP Support" in the *Network Management Configuration Guide*

Standards

Standard	Title
draft-ietf-mpls-te-mib-05	*MPLS Traffic Engineering Management Information Base Using SMIv2*

MIBs

MIB	MIBs Link
• MPLS TE MIB • Interfaces MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2026	*The Internet Standards Process*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the MPLS Traffic Engineering MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 76: Feature Information for the MPLS Traffic Engineering MIB

Feature Name	Releases	Feature Information
MPLS Traffic Engineering MIB	Cisco IOS XE Release 2.3	The MPLS Traffic Engineering MIB feature enables the SNMP agent support in Cisco software for MPLS TE management, as implemented in the MPLS TE MIB. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **snmp-server community**, **snmp-server enable traps** , **snmp-server hpst**.

Glossary

affinity bits—An MPLS traffic engineering tunnel's requirements on the attributes of the links it will cross. The tunnel's affinity bits and affinity mask must match with the attributes of the various links carrying the tunnel.

call admission precedence—An MPLS traffic engineering tunnel with a higher priority will, if necessary, preempt an MPLS traffic engineering tunnel with a lower priority. An expected use is that tunnels that are more difficult to route will have a higher priority, and can preempt tunnels that are less difficult to route, on the assumption that those lower priority tunnels can find another path.

constraint-based routing—Procedures and protocols used to determine a route across a backbone taking into account resource requirements and resource availability, instead of simply using the shortest path.

flow —A traffic load entering the backbone at one point—point of presence (POP)—and leaving it from another that must be traffic engineered across the backbone. The traffic load will be carried across one or more LSP tunnels running from the entry POP to the exit POP.

headend —The LSR at which the tunnel originates. The tunnel's "head" or tunnel interface will reside at this LSR as well.

informs —A type of notification message that is more reliable than a conventional trap notification message because an informs message requires acknowledgment.

label —A short, fixed-length data construct that tells switching nodes how to forward data (packets or cells).

label switched path (LSP) tunnel—A configured connection between two routers, using label switching to carry the packets.

LSP —label switched path. A path that is followed by a labeled packet over several hops, starting at an ingress LSR and ending at an egress LSR.

LSR —label switch router. A Layer 3 router that forwards a packet based on the value of a label encapsulated in the packet.

MIB —Management Information Base. A database of network management information (consisting of MIB objects) that is used and maintained by a network management protocol such as SNMP. The value of a MIB object can be changed or retrieved using SNMP commands, usually by a GUI-based network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS —Multiprotocol Label Switching. Switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

NMS —network management station. An NMS is a powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

notification —A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS software has occurred (see traps).

OSPF —Open Shortest Path First. A link-state routing protocol used for routing IP.

RSVP —Resource Reservation Protocol. Protocol for reserving network resources to provide quality of service (QoS) guarantees to application flows.

SNMP —Simple Network Management Protocol. A network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, manage configurations, collect statistics, monitor performance, and ensure network security.

tailend —The downstream, receive end of a tunnel.

traffic engineering—Techniques and processes that cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

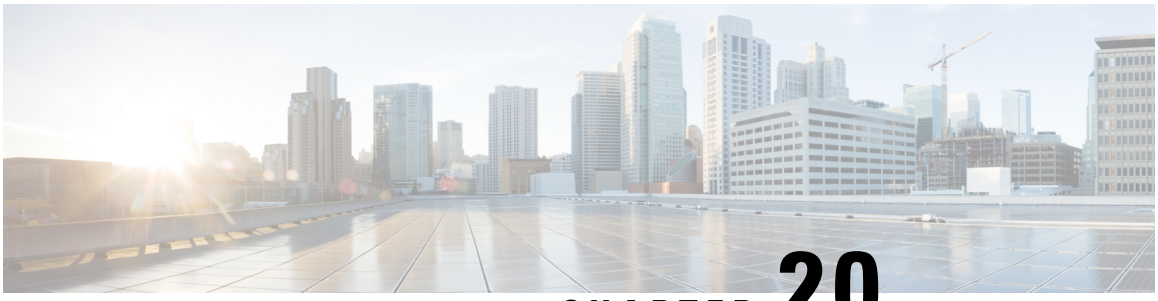
trap —A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS software has occurred. Traps (notifications) are less reliable than inform requests, because the receiver of the trap does not send an acknowledgment of receipt; furthermore, the sender of the trap cannot determine if the trap was received (see notification).

VCC —virtual channel connection. A VCC is a logical circuit consisting of VCLs that carries data between two endpoints in an ATM network. Sometimes called a virtual circuit connection.

VCI —virtual channel identifier. A 16-bit field in the header of an ATM cell. The VCI, together with the VPI, is used to identify the next network VCL as the cell passes through a series of ATM switches on its way to its final destination.

VCL —virtual channel link. A VCL is the logical connection that exists between two adjacent switches in an ATM network.

VPI —virtual path identifier. An 8-bit field in the header of an ATM cell. The VPI, together with the VCI, is used to identify the next network VCL as the cell passes through a series of ATM switches on its way to its final destination.



CHAPTER 20

Point-to-Multipoint MPLS-TE MIB

The point-to-multipoint (P2MP) Multiprotocol Label Switching (MPLS)-traffic engineering (TE) MIB describes the Label Switched Path (LSP) in an IP MPLS network. An administrator can use the P2MP MPLS-TE MIB to determine optimal branch points in the network so that optimal links are used.

The P2MP MPLS-TE MIB has the following benefits:

- Provides a standards-based Simple Network Management Protocol (SNMP) interface for retrieving information about P2MP MPLS-TE tunnels.
- Shows traffic flow information for P2MP MPLS-TE tunnels.
- Shows P2MP MPLS-TE tunnel routes, including the configured route, the Interior Gateway Protocol (IGP) calculated route, and the actual route traversed.
- Shows tunnel reroute information in the event of a link failure in accordance with the Interfaces MIB.
- Shows information about the configured resources used for an P2MP MPLS-TE tunnel.
- Supports the generation of notifications. These notifications call attention to major changes in the operational status of P2MP MPLS-TE tunnels.
- Forwards notification messages to a designated network management server (NMS) for evaluation or action by administrators.

Restrictions for Point-to-Multipoint MPLS-TE MIB

The P2MP MPLS-TE MIB supports read-only access to the objects.

Information About the Point-to-Multipoint MPLS-TE MIB

Point-to-Multipoint MPLS-TE MIB Cisco Implementation

The P2MP MPLS-TE MIB is based on the Internet Engineering Task Force (IETF) draft MIB entitled *draft-ietf-mpls-p2mp-te-mib-09* , which includes objects describing features that support P2MP MPLS-TE.

The SNMP objects defined in the P2MP MPLS-TE MIB can be displayed by any standard SNMP utility. All supported P2MP MPLS-TE MIB objects are based on the IETF draft MIB; thus, no specific Cisco SNMP application is required to support the functions and operations pertaining to the P2MP MPLS-TE MIB.

The P2MP MPLS-TE LSP determines optimal branch points in the IP and MPLS network and must be managed using SNMP.

MPLS was defined in RFC 3031, and a signaling protocol for P2MP MPLS-TE, that is, TE extensions to the Resource Reservation Protocol (RSVP-TE) was defined in RFC 3209. RSVP-TE is extended for use in a P2MP MPLS-TE environment by RFC 4875 following the requirements set out in RFC 4461.

RFC 3812 provides a MIB module for modeling and controlling P2MP MPLS-TE in conjunction with Textual Conventions defined in RFC 3811. In addition, RFC 3813 defines a MIB module for modeling and controlling an MPLS Label Switching Router (LSR) that may support MPLS-TE. An overview of MPLS MIB modules can be found in RFC 4221.

In addition, there is a description for how to use the LSR MIB module RFC 3813 to model and control an LSR that supports P2MP MPLS-TE.

Functionality Supported by the Point-to-Multipoint MPLS-TE MIB

- The ability to generate and queue notification messages that signal changes in the operational status of P2MP MPLS-TE tunnels.
- The ability to enable, disable, and configure notification messages for P2MP MPLS-TE tunnels through extensions to existing SNMP commands. .
- The ability to specify the name or the IP address of a network management station (NMS) in the operating environment to which notification messages are to be sent.
- The ability to write notification configurations into nonvolatile memory.

Notification Generation Events for the Point-to-Multipoint MPLS-TE MIB

When P2MP MPLS-TE notifications are enabled (see the **snmp-server enable traps mpls p2mp-traffic-eng** command), notification messages relating to specific events within Cisco software are generated and sent to a specified NMS in the network. For example, an mplsTeP2mpTunnelDestUp notification is sent to an NMS when an P2MP MPLS-TE tunnel is configured and the tunnel transitions from an operationally “down” state to an “up” state. Conversely, an mplsTeP2mpTunnelDestDown notification is generated and sent to an NMS when an MPLS-TE tunnel transitions from an operationally “up” state to a “down” state.

An up or down trap is generated per sub-LSP when a sub-LSP up or down event is triggered. The traps have two Varbinds:

- cmplsTeP2mpTunnelDestAdminStatus
- cmplsTeP2mpTunnelDestOperStatus

Each trap object is indexed by 12 indices:

- mplsTunnelIndex
- mplsTunnelInstance
- mplsTunnelIngressLsrId
- mplsTunnelEgressLsrId
- mplsTeP2mpTunnelDestSrcSubGroupOriginType
- mplsTeP2mpTunnelDestSrcSubGroupOrigin
- mplsTeP2mpTunnelDestSrcSubGroupID
- mplsTeP2mpTunnelDestSubGroupOriginType
- mplsTeP2mpTunnelDestSubGroupOrigin
- mplsTeP2mpTunnelDestSubGroupID
- mplsTeP2mpTunnelDestDestinationType
- mplsTeP2mpTunnelDestDestination

Supported Objects in the Point-to-Multipoint MPLS-TE MIB

Using any standard SNMP network management application, you can retrieve and display information from the P2MP MPLS-TE MIB (CISCO-IETF-MPLS-TE-P2MP-MIB) by using GET operations; similarly, you can traverse information in the MIB database for display by using GETNEXT operations.

Note The **show mpls traffic-eng tunnels** command can be used to see the value of various objects discussed in this section through the CLI.

The point-to-multipoint MIB tables and objects supported in Cisco IOS releases follow. Important MIB tables (those highlighted in bold type) are described briefly in accompanying text.

Following is a list and description of each entry for the supported P2MP MPLS-TE MIB Scalar objects:

- mplsTeP2mpTunnelConfigured—Total number of P2MP MPLS-TE tunnels configured on this device. A tunnel is considered configured if the mplsTunnelRowStatus in MPLS-TE-STD-MIB is active.
- mplsTeP2mpTunnelActive—Total number of P2MP MPLS-TE tunnels active on this device.

mplsTeP2mpTunnelTable

The mplsTeP2mpTunnelTable allows new P2MP MPLS-TE tunnels to be created between an LSR and one or more remote endpoints, and existing P2MP MPLS-TE tunnels to be reconfigured or removed. This table augments the mplsTunnelTable in the MPLS-TE-STD-MIB so that the entries in that table can be flagged as

point-to-multipoint, and can be configured and monitored appropriately. The mplsTeP2mpTunnelTable is indexed by four indices similar to the mplsTunnelTable:

- mplsTunnelIndex
- mplsTunnelInstance
- mplsTunnelIngressLSRId
- mplsTunnelEgressLSRId

The mplsTunnelEgressLSRId maps to the P2MP MPLS-TE subgroup-id of the sub-lsp. The subgroup-id is zero for the P2MP MPLS-TE LSP virtual interface (VIF). Therefore, the mplsTunnelEgressLSRId has a zero value for the P2MP MPLS-TE LSP VIF. The mapping of the remainder of the indices is similar to those found in the MPLS-TE MIB.

The following is a list and description of each supported mplsTeP2mpTunnelTable object:

- mplsTeP2mpTunnelBranchRole—Supplements the value in the object mplsTunnelRole in MPLS-TE-STD-MIB that indicates the role of this LSR in the tunnel represented by this entry in mplsTeP2mpTunnelTable. This object supports the head, transit, and tail in order to describe the role of LSR for the tunnel. The following objects describe the mplsTunnelRole:
  - RRR_MGMT_TUN_ROLE_HEAD–Contains the value D_cmplsTeP2mpTunnelBranchRole_notBranch.
  - RRR_MGMT_TUN_ROLE_MID–Contains the value D_cmplsTeP2mpTunnelBranchRole_branch.
  - RRR_MGMT_TUN_ROLE_TAIL–Contains the value D_cmplsTeP2mpTunnelBranchRole_notBranch.
- mplsTeP2mpTunnelP2mpXcIndex– Contains the value of mplsXCIndex, the primary index of the mplsXCTable for all cross-connect entries for this P2MP MPLS-TE LSP.
- mplsTeP2mpTunnelRowStatus– Contains the variable used to create, modify, and delete a row in this table. The row in this table is in the active state only and no objects in that row can be modified by the agent except for the mplsTeP2mpTunnelRowStatus and mplsTeP2mpTunnelStorageType.
- mplsTeP2mpTunnelStorageType– Contains the storage type for this tunnel entry that may or may not be marked “volitile” because this value may or may not be persisted.

mplsTeP2mpTunnelDestTable

An entry in this table represents a destination of a P2MP MPLS-TE tunnel. Entries in this table share some index fields with the mplsTeP2mpTunnelTable and the mplsTunnelTable in MPLS-TE-STD-MIB. Entries in this table have no meaning unless there is a corresponding entry in mplsTeP2mpTunnelTable (which, itself, depends on a corresponding entry in mplsTunnelTable). This table shows information about sub-LSPs (one entry per destination of the tunnel). An entry to a destination is considered as sub-lsp.

The first three mplsTeP2mpTunnelDestTable indices are the same as the MPLS-TE MIB:

- mplsTunnelIndex
- mplsTunnelInstance
- mplsTunnelIngressLSRId

The fourth index entry, mplsTunnelEgressLSRId maps to the P2MP MPLS-TE ID. The remainder of the indices are as follows:

- mplsTeP2mpTunnelDestSrcSubGroupOriginType
- mplsTeP2mpTunnelDestSrcSubGroupOrigin
- mplsTeP2mpTunnelDestSrcSubGroupID
- mplsTeP2mpTunnelDestSubGroupOriginType
- mplsTeP2mpTunnelDestSubGroupOrigin
- mplsTeP2mpTunnelDestSubGroupID
- mplsTeP2mpTunnelDestDestinationType
- mplsTeP2mpTunnelDestDestination

The following is a list and description of each supported mplsTeP2mpTunnelDestTable object:

- mplsTeP2mpTunnelDestSrcSubGroupOriginType—Identifies the IPv4 address carried in mplsTeP2mpTunnelDestSrcSubGroupOrigin object at a transit or egress LSR and has a value unknown (0) for an ingress LSR.
- mplsTeP2mpTunnelDestSrcSubGroupOrigin—Contains the TE Router ID (reachable and stable IP address) of the originator of the P2MP MPLS-TE subgroup as received on a Path message by a transit or egress LSR. This object maps to the ingress LSR ID for a egress and transit LSR and unknown (0) for ingress LSR.
- mplsTeP2mpTunnelDestSrcSubGroupID—Contains the unique identifier assigned by the subgroup originator for this subgroup of this P2MP MPLS-TE tunnel as received on a Path message by a transit or egress LSR. This object maps to the subgroup ID for the sub-LSP to this destination for an egress and transit LSR ID and 0 for Ingress. The rrr_get_lsp_id_p2mp_subgroup_id (lsp_id) is used to fetch the value at egress or transit.
- mplsTeP2mpTunnelDestSubGroupOriginType—Identifies the IPv4 address carried in the mplsTeP2mpTunnelDestSubGroupOrigin object.
- mplsTeP2mpTunnelDestSubGroupOrigin—Contains the TE Router ID (reachable and stable IP address) of the originator of the P2MP MPLS-TE subgroup. In many cases, this is the ingress LSR of the P2MP MPLS-TE tunnel and is the received signaled value as available in mplsTeP2mpTunnelDestSrcSubGroupOrigin object. This object maps to Ingress LSR ID.

Note The **show mpls traffic-eng tunnels** command can be used to list the source address for tunnel that maps to this object's value.

- mplsTeP2mpTunnelDestSubGroupOrigin—Contains the unique identifier assigned by the subgroup originator for this subgroup of this P2MP MPLS-TE tunnel. This object has the value of subgroup ID of the sub-LSP. At egress, this contains the value of mplsTeP2mpTunnelDestSrcSubGroupID.
- mplsTeP2mpTunnelDestDestinationType—Identifies the IPv4 address carried in the mplsTeP2mpTunnelDestDestination object.
- mplsTeP2mpTunnelDestDestination—Identifies the single destination of this P2MP MPLS-TE tunnel that is the TE address of a leaf that can be routed. This is often the TE Router ID of the leaf, but can be any interface address. When a signaling protocol is used, this object corresponds to the S2L sub-LSP

destination address field in the S2L_SUB_LSP object. This object maps to the destination address of the sub-LSP.

- mplsTeP2mpTunnelDestBranchOutSegment—Identifies the outgoing branch from this LSR towards the destination represented by this table entry. It must be a unique identifier within the scope of this tunnel. This object contains an index into mplsOutSegmentTable object. This value maps to mplsOutSegmentIndex of the MPLS-LSR-STD-MIB. The LSR MIB shows the outgoing branches at a bud node. This outsegmentindex value (for each outsegment) is mapped to each sub-LSP.
- mplsTeP2mpTunnelDestHopTableIndex—Provides the index into the mplsTunnelHopListIndex of mplsTunnelHopTable of RFC TE MIB entry that specifies the explicit route hops for this destination of the P2MP MPLS-TE tunnel. This index is only valid for head entries. For nonhead entries the value is zero.
- mplsTeP2mpTunnelDestPathInUse—Contains the value that denotes the configured path that was chosen as the explicit path to this destination of this P2MP MPLS-TE tunnel. This value reflects the secondary index into mplsTunnelHopTable where the primary index comes from mplsTeP2mpTunnelDestHopTableIndex. This object maps to mplsTunnelPathOptionIndex of RFC TE MIB. TE only supports one path-option to be configured per sub-LSP, so the mplsTeP2mpTunnelDestPathInUse object is always 1.
- mplsTeP2mpTunnelDestCHopTableIndex—Provides the index into the mplsTunnelCHopTable that identifies the explicit path for this destination of the P2MP MPLS-TE tunnel. This object maps to mplsTunnelCHopListIndex of RFC TE MIB.
- mplsTeP2mpTunnelDestARHopTableIndex—Provides the index into the mplsTunnelARHopTable that identifies the actual hops traversed to this destination of the P2MP MPLS-TE tunnel. This is automatically updated by the agent when the actual hops become available. This object maps to mplsTunnelARHopTableIndex of RFC TE MIB.
- mplsTeP2mpTunnelDestAdminStatus—Indicates the desired operational status of this destination of this P2MP MPLS-TE tunnel. Status can be up or down.
- mplsTeP2mpTunnelDestOperStatus—Indicates the actual operational status of this destination of this P2MP MPLS-TE tunnel.
- mplsTeP2mpTunnelDestRowStatus—Creates, modifies, and/or deletes a row in this table. When a row in this table is in the active (1) state, no objects in that row can be modified by SET operations except mplsTeP2mpTunnelDestAdminStatus and mplsTeP2mpTunnelDestStorageType.
- mplsTeP2mpTunnelDestStorageType—Indicates the storage type for this table entry which is permanent. Conceptual rows having the value “permanent” do not allow write-access to any columnar objects in the row.

mplsTeP2mpTunnelBranchPerfBranch

An entry in this table provides information about P2MP MPLS-TE Tunnel. The mapping of the first four indices is similar to mplsTeP2mpTunnelTable. The table shows the outsegment information for head and transit entries. For tail entries, the entries are shown to display that tails are configured but values for objects are not provided.

This table has five indices:

- mplsTunnelIndex
- mplsTunnelInstance

- mplsTunnelIngressLSRId
- mplsTunnelEgressLSRId
- mplsTunnelBranchPerfBranch

The following is a list and description of each supported mplsTeP2mpTunnelBranchPerfBranch object:

- mplsTeP2mpTunnelBranchPerfBranch—Identifies an outgoing branch from this LSR for this tunnel. Its value is unique within the context of the tunnel. If MPLS-LSR-STD-MIB is implemented, this object should contain an index into mplsOutSegmentTable. Under all circumstances, this object should contain the same value as mplsTeP2mpTunnelDestBranchOutSegment for destinations reached on this branch. This value maps to mplsOutSegmentIndex of MPLS-LSR-STD-MIB.
- mplsTeP2mpTunnelBranchPerfPackets—Displays the number of packets forwarded by the tunnel onto this branch. This object should represents the 32-bit value of the least significant part of the 64-bit value if both mplsTeP2mpTunnelBranchPerfHCPackets is returned. This object should be read in conjunction with mplsTeP2mpTunnelBranchDiscontinuityTime object.

Note The **show interface tunnel** command can be used to see the value of packets.

- mplsTeP2mpTunnelBranchPerfHCPackets—Displays the High Capacity counter for number of packets forwarded by the tunnel onto this branch.
- mplsTeP2mpTunnelBranchPerfErrors—Displays the number of packets dropped because of errors or for other reasons, that were supposed to be forwarded onto this branch for this tunnel. This object should be read in conjunction with mplsTeP2mpTunnelBranchDiscontinuityTime object.
- mplsTeP2mpTunnelBranchPerfBytes—Displays the number of bytes forwarded by the tunnel onto this branch. This object should represent the 32-bit value of the least significant part of the 64-bit value if both mplsTeP2mpTunnelBranchPerfHCBytes is returned. This object should be read in conjunction with mplsTeP2mpTunnelBranchDiscontinuityTime object.

Note The **show mpls forwarding-table** command can be used to verify the values.

- cmplsTeP2mpTunnelBranchLocalLabel—Displays the local MPLS label for this branch.

Note Use the **show mpls traffic-eng forwarding path-set** and **show cef path set detail** commands can also be used to view label information.

- cmplsTeP2mpTunnelBranchOutIfIndex—Displays the index of outgoing TE link physical interface for the P2MP MPLS-TE egress tunnel branch. It is nonzero when the P2MP MPLS-TE tunnel's outgoing branch has been signaled.
- cmplsTeP2mpTunnelBranchOutLabel—Displays the outgoing MPLS label for this branch.

Note Use the **show mpls traffic-eng forwarding path-set** and **show cef path set detail** commands can also be used to view label information.

- cmplsTeP2mpTunnelBranchInIfIndex—Displays the index of incoming TE link physical interface for the P2MP MPLS-TE ingress tunnel branch. It is nonzero when the P2MP MPLS-TE tunnel's incoming branch has been signaled and it is applicable only on midpoint routers for tunnels having out-segment up and running.

How to Configure the Point-to-Multipoint MPLS-TE MIB

Configuring the Router to Send SNMP Notifications to a Host for Monitoring Point-to-Multipoint MPLS-TE

Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend that you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.

Perform this task to configure the router to send SNMP notifications to a host to monitor P2MP MPLS-TE. The ability to display SNMP notifications helps in managing P2MP MPLS-TE sessions by determining if any P2MP MPLS-TE sessions between peers are up or down.

The **snmp-server host** command specifies which hosts receive notifications or traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

For a host to receive a trap, an **snmp-server host** command must be configured for that host, and, generally, the trap must be enabled globally through the **snmp-server enable traps** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-address community-string* **udp-port** *port* **p2mp-traffic-eng**
4. **snmp-server enable traps mpls p2mp-traffic-eng** [**down** | **up**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**snmp-server host** *host-address community-string* **udp-port** *port* **p2mp-traffic-eng** **Example:** `Router(config)# snmp-server host 172.20.2.160 comp2mppublic udp-port 162 p2mp-traffic-eng`	Specifies the recipient of an SNMP notification operation. • The *host-address* argument specifies the name or Internet address of the host (the targeted recipient). • The *community-string* argument is a password-like community string sent with the notification operation. • The **udp-port** *port* keyword-argument pair names the UDP port of the host to use. The default is 162. • The **p2mp-traffic-eng** keyword specifies that P2MP MPLS-TE SNMP traps are allowed to be sent to the host.
Step 4	**snmp-server enable traps mpls p2mp-traffic-eng** [**down** | **up**] **Example:** `Router(config)# snmp-server enable traps mpls p2mp-traffic-eng`	Enables the router to send P2MP MPLS-TE SNMP traps. • (Optional) The **down** keyword enables or disables P2MP MPLS-TE tunnel down trap notifications (mplsTeP2mpTunnelDestDown). This message is generated when a P2MP MPLS-TE tunnel between the router and its destination is terminated. • (Optional) The **up** keyword enables or disables P2MP MPLS-TE tunnel up trap notifications (mplsTeP2mpTunnelDestUp). This notification is generated when the router establishes a P2MP MPLS-TE tunnel between the router and its destination is established.
Step 5	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS: Embedded Management and MIBs	*MPLS: Embedded Management and MIBs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
draft-ietf-mpls-p2mp-te-mib-09	*Point-to-Multipoint Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB) module*
RFC 3031	*Multiprotocol Label Switching Architecture*
RFC 3209	*RSVP-TE: Extensions to RSVP for LSP Tunnels*
RFC 3811	*Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management*
RFC 3812	*Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)*
RFC 3813	*Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)*
RFC 4221	*Multiprotocol Label Switching (MPLS) Management Overview*
RFC 4875	*Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)*
RFC4461	*Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)*

MIBs

MIB	MIBs Link
CISCO-IETF-MPLS-TE-P2MP-MIB (draft-ietf-mpls-p2mp-te-mib-09) MPLS-TE-MIB Interfaces MIB MPLS-TE-STD-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

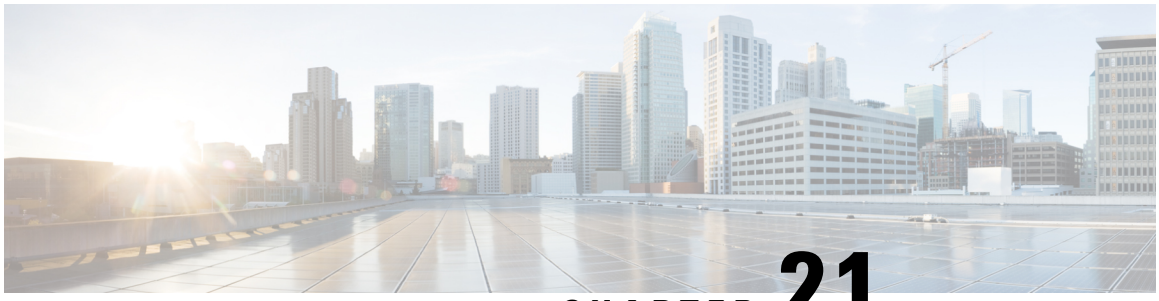
Feature Information for Point-to-Multipoint MPLS-TE MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 77: Feature Information for Point-to-Multipoint MPLS-TE MIB

Feature Name	Releases	Feature Information
Point-to-Multipoint MPLS-TE MIB	15.2(1)S	The point-to-multipoint (P2MP) Multiprotocol Label Switching (MPLS)-traffic engineering (TE) MIB describes the Label Switched Path (LSP) in the IP and MPLS network. An administrator can use the P2MP MPLS-TE MIB to determine optimal branch points in the network so that optimal links are used. This feature was introduced in Cisco IOS Release 15.2(1)S. The following commands were introduced or modified: **snmp-server enable traps mpls p2mp-traffic-eng**, **snmp-server host udp-port p2mp-traffic-eng**



CHAPTER 21

MPLS-TP MIB

The Multiprotocol Label Switching Transport Profile (MPLS-TP) allows you to meet your transport requirements as those requirements evolve from Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) time-division multiplexing (TDM) technologies to MPLS and Ethernet technologies. Currently, a strong momentum for MPLS-TP in terms of both rapid standards development and increasing market demand exists. MPLS-TP technologies have been recently requested by multiple service providers for packet transport primarily in the aggregation networks and access networks while the core network remains MPLS (MPLS-TP is being considered for core transport as well by one or two providers). Service providers aim at using MPLS-TP to support the following deployment scenarios: Ethernet services, mobile backhaul, Asynchronous Transfer Mode (ATM) aggregation replacement, video transport, and long haul transport.

MPLS TP MIB allows you to poll MPLS-TP configured nodes via Simple Network Management Protocol (SNMP) and monitor and manage the MPLS-TP network.

Prerequisites for MPLS-TP MIB

- General knowledge of SNMP
- Software used to query Cisco devices via SNMP

Restrictions for MPLS-TP MIB

- MPLS-TP MIB doesn’t specify any traps for TP and thus no trap support is provided.
- The MPLS-TP MIB module supports point-to-point, co-routed bi-directional tunnels.

Information about MPLS-TP MIB

Overview of MPLS-TP MIB

MPLS-TP MIB is part of the SNMP process. The MIB interacts with MPLS-TP functions to get the data required for objects and indices.

The following MIBs are implemented:

- CISCO-MPLS-TC-EXT-STD-MIB
- CISCO-MPLS-ID-STD-MIB
- CISCO-MPLS-LSR-EXT-STD-MIB
- CISCO-MPLS-TE-EXT-STD-MIB

CISCO-MPLS-TC-EXT-STD-MIB

This MIB module contains textual conventions for MPLS-based transport networks.

Textual Convention	Description (from IETF draft)
MplsGlobalId	This object contains the textual convention of the operator unique identifier (Global_ID). The Global_ID can contain the 2-octet or 4-octet value of the operator's Autonomous System Number (ASN). When the Global_ID is derived from a 2-octet AS number, the two high-order octets of this 4-octet identifier MUST be set to zero. ASN 0 is reserved. A Global_ID of zero means that no Global_ID is present.
MplsNodeId	The Node_ID is assigned within the scope of the Global_ID. The value 0 (or 0.0.0.0 in dotted decimal notation) is reserved and MUST NOT be used. When IPv4 addresses are in use, the value of this object can be derived from the LSR's /32 IPv4 loop back address.

MplsLocalId	This textual convention is used in accommodating the bigger size Global_Node_ID and/or ICC with lower size LSR identifier in order to index mplsTunnelTable. The Local Identifier is configured between 1 and 16777215, as the valid IP address range starts from 16777216 (01.00.00.00). This range is chosen to identify the mplsTunnelTable's Ingress/Egress. LSR-id is the IP address or local identifier. If the configured range is not an IP address, the administrator is expected to retrieve the complete information (Global_Node_ID) from mplsNodeConfigTable. This way, the existing mplsTunnelTable is reused for bidirectional tunnel extensions for MPLS-based transport networks.

CISCO-MPLS-ID-EXT-STD-MIB

This MIB module contains generic object definitions for MPLS Traffic Engineering in transport networks.

Object	Description (from IETF draft)
mplsGlobalId	This object allows the administrator to assign a unique operator identifier also called MPLS-TP Global_ID.
mplsNodeId	This object allows the operator or service provider to assign a unique MPLS-TP Node_ID. The Node_ID is assigned within the scope of the Global_ID.

MPLS LSR STD MIB

Existing Label Switch Router (LSR) MIB functions are used to fetch values for the tables below. For TP, an FPI type of FPI_IF4 is used for IPv4. Only IPv4 is supported in this release.

- **At the endpoints.** For each tunnel, there is one entry for mplsOutSegmentTable [RFC 3813] showing the outsegment label and one entry for mplsInSegmentTable [RFC 3813] for the working LSP. Similarly, an entry is shown to the protected LSP. The assumption is that both working and protected LSPs are configured. If only one working LSP and one protected LSP is configured, the entries are displayed accordingly. There are 2 entries per tunnel for mplsXCTable [RFC 3813] for a working LSP and similarly to a protected LSP.

- **At the midpoints.** For a co-routed bidirectional tunnel, a midpoint has forward and reverse LSPs configured. Thus, there are a pair of mplsInSegmentTable and mplsOutSegmentTable entries for the forward LSP and a pair of mplsInSegmentTable and mplsOutSegmentTable entries for the reverse LSP. If the working and protected LSPs are configured then the above listed entries are shown for both protected and working LSPs. For mplsXCTable, there are two entries—one for the forward LSP and one for the reverse LSP. If the config has working and protected LSPs configured, then the above listed mplsXCTable entries are shown for both protected and working LSPs.

- **Indexing for mplsOutSegmentTable, mplsInSegmentTable and mplsXCTable.** mplsXCTable is indexed by mplsXCIndex [RFC3813], mplsXCInSegmentIndex [RFC3813], and mplsXCOutSegmentIndex [RFC3813]. The mplsXCInSegmentIndex, which is the same as mplsInSegmentIndex, is a 4-byte octet string containing the local label. The mplsXCIndex for TP is represented in the octet string format. The FPI value of FPI_IF4 is taken from file lsd_common_issu_sensitive.enum. The FPI value of 3 is used for TP.
  - At the endpoint, mplsXCIndex is represented as an octet string that contains fpi_type, tunnel index, and the LSP identifier. The LSP identifier specifies if the LSP is working or protected. The LSP identifier can be of either two types: CFC_MPLS_CP_LSP_TYPE_WORKING - working LSP (integer value 2) or CFC_MPLS_CP_LSP_TYPE_PROTECT - protected LSP (integer value 3).

```
|----|          |----||----||----||----|         |----|
FPI = 3         Tunnel-id                    LSP_ident
```

Note Internally, Tunnel-id is used to get if_number (outgoing interface) and if_number is used to poll MFI where

```
|----|
```

equals 1 byte.

  - At the midpoint, mplsXCIndex is represented by an octet string that contains fpi_type and in-label. The Fpi_type value is 0 for label.

```
|----|          |----||----||----||----|
FPI = 0         Label
```

mplsXCOutSegmentIndex is the same as mplsOutSegmentIndex, which is the same as mplsXCIndex plus moi_index. The last two bytes in mplsOutSegmentIndex contain the MOI list index.

A new cfc_mpls_cp_lsrmib_rfc_get_tp_label_id MIB function will be created for the MIB team to fetch TP-related data.

Object	Value and function used to get the value
mplsOutSegmentTable	
mplsOutSegmentIndex	This object contains the outsegment index as explained above. The cfc_mpls_cp_lsrmib_rfc_get_outseg_entry function is used to get this value.
mplsOutSegmentInterface	This object contains the outsegment interface that comes from the IDB. The cfc_mpls_cp_lsrmib_rfc_get_outseg_entry function is used to get this value.
mplsOutSegmentPushTopLabel	This object is set to D_mplsOutSegmentPushTopLabel_true.
mplsOutSegmentTopLabel	The lsrmib_get_top_label function is used to get this value.

mplsOutSegmentTopLabelPtr	Set to 0.0.
mplsOutSegmentNextHopAddrType	The value of mfi_out_info.nh.type provides the value of this object.
mplsOutSegmentNextHopAddr	The value of mfi_out_info.nh.ip_addr provides the value of this object.
mplsOutSegmentXCIndex	This object contains mplsXCIndex from mplsXCTable. The cfc_mpls_cp_lsrmib_rfc_get_xc_search_indices function is used to get this value.
mplsOutSegmentOwner	Will add a new a macro: LSRMIB_MPLS_FPI_IF4 and this will map to D_mplsOutSegmentOwner_tp.
mplsOutSegmentTrafficParamPtr	Always set to 0.0.
mplsOutSegmentRowStatus	D_mplsOutSegmentRowStatus_active
mplsOutSegmentStorageType	D_mplsInSegmentStorageType_volatile
mplsOutSegmentPerfTable	
mplsOutSegmentPerfOctets	mfi_out_info.bytes
mplsOutSegmentPerfPackets	mfi_out_info.packets
mplsOutSegmentPerfErrors	mfi_out_info.errors
mplsOutSegmentPerfDiscards	mfi_out_info.discards
mplsOutSegmentPerfHCOctets	Get from MFI.
mplsOutSegmentPerfDiscontinuityTime	lsrmib_get_discontinuity_time()
mplsInSegmentTable	
mplsInSegmentIndex	This object contains the insegmen index as explained above. The lsrmib_get_in_label_id function is used to get the value.
mplsInSegmentInterface	This is set to 0.
mplsInSegmentLabel	lsrmib_get_in_label_id function is used.
mplsInSegmentLabelPtr	Always set to 0.0.
mplsInSegmentNPop	Set to default value 1.
mplsInSegmentAddrFamily	Set to D_mplsInSegmentAddrFamily_ipV4.
mplsInSegmentXCIndex	This object contains mplsXCIndex. The cfc_mpls_cp_lsrmib_rfc_mfi_info_to_xc function is used to get this value.

mplsInSegmentOwner	D_mplsInSegmentOwner_other
mplsInSegmentTrafficParamPtr	0.0
mplsInSegmentRowStatus	D_mplsInSegmentRowStatus_active
mplsInSegmentStorageType	D_mplsInSegmentStorageType_volatile
mplsInSegmentPerfTable	
mplsInSegmentPerfOctets	mfi_out_info.bytes
mplsInSegmentPerfPackets	mfi_out_info.packets
mplsInSegmentPerfErrors	mfi_out_info.errors
mplsInSegmentPerfDiscards	mfi_out_info.discards
mplsInSegmentPerfHCOctets	Get from MFI.
mplsInSegmentPerfDiscontinuityTime	lsrmib_get_discontinuity_time()
mplsXCTable	
mplsXCIndex	cfc_mpls_cp_lsrmib_rfc_get_xc_search_indices function is used to get this value.
mplsXCInSegmentIndex	cfc_mpls_cp_lsrmib_rfc_get_xc_search_indices function is used to get this value.
mplsXCOutSegmentIndex	cfc_mpls_cp_lsrmib_rfc_get_xc_search_indices function is used to get this value.
mplsXCLSPId	cfc_mpls_cp_lsrmib_rfc_get_xc_search_indices is used to get this value.
mplsXCLabelStackIndex	This object contains the octet string 0.0. which indicates that no labels are to be stacked beneath the top label.
mplsXCOwner	RFC LSR MIB doesn't provide a specific value for TP. Thus, D_mplsXCOwner_other is used to fetch this value.
mplsXCRowStatus	Set to D_mplsXCRowStatus_active.
mplsXCStorageType	Set to D_mplsXCStorageType_volatile.
mplsXCAdminStatus	Set to D_mplsXCAdminStatus_up.
mplsXCOperStatus	Set to D_mplsXCOperStatus_up.

CISCO-MPLS-LSR-EXT-STD-MIB

mplsXCExtEntry: An entry in this table extends the cross connect information represented by an entry in the mplsXCTable through a sparse augmentation. The indices for this table are mplsXCIndex, mplsXCInSegmentIndex, and mplsXCOutSegmentIndex.

- **Midpoint.** At the midpoint there are 2 entries, one for the forward LSP and one for the reverse LSP. If both working and protected LSPs are configured, then there will be 2 entries for each of the LSPs.
- **Endpoint.** At the endpoint there are two entries in mplsXCExtTunnelPointer. If both working and protected LSPs are configured, then there will be 2 entries for each LSP.

Object	Description	Value and function used to get the value
mplsXCExtTunnelPointer	This object indicates the back pointer to the tunnel entry segment. This object cannot be modified if mplsXCRowStatus for the corresponding entry in the mplsXCTable is active(1).	Both the entries (per tunnel) point to the same tunnel entry. A new function to fetch this information from TP will be created. At the endpoint, the MIB code provides the tunnel number and the LSP identifier (working/protected) and expects in return from the TP the other two tunnel indices—the local ID for the source and the local ID for the destination of this tunnel. At midpoint, the MIB code provides the incoming label and expects the TP to return the unique tunnel entry that provides the tunnel index, LSP instance, source-local-id, and destination-local-id.

mplsXCOppositeDirXCPtr	This object indicates the pointer to the opposite direction XC entry. This object cannot be modified if mplsXCRowStatus for the corresponding entry in the mplsXCTable is active(1).	For the endpoint, there are two entries for this object. At the endpoint, the entry that represents the outgoing segment contains the mplsXCLspId entry that corresponds to the reverse direction in-label. The entry that corresponds to the in-label contains the mplsXCLspId representing the outgoing segment (so, in essence, contains the indices with FPI type 3 for the TP tunnel). For the midpoint, there are two entries for this object. Each entry contains the mplsXCLspId representing the reverse direction in-label.

MPLS-TE-STD-MIB and MPLS Draft TE MIB

mplsTunnelTable from MPLS-TE-STD-MIB shows TP tunnel entries. For details on object description, refer to RFC 3812. Protected LSP is assumed to be configured for every working LSP.

TP configuration allows partial configuration. If an LSP is partially configured where destination node-id/global ID is not specified, then the local-id is set to 0.

- **Endpoint.** mplsTunnelTable has one entry per LSP.
- **Midpoint.** For the working LSP, mplsTunnelTable has one entry for the forward LSP and one entry for the reverse LSP. Similarly, if the protected LSP is configured, entries for the protected LSP are shown.

Object	Value and function used to get the value
mplsTunnelIndex	At an endpoint, mplsTunnelIndex contains the source tunnel number. At a midpoint, the mplsTunnelIndex contains the source tunnel number for the forward LSP and the destination tunnel number for the reverse LSP.
mplsTunnelInstance	This contains the LSP number. The tp_get_tunnel_detail function is used to get this value.
mplsTunnelIngressLSRId	At an endpoint, this contains the value of mplsNodeConfigLocalId for the source of the tunnel. At a midpoint, it stores the mplsNodeConfigLocalId for the source of the tunnel for the forward LSP and mplsNodeConfigLocalId for the destination of the reverse LSP. This value ranges between 1 and 16777215. The tp_get_tunnel_detail function is used to get this value.

mplsTunnelEgressLSRId	At an endpoint, this contains the value of mplsNodeConfigLocalId for the destination node of the tunnel. At a midpoint, it stores the mplsNodeConfigLocalId for the destination of the tunnel for the forward LSP and mplsNodeConfigLocalId for the source of the tunnel for the reverse LSP. This value ranges between 1 and 16777215. The tp_get_tunnel_detail function is used to get this value.
mplsTunnelName	This contains the tunnel name, applicable at both endpoint and midpoint. The tp_get_tunnel_detail function is used to get this value.
mplsTunnelDescr	This contains the tunnel description. The tp_get_tunnel_detail function is used to get this value.
mplsTunnelIsIf	This is always true because the TP tunnel is always an interface.
mplsTunnelIfIndex	This contains the tunnel ifindex. The tp_get_tunnel_detail function provides the IF number. The interface number can be used to get the interface index.
mplsTunnelOwner	This is set to D_mplsTunnelOwner_other.
mplsTunnelRole	The tp_get_tunnel_detail function is used to get this value.
mplsTunnelXCPointer	The cfc_mpls_cp_lsrmib_rfc_make_XC_pointer function is used.
mplsTunnelSignallingProto	None(1). The MPLS TP implementation on Cisco IOS does not have a control plane and there is no signaling protocol.
mplsTunnelSetupPrio	0. By default, MPLS TP LSPs have 0 priority.
mplsTunnelHoldingPrio	0. By default, MPLS TP LSPs have 0 priority.
mplsTunnelSessionAttributes	N/A. 0.
mplsTunnelLocalProtectInUse	This object indicates whether a protected LSP is being used. The tp_get_tunnel_detail function is used to get this value.
mplsTunnelResourcePointer	0.0. Not supported.
mplsTunnelPrimaryInstance	This is used to indicate the LSP number of the working LSP. If the working LSP is not configured, then this shows a default value of 0.

mplsTunnelInstancePriority	N/A. 0.
mplsTunnelHopTableIndex	N/A. 0.
mplsTunnelPathInUse	N/A. 0.
mplsTunnelARHopTableIndex	N/A. 0.
mplsTunnelCHopTableIndex	N/A. 0.
mplsTunnelIncludeAnyAffinity	N/A. 0.
mplsTunnelIncludeAllAffinity	N/A. 0.
mplsTunnelTotalUpTime	The tp_get_tunnel_detail function is used to get this value.
mplsTunnelInstanceUpTime	The tp_get_tunnel_detail function is used to get this value.
mplsTunnelPrimaryUpTime	The tp_get_tunnel_detail function is used to get this value.
mplsTunnelPathChanges	N/A. 0.
mplsTunnelLastPathChange	N/A.
mplsTunnelCreationTime	The tp_get_tunnel_detail function is used to get this value.
mplsTunnelStateTransitions	N/A. 0.
mplsTunnelAdminStatus	At endpoint, the tp_get_tunnel_detail function is used to get this value. At midpoint, this is set to "testing(3)" as the TP does not maintain admin status at the midpoint.
mplsTunnelOperStatus	At endpoint, the tp_get_tunnel_detail function is used to get this value. At midpoint, this is set to "testing(3)" as the TP does not maintain oper status at the midpoint.
mplsTunnelRowStatus	D_mplsTunnelRowStatus_active
mplsTunnelStorageType	D_mplsTunnelStorageType_readOnly
mplsTunnelPerfTable: This counter is not supported.	

CISCO-MPLS-TE-EXT-STD-MIB

This MIB module contains generic object definitions for MPLS Traffic Engineering in transport networks.

Object	Description (as IETF draft defines it)	Value and function used to get the value
mplsNodeConfigTable		
mplsNodeConfigLocalId	This object allows the administrator to assign a unique local identifier to map Global_Node_ID.	This table is used to represent a node in a TP network. This object provides a unique local value for the node. The value of this object lies between 1 and 16777215. The TP provides a new tp_get_node_detail function. This is used to get this object's value.
mplsNodeConfigGlobalId	This object indicates the Global Operator Identifier.	This maps to the mpls_tp_global_id_t global_id field of the TP data structure. tp_get_node_detail is used to get this object's value.
mplsNodeConfigNodeId	This object indicates the Node_ID within the operator. This object value should be zero when mplsNodeConfigIccId is configured with non-null value.	This object maps to mpls_tp_node_id_t node_id field of TP data structure. The tp_get_node_detail function is used to get this object's value.
mplsNodeConfigIccId	This object allows the operator or service provider to configure a unique MPLS-TP ITU-T Carrier Code (ICC) either for Ingress ID or Egress ID. This object value should be zero when mplsNodeConfigGlobalId and mplsNodeConfigNodeId are assigned with a non-zero value.	This object is set to 0. Cisco IOS implementation only supports IP-compatible implementation.
mplsNodeConfigRowStatus	This object allows the administrator to create, modify, and/or delete a row in this table.	This is set to 'active'.
mplsNodeConfigStorageType	This variable indicates the storage type for this object. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row.	This is set to 'readonly' because write access to any object is not allowed.
mplsNodeIpMapTable: This table is indexed by mplsNodeIpMapNodeId and mplsNodeIpMapLocalId		
mplsNodeIpMapGlobalId	This object indicates the Global_ID.	The tp_get_node_detail function is used to get this object's value.

mplsNodeIpMapNodeId	This object indicates the Node_ID within the operator.	The tp_get_node_detail function is used to get this object's value.
mplsNodeIpMapLocalId	This object contains an IP compatible local identifier that is defined in mplsNodeConfigTable.	The tp_get_node_detail function is used to get this object's value.
mplsTunnelExtTable : The indices of this table are the same as mplsTunnelTable (RFC 3812)		
mplsTunnelOppositeDirPtr	This object is applicable only for the bidirectional tunnel that has the forward and reverse LSPs in the same tunnel or in different tunnels. This object holds the opposite direction tunnel entry if the bidirectional tunnel is set up by configuring two tunnel entries in mplsTunnelTable. The value of zeroDotZero indicates single tunnel entry is used for bidirectional tunnel setup.	Because only one entry per tunnel per LSP for mplsTunnelTable is shown, this object will contain the value 0.0.
mplsTunnelReversePerfTable: This counter is not supported.		
mplsNodeIccMapTable: Because only IP-compatible implementation of the TP is supported, this table is not supported.		

How to Configure MPLS-TP MIB

Configuring MPLS-TP MIB

A generic SNMP configuration automatically enables MPLS-TP MIB. However, the MPLS TP feature must be configured. See the MPLS Transport Profile document for more information.

You should perform the following generic SNMP configuration tasks:

- Enabling the SNMP agent (required)
- Verifying the status of the SNMP agent (optional)

Enabling the SNMP Agent

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**][*number*]

5. **end**
6. **write memory**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**][*number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) community strings for the MPLS-TP MIB. • The *string* argument functions like a password, permitting access to SNMP functionality on label switch routers (LSRs) in an MPLS network. • The optional **ro** keyword configures read-only (ro) access to the objects in the MPLS-TP MIB.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.
Step 7	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router so that you can determine if an SNMP agent is already running on the device.

	Command or Action	Purpose
		If you see any snmp-server statements, SNMP has been enabled on the router. If any SNMP information is displayed, you can modify the information or change it as desired.

Verifying the Status of the SNMP Agent

To verify that the SNMP agent has been enabled on a host network device, perform the steps shown in the following table:

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Device# show running-config`	Displays the running configuration on the target device.

Configuration Examples for MPLS-TP MIB

Example Enabling the SNMP Agent

The following example shows how to enable an SNMP agent on a host network device.

```
Device# config terminal
Device(config)# snmp-server community
```

The following example shows how to enable SNMPv1 and SNMPv2C. The configuration permits any SNMP agent to access all MPLS TP MIB objects with read-only permissions using the community string *public*.

```
Device(config)#  snmp-server community public
```

The following example shows how to allow read-only access to all MPLS TP MIB objects relating to members of access list 4 that specify the *comaccess* community string. No other SNMP agents will have access to any MPLS TP MIB objects.

```
Device(config)#  snmp-server community comaccess ro 4
```

Example Verifying the Status of the SNMP Agent

The following example shows how to verify the status of the SNMP agent.

```
Device# show running-config
...

...

snmp-server community public RO

snmp-server community private RO
```

Any snmp-server statement that appears in the output and which takes the form shown above verifies that SNMP has been enabled on that device.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS Transport Profile configuration document	MPLS Transport Profile

Standards and RFCs

Standard/RFC	Title
draft-ietf-mpls-tp-te-mib-02.txt	MPLS-TP Traffic Engineering (TE) Management Information Base (MIB)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS-TP MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

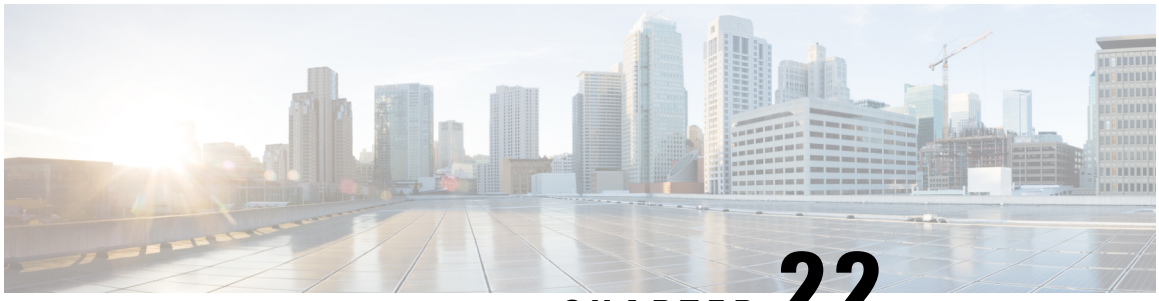
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 78: Feature Information for MPLS-TP MIB

Feature Name	Releases	Feature Information
MPLS-TP MIB	15.3(1)S XE 3S	Allows you to meet your transport requirements as those requirements evolve from Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) time-division multiplexing (TDM) technologies to MPLS and Ethernet technologies.

PART III

MPLS High Availability



CHAPTER 22

MPLS LDP Graceful Restart

When a router is configured with Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) Graceful Restart (GR), it assists a neighboring router that has MPLS LDP Stateful Switchover/Nonstop Forwarding (SSO/NSF) Support and Graceful Restart to recover gracefully from an interruption in service. MPLS LDP GR functions strictly in helper mode, which means it can only help other routers that are enabled with MPLS SSO/NSF and GR to recover. If the router with LDP GR fails, its peer routers cannot help the router recover.

For brevity, the following are used in this document:

- MPLS LDP SSO/NSF Support and Graceful Restart is called LDP SSO/NSF.
- The MPLS LDP GR feature described in this document refers to helper mode.

When you enable MPLS LDP GR on a router that peers with an MPLS LDP SSO/NSF-enabled router, the SSO/NSF-enabled router can maintain its forwarding state when the LDP session between them is interrupted. While the SSO/NSF-enabled router recovers, the peer router forwards packets using stale information. This enables the SSO/NSF-enabled router to become operational more quickly.

Prerequisites for MPLS LDP Graceful Restart

You must enable MPLS LDP GR on all route processors for an LDP session to be preserved during an interruption in service.

Restrictions for MPLS LDP Graceful Restart

- MPLS LDP GR is supported in strict helper mode.
- MPLS LDP GR cannot be configured on label-controlled ATM (LC-ATM) interfaces.

Information About MPLS LDP Graceful Restart

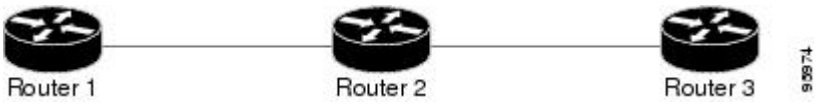
How MPLS LDP Graceful Restart Works

MPLS LDP GR works in strict helper mode, which means it helps a neighboring route processor that has MPLS LDP SSO/NSF to recover from disruption in service without losing its MPLS forwarding state. The disruption in service could be the result of a TCP or UDP event or the stateful switchover of a route processor. When the neighboring router establishes a new session, the LDP bindings and MPLS forwarding states are recovered.

In the topology shown in the figure below, the following elements have been configured:

- LDP sessions are established between Router 1 and Router 2, as well as between Router 2 and Router 3.
- Router 2 has been configured with MPLS LDP SSO/NSF. Routers 1 and 3 have been configured with MPLS LDP GR.
- A label switched path (LSP) has been established between Router 1 and Router 3.

Figure 40: Example of a Network Using LDP Graceful Restart



The following process shows how Routers 1 and 3, which have been configured with MPLS LDP GR, help Router 2, which has been configured with LDP SSO/NSF, recover from a disruption in service:

1. Router 1 notices an interruption in service with Router 2. (Router 3 also performs the same actions in this process.)
2. Router 1 marks all the label bindings from Router 2 as stale, but it continues to use the bindings for MPLS forwarding.

Router 1 reestablishes an LDP session with Router 2, but keeps its stale label bindings. If you issue a **showmplsldpneighbor** command with the **graceful-restart** keyword, the command output displays the recovering LDP sessions.

1. Both routers readvertise their label binding information. If Router 1 relearns a label from Router 2 after the session has been established, the stale flags are removed. The **showmplsforwarding-table**command displays the information in the MPLS forwarding table, including the local label, outgoing label or VC, prefix, label-switched bytes, outgoing interface, and next hop.

You can set various graceful restart timers. See the following commands for more information:

- **mpls ldp graceful-restart timers neighbor-liveness**
- **mpls ldp graceful-restart timers max-recovery**

How a Route Processor Advertises That It Supports MPLS LDP Graceful Restart

A Route Processor (RP) that is configured to perform MPLS LDP GR includes the Fault Tolerant (FT) Type Length Value (TLV) in the LDP initialization message. The RP sends the LDP initialization message to a neighbor to establish an LDP session.

The FT session TLV includes the following information:

- The Learn from Network (L) flag is set to 1, which indicates that the route processor is configured to perform MPLS LDP GR.
- The Reconnect Timeout field shows the time (in milliseconds) that the neighbor should wait for a reconnection if the LDP session is lost. In this release, the timer is set to 0, which indicates that if the local router fails, its peers should not wait for it to recover. The timer setting indicates that the local router is working in helper mode.
- The Recovery Time field shows the time (in milliseconds) that the neighbor should retain the MPLS forwarding state during a recovery. If a neighbor did not preserve the MPLS forwarding state before the restart of the control plane, the neighbor sets the recovery time to 0.

What Happens If a Route Processor Does Not Have MPLS LDP Graceful Restart

If two route processors establish an LDP session and one route processor is not configured for MPLS LDP GR, the two route processors create a normal LDP session but do not have the ability to perform MPLS LDP GR. Both route processors must be configured for MPLS LDP GR.

How to Configure MPLS LDP Graceful Restart

Configuring MPLS LDP Graceful Restart

To configure MPLS LDP Graceful Restart, perform the following task.

You must enable MPLS LDP GR on all route processors for an LDP session to be preserved during an interruption in service.

MPLS LDP GR is enabled globally. When you enable MPLS LDP GR, it has no effect on existing LDP sessions. New LDP sessions that are established can perform MPLS LDP GR.

Note You can also issue the **mpls label protocol ldp** command in global configuration mode, which enables LDP on all interfaces configured for MPLS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **mpls ldp graceful-restart**

5. **interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*]
6. **mpls ip**
7. **mpls label protocol ldp**
8. **exit**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding.
Step 4	**mpls ldp graceful-restart** **Example:** `Router(config)# mpls ldp graceful-restart`	Enables the router to protect the LDP bindings and MPLS forwarding state during a disruption in service.
Step 5	**interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*] **Example:** `Router(config)# interface pos 0/3/0`	Specifies an interface and enters interface configuration mode.
Step 6	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Configures MPLS hop-by-hop forwarding for an interface.
Step 7	**mpls label protocol ldp** **Example:** `Router(config-if)# mpls label protocol ldp`	Configures the use of LDP for an interface.
Step 8	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the MPLS LDP Graceful Restart Configuration

To verify that MPLS LDP Graceful Restart is configured correctly, perform the following task.

SUMMARY STEPS

1. **enable**
2. **show mpls ldp neighbor graceful restart**
3. **show mpls ldp graceful-restart**
4. **exit**

DETAILED STEPS

Step 1 enable

Use this command to enable privileged ECEC mode. Enter your password if prompted. For example:

Example:

```
Router>? enable
Router#
```

Step 2 show mpls ldp neighbor graceful restart

Use this command to display graceful restart information for LDP sessions. For example:

Example:

```
Router# show mpls ldp neighbor graceful restart
Peer LDP Ident: 10.20.20.20:0; Local LDP Ident 10.17.17.17:0
   TCP connection: 10.20.20.20.16510 - 10.17.17.17.646
    State: Oper; Msgs sent/rcvd: 8/18; Downstream
    Up time: 00:04:39
    Graceful Restart enabled; Peer reconnect time (msecs): 120000
Peer LDP Ident: 10.19.19.19:0; Local LDP Ident 10.17.17.17:0
    TCP connection: 10.19.19.19.11007 - 10.17.17.17.646
    State: Oper; Msgs sent/rcvd: 8/38; Downstream
    Up time: 00:04:30
    Graceful Restart enabled; Peer reconnect time (msecs): 120000
```

Step 3 show mpls ldp graceful-restart

Use this command to display graceful restart sessions and session parameters. For example:

Example:

```
Router# show mpls ldp graceful-restart
LDP Graceful Restart is enabled
Neighbor Liveness Timer: 5 seconds
Max Recovery Time: 200 seconds
Down Neighbor Database (0 records):
Graceful Restart-enabled Sessions:
VRF default:
    Peer LDP Ident: 10.18.18.18:0, State: estab
    Peer LDP Ident: 10.17.17.17:0, State: estab
```

Step 4 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

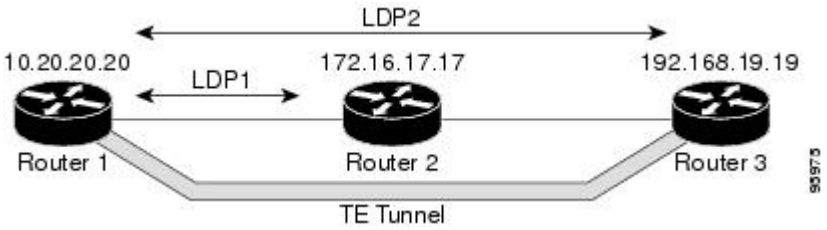
```
Router# exit
Router>
```

Configuration Examples for MPLS LDP Graceful Restart

Configuring MPLS LDP Graceful Restart Example

The figure below shows a configuration where MPLS LDP GR is enabled on Router 1 and MPLS LDP SSO/NSF is enabled on Routers 2 and 3. In this configuration example, Router 1 creates an LDP session with Router 2. Router 1 also creates a targeted session with Router 3 through a traffic engineering tunnel using Router 2.

Figure 41: MPLS LDP Graceful Restart Configuration Example



Router 1 configured with LDP GR:

```
!
ip subnet-zero
ip cef
mpls label range 16 10000 static 10001 1048575
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls ldp graceful-restart
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp router-id Loopback0 force
!
```

```
interface Loopback0
 ip address 20.20.20.20 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Tunnel1
 ip unnumbered Loopback0
 no ip directed-broadcast
 mpls label protocol ldp
 mpls ip
 tunnel destination 19.19.19.19
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface ATM5/1/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM5/1/0.5 point-to-point
 ip address 10.12.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 pvc 6/100
  encapsulation aal5snap
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ip
ip rsvp bandwidth 1000
!
router ospf 100
 log-adjacency-changes
 redistribute connected
     network 10.12.0.0 0.255.255.255 area 100
 network 10.20.20.20 0.0.0.0 area 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 100
```

Router 2 configured with LDP SSO/NSF:

```
!
redundancy
  mode sso
!
ip cef
no ip domain-lookup
mpls label range 17 10000 static 10001 1048575
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls ldp graceful-restart
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
no mpls advertise-labels
mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.17.17.17 255.255.255.255
 no ip directed-broadcast
```

```
!
interface ATM4/0/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM4/0/0.5 point-to-point
 ip address 10.12.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 pvc 6/100
  encapsulation aal5snap
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ip
ip rsvp bandwidth 1000
!
interface POS5/1/0
 ip address 10.11.0.1 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
 mpls label protocol ldp
 mpls traffic-eng tunnels
 mpls ip
 no peer neighbor-route
 clock source internal
 ip rsvp bandwidth 1000
!
router ospf 100
 log-adjacency-changes
 redistribute connected
 nsf enforce global
 network 10.11.0.0 0.255.255.255 area 100
 network 10.12.0.0 0.255.255.255 area 100
 network 10.17.17.17 0.0.0.0 area 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 100
!
ip classless
```

Router 3 configured with LDP SSO/NSF:

```
!
redundancy
  mode sso
!
ip subnet-zero
ip cef
!
no ip finger
no ip domain-lookup
mpls label protocol ldp
mpls ldp neighbor 10.11.11.11 targeted ldp
mpls ldp logging neighbor-changes
mpls ldp graceful-restart
mpls traffic-eng tunnels
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp discovery directed-hello interval 12
mpls ldp discovery directed-hello holdtime 130
```

```
mpls ldp discovery directed-hello accept
mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.19.19.19 255.255.255.255
 no ip directed-broadcast
!
interface POS1/0
 ip address 10.11.0.2 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
 mpls label protocol ldp
 mpls traffic-eng tunnels
 mpls ip
 no peer neighbor-route
 clock source internal
 ip rsvp bandwidth 1000
!
router ospf 100
 log-adjacency-changes
 redistribute connected
 nsf enforce global
network 10.11.0.0 0.255.255.255 area 100
network 10.19.19.19 0.0.0.0 area 100
mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 100
!
ip classless
```

Additional References

The following sections provide references related to MPLS LDP GR.

Related Documents

Related Topic	Document Title
MPLS Label Distribution Protocol	MPLS Label Distribution Protocol (LDP)
LDP commands	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
MPLS Label Distribution Protocol MIB Version 8 Upgrade	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3036	*LDP Specification*
RFC 3478	*Graceful Restart Mechanism for Label Distribution*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

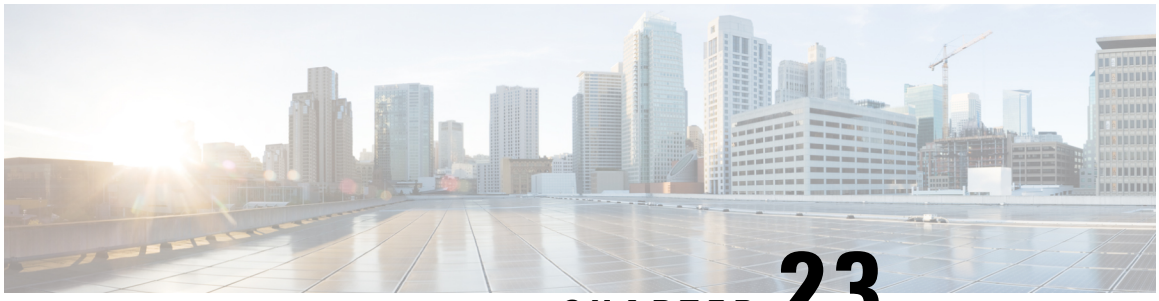
Feature Information for MPLS LDP Graceful Restart

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 79: Feature Information for MPLS LDP Graceful Restart

Feature Name	Releases	Feature Information
MPLS LDP Graceful Restart	Cisco IOS XE Release 2.1	When a router is configured with Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) Graceful Restart (GR), it assists a neighboring router that has MPLS LDP Stateful Switchover/Nonstop Forwarding (SSO/NSF) Support and Graceful Restart to recover gracefully from an interruption in service. MPLS LDP GR functions strictly in helper mode, which means it can only help other routers that are enabled with MPLS SSO/NSF and GR to recover. If the router with LDP GR fails, its peer routers cannot help the router recover. In Cisco IOS XE Release 2.1, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **debugmplsldpgraceful-restart**, **mplsldpgraceful-restart**, **mplsldpgraceful-restarttimersmax-recovery**, **mplsldpgraceful-restarttimersneighbor-liveness**, **showmplsipbinding**, **showmplsldpbindings**, **showmplsldpgraceful-restart**, **showmplsldpneighbor**.



CHAPTER 23

Configuring NSF SSO--MPLS VPN

The NSF/SSO--MPLS VPN feature allows a provider edge (PE) router to preserve data forwarding information in a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) when the primary Route Processor (RP) restarts. This module describes how to enable nonstop forwarding (NSF) in a basic MPLS VPN network.

Restrictions for NSF SSO--MPLS VPN

- Tag Distribution Protocol (TDP) sessions are not supported. Only Label Distribution Protocol (LDP) sessions are supported.
- The NSF/SSO--MPLS VPN feature cannot be configured on label-controlled ATM (LC-ATM) interfaces.

Information About NSF SSO--MPLS VPN

Elements That Enable NSF SSO--MPLS VPN

VPN NSF requires several elements in order to work:

- VPN NSF uses the BGP Graceful Restart mechanisms to create MPLS forwarding entries for VPNv4 prefixes in NSF mode. The forwarding entries are preserved during a restart. BGP also saves prefix and corresponding label information and recovers the information after a restart.
- The NSF/SSO--MPLS VPN feature also uses NSF for the label distribution protocol in the core network (either MPLS Label Distribution Protocol, traffic engineering, or static labeling).
- The NSF/SSO--MPLS VPN feature uses NSF for the Interior Gateway Protocol (IGP) used in the core (OSPF or IS-IS).
- The NSF/SSO--MPLS VPN feature uses NSF for the routing protocols between the PE and CE routers.

How VPN Prefix Information Is Checkpointed to the Backup Route Processor

When BGP allocates local labels for prefixes, it checkpoints the local label binding in the backup RP. The checkpointing function copies state information from the active RP to the backup RP, thereby ensuring that the backup RP has an identical copy of the latest information. If the active RP fails, the backup RP can take over with no interruption in service. Checkpointing begins when the active RP does a bulk synchronization, which copies all of the local label bindings to the backup RP. After that, the active RP dynamically checkpoints individual prefix label bindings when a label is allocated or freed. This allows forwarding of labeled packets to continue before BGP reconverges.

How BGP Graceful Restart Preserves Prefix Information During a Restart

When a BGP Graceful Restart-capable router loses connectivity, it performs the following actions as the restarting router:

1. The restarting router establishes BGP sessions with other routers and relearns the BGP routes from other routers that are also capable of Graceful Restart. The restarting router waits to receive updates from the neighboring routers. When the neighboring routers send end-of-Routing Information Base (RIB) markers to indicate that they are done sending updates, the restarting router starts sending its own updates.
2. The restarting router accesses the checkpoint database to find the label that was assigned for each prefix. If it finds the label, it advertises it to the neighboring router. If it does not find the label, it allocates a new label and advertises it.
3. The restarting router removes any stale prefixes after a timer for stale entries expires.

A BGP Graceful Restart-capable peer router performs the following actions when it encounters a restarting router:

1. The peer router sends all the routing updates to the restarting router. When it has finished sending updates, the peer router sends an end-of-RIB marker to the restarting router.
2. The peer router does not immediately remove the BGP routes learned from the restarting router from its BGP routing table. As it learns the prefixes from the restarting router, the peer refreshes the stale routes if the new prefix and label information matches the old information.

If a router is not configured for the NSF/SSO--MPLS VPN feature and it attempts to establish a BGP session with a router that is configured with the NSF/SSO--MPLS VPN feature, the two routers create a normal BGP session but do not have the ability to perform the NSF/SSO--MPLS VPN feature.

How to Configure NSF SSO--MPLS VPN

Configuring NSF Support for Basic VPNs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]

4. **router bgp** *autonomous-system-number*
5. **bgp graceful-restart**
6. **bgp graceful-restart restart-time** *seconds*
7. **bgp graceful-restart stalepath-time** *seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** [**distributed**] **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding. • Use this command if Cisco Express Forwarding is not enabled by default on the router.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode.
Step 5	**bgp graceful-restart** **Example:** `Router(config-router)# bgp graceful-restart`	Enables BGP Graceful Restart on the router.
Step 6	**bgp graceful-restart restart-time** *seconds* **Example:** `Router(config-router)# bgp graceful-restart restart-time 200`	(Optional) Specifies the maximum time to wait for a graceful-restart-capable neighbor to come back up after a restart.
Step 7	**bgp graceful-restart stalepath-time** *seconds* **Example:** `Router(config-router)# bgp graceful-restart stalepath-time 400`	(Optional) Specifies the maximum time to hold on to the stale paths of a gracefully restarted peer. All stale paths are deleted after the expiration of this timer.
Step 8	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-router)# end`	

Verifying the Configuration

SUMMARY STEPS

1. **show ip bgp vpnv4 all labels**
2. **show ip bgp vpnv4 all neighbors**
3. **show ip bgp labels**
4. **show ip bgp neighbors**

DETAILED STEPS

Step 1 **show ip bgp vpnv4 all labels**

This command displays incoming and outgoing BGP labels for each route distinguisher. The following is sample output from the command:

Example:

```
Router# show ip bgp vpnv4 all labels

Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (vpn1)
   10.3.0.0/16      10.0.0.5        25/20
                    10.0.0.1        25/23
                    10.0.0.2        25/imp-null
   10.0.0.9/32      10.0.0.1        24/22
                    10.0.0.2        24/imp-null
```

Step 2 **show ip bgp vpnv4 all neighbors**

This command displays whether the BGP peers are capable of Graceful Restart. The following is sample output from the command:

Example:

```
Router# show ip bgp vpnv4 all neighbors
BGP neighbor is 10.0.0.1,  remote AS 100, internal link
  BGP version 4, remote router ID 10.0.0.1
  BGP state = Established, up for 02:49:47
  Last read 00:00:47, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family VPNv4 Unicast: advertised and received
    Graceful Restart Capabilty: advertised and received
      Remote Restart timer is 120 seconds
      Address families preserved by peer:
        VPNv4 Unicast
.
.
.
```

Step 3 **show ip bgp labels**

This command displays information about MPLS labels in the Exterior Border Gateway Protocol (EBGP) route table. The following is sample output from the command:

Example:

```
Router# show ip bgp labels
   Network          Next Hop      In label/Out label
   10.3.0.0/16      10.0.0.1        imp-null/imp-null
                    0.0.0.0         imp-null/nolabel
   10.0.0.9/32      10.0.0.1        21/29
   10.0.0.11/32     10.0.0.1        24/38
   10.0.0.13/32     0.0.0.0         imp-null/nolabel
   10.0.0.15/32     10.0.0.1        29/nolabel
                    10.0.0.1        29/21
```

Step 4 **show ip bgp neighbors**

This command displays whether the BGP peers are capable of Graceful Restart. The following is sample output from the command:

Example:

```
Router# show ip bgp neighbors
BGP neighbor is 10.0.0.1,  remote AS 100, external link
  BGP version 4, remote router ID 10.0.0.5
  BGP state = Established, up for 02:54:19
  Last read 00:00:18, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    ipv4 MPLS Label capability: advertised and received
    Graceful Restart Capabilty: advertised and received
      Remote Restart timer is 120 seconds
      Address families preserved by peer:
        IPv4 Unicast
.
.
.
```

Configuration Examples for NSF SSO--MPLS VPN

Example NSF SSO--MPLS VPN for a Basic MPLS VPN

The following sample output shows the configuration of the NSF/SSO--MPLS VPN feature on the CE and PE routers. SSO is enabled by default, and LDP is the default MPLS label protocol.

CE1 Router

```
ip cef
no ip domain-lookup
!
interface Loopback0
```

```
 ip address 10.10.10.10 255.255.255.255
!
interface GigabitEthernet1/0/4
 ip address 10.0.0.1 255.0.0.0
 media-type 10BaseT
!
router ospf 100
 redistribute bgp 101
 nsf enforce global
 passive-interface GigabitEthernet1/0/4
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 101
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart network 10.0.0.0
 network 10.0.0.0
 neighbor 10.0.0.2 remote-as 100
```

PE1 Router

```
redundancy
mode sso
!
ip cef distributed
mpls ldp graceful-restart
mpls label protocol ldp
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
!
interface GigabitEthernet1/0/4
 ip vrf forwarding vpn1
 ip address 10.0.0.2 255.0.0.0
 !
 mpls ip
interface ATM3/0/0
 no ip address
!
interface ATM3/0/0.1 point-to-point
 ip unnumbered Loopback0
 mpls ip
!
router ospf 100
 passive-interface GigabitEthernet1/0/4
 nsf enforce global
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 10.14.14.14 remote-as 100
 neighbor 10.14.14.14 update-source Loopback0
!
```

```
address-family ipv4 vrf vpn1
 neighbor 10.0.0.1 remote-as 101
 neighbor 10.0.0.1 activate
 exit-address-family
!
address-family vpnv4
 neighbor 10.14.14.14 activate
 neighbor 10.14.14.14 send-community extended
 exit-address-family
```

PE2 Router

```
redundancy
mode sso
!
ip cef distributed
mpls ldp graceful-restart
mpls label protocol ldp
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
no mpls aggregate-statistics
!
!
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
!
interface ATM1/0
 no ip address
!
interface ATM1/0.1 point-to-point
 ip unnumbered Loopback0
 mpls ip
!
interface FastEthernet3/0/0
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 ip route-cache distributed
!
router ospf 100
 nsf enforce global
 passive-interface FastEthernet3/0/0
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 10.12.12.12 remote-as 100
 neighbor 10.12.12.12 update-source Loopback0
!
address-family ipv4 vrf vpn1
 neighbor 10.0.0.2 remote-as 102
 neighbor 10.0.0.2 activate
 exit-address-family
!
address-family vpnv4
 neighbor 10.12.12.12 activate
```

```
 neighbor 10.12.12.12 send-community extended
 exit-address-family
```

CE2 Router

```
ip cef
!
interface Loopback0
 ip address 10.13.13.13 255.255.255.255
!
interface FastEthernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip mroute-cache
!
router ospf 100
 redistribute bgp 102
 nsf enforce global
 passive-interface FastEthernet0/1
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 102
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 network 10.0.0.0
 network 10.0.0.0
 neighbor 10.0.0.1 remote-as 100
```

Additional References

The following sections provide references related to the MPLS High Availability feature.

Related Documents

Related Topic	Document Title
MPLS VPNs Non Stop Forwarding	NSF/SSO—MPLS VPN
MPLS LDP Non Stop Forwarding	*NSF/SSO—MPLS LDP and LDP Graceful Restart*
AToM Non Stop Forwarding	NSF/SSO: Any Transport over MPLS and Graceful Restart
Cisco Express Forwarding	Cisco Express Forwarding: Command Changes
MIBs	• MPLS VPN: SNMP MIB Support • MPLS Label Distribution Protocol MIB Version 8 Upgrade • MPLS Label Switching Router MIB • MPLS Enhancements to Interfaces MIB • MPLS Traffic Engineering (TE) MIB

Related Topic	Document Title
NSF/SSO	Cisco Nonstop Forwarding MPLS High Availability: Command Changes

Standards

Standard	Title
draft-ietf-mpls-bgp-mpls-restart.txt	Graceful Restart Mechanism for BGP with MPLS
draft-ietf-mpls-idr-restart.txt	Graceful Restart Mechanism for BGP

MIBs

MIB	MIBs Link
• MPLS VPN MIB • MPLS Label Distribution Protocol MIB Version 8 Upgrade	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3478	Graceful Restart Mechanism for Label Distribution

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

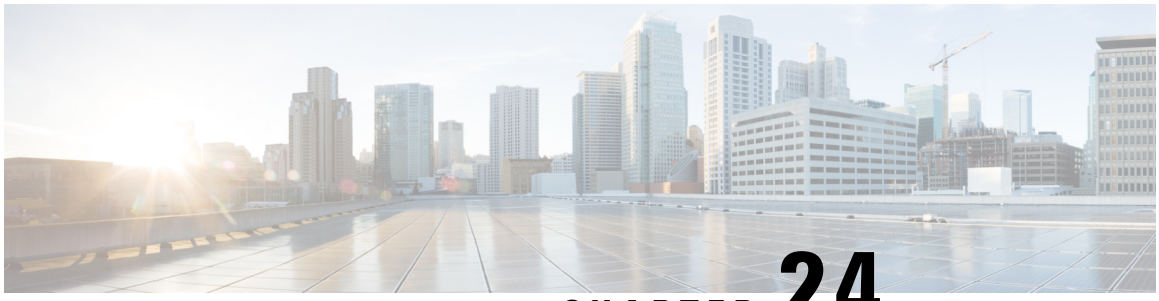
Feature Information for NSF SSO--MPLS VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 80: Feature Information for NSF/SSO--MPLS VPN

Feature Name	Releases	Feature Information
NSF/SSO--MPLS VPN	Cisco IOS XE Release 2.1	This feature allows a provider edge router to preserve data forwarding information in a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) when the primary Route Processor restarts.



CHAPTER 24

ISSU MPLS Clients

MPLS applications can be upgraded using the In Service Software Upgrade (ISSU) process. Thus, MPLS applications are considered ISSU's MPLS clients. The ISSU process allows Cisco IOS XE software to be updated or otherwise modified while packet forwarding continues.

Prerequisites for ISSU MPLS Clients

Before you perform an upgrade, you need to verify that the clients you are concerned about are compatible with the intended switchover. Use the commands listed in the Verifying the ISSU Process for an MPLS Client, on page 455 to determine compatibility.

The success performance of some clients in the upgraded network will depend upon their compatibility with other clients as described in the table below.

Table 81: MPLS Client Interdependencies

This client . . .	...can only work when this client is shown to be compatible
MPLS VPN	LSD Label Manager High Availability
LDP	LSD Label Manager High Availability
VRF ("Table ID")	LSD Label Manager High Availability
LSD Label Manager High Availability	Base clients: Checkpointing and Redundancy Facility
MFI Pull	XDR
MFI Push	XDR
LSPV Push within OAM	XDR

This client . . .	...can only work when this client is shown to be compatible
TE	Base clients: • Checkpointing and Redundancy Facility • MPLS TE High Availability

Information About ISSU MPLS Clients

Before examining ISSU coordination of MPLS clients, you should understand the following concepts:

This section provides information about upgrading MPLS-related applications through ISSU. Those MPLS applications are considered ISSU's MPLS "clients."

For more information on the ISSU procedure, see Cisco IOS XE In Service Software Upgrade Process document and see the Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide .

ISSU-Capable Protocols and Applications Clients

Protocols and applications that can be upgraded through the ISSU process are considered clients of ISSU. These include at least the following:

- Address Resolution Protocol (ARP)
- Asynchronous Transfer Mode (ATM)
- Cisco Express Forwarding
- Dynamic Host Configuration Protocol (DHCP)
- EtherChannel--port aggregration protocol (PagP) and Link Aggregration Control Protocol (LACP)
- Frame Relay (FR)
- Gateway Load Balancing Protocol (GLBP)
- High-Level Data Link Control (HDLC)
- Hot Standby Router Protocol (HSRP)
- IEEE 802.1x and 802.3af
- Internet Group Management Protocol (IGMP) snooping
- IP host
- Intermediate System-to-Intermediate System (IS-IS)
- Multiprotocol Label Switching (MPLS)
- PPP and Multilink PPP
- Port security
- Quality of service (QoS)

- Remote File System (RFS) versioning
- Simple Network Management Protocol (SNMP)
- Spanning Tree Protocol (STP)

Note For a complete list of ISSU- compliant protocols and applications that are supported for the Cisco ASR Series Routers for your release, see the Release Notes for Cisco ASR Series Aggregation Services Routers .

ISSU-Capable MPLS Feature Sets

Within the MPLS technology, ISSU supports the following feature sets as clients:

- Label Distribution Protocol (LDP)
- MPLS Virtual Private Network (MPLS VPN)
- VPN routing and forwarding (VRF), also called the "Table ID" client
- Label Switching Database Label Manager for high availability, usually called "LSD Label Manager for HA"
- MPLS Forwarding Infrastructure Pull, called "MFI Pull"
- MPLS Forwarding Infrastructure Push, called "MFI Push"
- Label Switched Path Verification Push within Operation, Administration, and Management (OAM), called "LSPV Push"
- TE

How to Verify that an MPLS Client Can Support an In Service Software Upgrade

Note For the complete task sequence that accomplishes ISSU see theCisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide .

Verifying the ISSU Process for an MPLS Client

Perform this task to verify that a particular MPLS client can be upgraded successfully during a particular ISSU session. The commands in this task also can be used to display other details about the ISSU MPLS clients, and should be entered in the order described.

Before you begin

Ensure that you have successfully loaded new Cisco IOS XE software onto the standby processor as described in the Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide.

SUMMARY STEPS

1. **enable**
2. **show issu clients**
3. **show issu sessions** *clientID*
4. **show issu negotiated version** *sessionID*
5. **show issu negotiated capability** *sessionID*
6. **show issu message types** *clientID*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show issu clients** **Example:** `Router# show issu clients`	Lists network applications and protocols currently supported by ISSU. • You can use this command to discover the client ID that you will need to enter in Steps 3 and 6.
Step 3	**show issu sessions** *clientID* **Example:** `Router# show issu sessions 2002`	Displays detailed information about a particular ISSU client that includes whether a particular client is compatible with the intended upgrade. • You can use this command to discover the session ID that you will need to enter in Steps 4 and 5.

Step 4	**show issu negotiated version** *sessionID*
	Example:
	`Router#`
	`show issu negotiated version 33`
		Displays details of the session's negotiated message version.

Step 5	**show issu negotiated capability** *sessionID*
	Example:
	`Router#`
	`show issu negotiated capability 33`
		Displays results of a negotiation about the client application's capabilities.

Step 6	**show issu message types** *clientID* **Example:**	Displays the message formats ("types") and versions supported by the specified client.

	Command or Action	Purpose
	`Router# show issu message types 2002`	

Configuration Examples for ISSU MPLS Clients

To examine any ISSU client, you must specify its unique client ID when entering the **show issu sessions** command. If you do not already know that client ID, enter the **show issu clients**command in user EXEC or privileged EXEC mode. Each ISSU client on the network will then be listed, with its client ID and client name on the same line, as shown in the following example:

```
Router# show issu clients
Client_ID = 2,  Client_Name = ISSU Proto client,  Entity_Count = 1
Client_ID = 3,  Client_Name = ISSU RF,  Entity_Count = 1
Client_ID = 4,  Client_Name = ISSU CF client,  Entity_Count = 1
Client_ID = 5,  Client_Name = ISSU Network RF client,  Entity_Count = 1
Client_ID = 7,  Client_Name = ISSU CONFIG SYNC,  Entity_Count = 1
Client_ID = 8,  Client_Name = ISSU ifIndex sync,  Entity_Count = 1
Client_ID = 9,  Client_Name = ISSU IPC client,  Entity_Count = 1
Client_ID = 10,  Client_Name = ISSU IPC Server client,  Entity_Count = 1
Client_ID = 11,  Client_Name = ISSU Red Mode Client,  Entity_Count = 1
Client_ID = 12,  Client_Name = ISSU EHSA services client,  Entity_Count = 1
Client_ID = 100,  Client_Name = ISSU rfs client,  Entity_Count = 1
Client_ID = 110,  Client_Name = ISSU ifs client,  Entity_Count = 1
Client_ID = 1001,  Client_Name = OC3POS-6,  Entity_Count = 4
Client_ID = 1002,  Client_Name = C10K ATM,  Entity_Count = 1
Client_ID = 1003,  Client_Name = C10K CHSTM1,  Entity_Count = 1
Client_ID = 1004,  Client_Name = C10K CT3,  Entity_Count = 1
Client_ID = 1005,  Client_Name = C10K GE,  Entity_Count = 1
Client_ID = 1006,  Client_Name = C10K ET,  Entity_Count = 1
Client_ID = 1007,  Client_Name = C10K CHE1T1,  Entity_Count = 1
Client_ID = 1009,  Client_Name = C10K MFE,  Entity_Count = 1
Client_ID = 1010,  Client_Name = C10K APS,  Entity_Count = 1
Client_ID = 1013,  Client_Name = C10K CARD OIR,  Entity_Count = 1
Client_ID = 2002,  Client_Name = CEF Push ISSU client,  Entity_Count = 1
Client_ID = 2003,  Client_Name = ISSU XDR client,  Entity_Count = 1
Client_ID = 2004,  Client_Name = ISSU SNMP client,  Entity_Count = 1
Client_ID = 2005,  Client_Name = ISSU HDLC Client,  Entity_Count = 1
Client_ID = 2006,  Client_Name = ISSU QoS client,  Entity_Count = 1
Client_ID = 2007,  Client_Name = ISSU LSD Label Mgr HA Client,  Entity_Count = 1
Client_ID = 2008,  Client_Name = ISSU Tableid Client,  Entity_Count = 1
Client_ID = 2009,  Client_Name = ISSU MPLS VPN Client,  Entity_Count = 1
Client_ID = 2010,  Client_Name = ARP HA,  Entity_Count = 1
Client_ID = 2011,  Client_Name = ISSU LDP Client,  Entity_Count = 1
Client_ID = 2012,  Client_Name = ISSU HSRP Client,  Entity_Count = 1
Client_ID = 2013,  Client_Name = ISSU ATM Client,  Entity_Count = 1
Client_ID = 2014,  Client_Name = ISSU FR Client,  Entity_Count = 1
Client_ID = 2015,  Client_Name = ISSU REDSSOC client,  Entity_Count = 1
Client_ID = 2019,  Client_Name = ISSU TCP client,  Entity_Count = 1
Client_ID = 2020,  Client_Name = ISSU BGP client,  Entity_Count = 1
Client_ID = 2021,  Client_Name = XDR Int Priority ISSU client,  Entity_Count = 1
Client_ID = 2022,  Client_Name = XDR Proc Priority ISSU client,  Entity_Count = 1
Client_ID = 2023,  Client_Name = FIB HWIDB ISSU client,  Entity_Count = 1
Client_ID = 2024,  Client_Name = FIB IDB ISSU client,  Entity_Count = 1
Client_ID = 2025,  Client_Name = FIB HW subblock ISSU client,  Entity_Count = 1
Client_ID = 2026,  Client_Name = FIB SW subblock ISSU client,  Entity_Count = 1
Client_ID = 2027,  Client_Name = Adjacency ISSU client,  Entity_Count = 1
Client_ID = 2028,  Client_Name = FIB IPV4 ISSU client,  Entity_Count = 1
```

```
Client_ID = 2030,  Client_Name = MFI Pull ISSU client,  Entity_Count = 1
Client_ID = 2031,  Client_Name = MFI Push ISSU client,  Entity_Count = 1
Client_ID = 2051,  Client_Name = ISSU CCM Client,  Entity_Count = 1
Client_ID = 2052,  Client_Name = ISSU PPP SIP CCM Client,  Entity_Count = 1
Client_ID = 2053,  Client_Name = ISSU MPLS TE Client,  Entity_Count = 1
Client_ID = 2054,  Client_Name = ISSU process client,  Entity_Count = 1
Client_ID = 2089,  Client_Name = MPLS LSPV Push client,  Entity_Count = 1
.
.
.
.
Base Clients:
 Client_Name = ISSU Proto client
 Client_Name = ISSU RF
 Client_Name = ISSU CF client
 Client_Name = ISSU Network RF client
 Client_Name = ISSU CONFIG SYNC
 Client_Name = ISSU ifIndex sync
 Client_Name = ISSU IPC client
 Client_Name = ISSU IPC Server client
 Client_Name = ISSU Red Mode Client
 Client_Name = ISSU EHSA services client
 Client_Name = ISSU rfs client
 Client_Name = ISSU ifs client
 Client_Name = ISSU EM client
 Client_Name = ISSU Platform Medialayer Client
 Client_Name = ISSU FM Client
 Client_Name = ISSU TCAM Manager Client
 Client_Name = ISSU L2 Cmn Client
 Client_Name = ISSU L3 Manager HA Client
 Client_Name = ISSU L3 Manager Client
 Client_Name = ISSU CFIB BASE Client
 Client_Name = ISSU PF CONFIG SYNC Client
 Client_Name = ISSU MLS CEF Client
 Client_Name = ISSU Cat6k Logger Client
```

Verifying the ISSU Process for an MPLS LDP Client Example

This example shows how to verify the ISSU process for an LDP client.

The first command shows you whether the LDP client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2011
------------------------------------------------------------------------
 Client_ID = 2011,  Entity_ID = 1 :
 *** Session_ID = 46,  Session_Name = LDP Session :
    Peer    Peer  Negotiate  Negotiated   Cap      Msg     Session
  UniqueID  Sid     Role       Result   GroupID  GroupID  Signature
     4       34   PRIMARY   COMPATIBLE    1        1         0
                           (no policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 46
         Nego_Session_Name = LDP Session
         Transport_Mtu = 3948
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, to see the negotiated message version:

```
Router# show issu negotiated version 46
 Session_ID = 46 :
```

```
     Message_Type = 1,  Negotiated_Version = 2,  Message_MTU = 20
     Message_Type = 2,  Negotiated_Version = 2,  Message_MTU = 20
     Message_Type = 3,  Negotiated_Version = 2,  Message_MTU = 4
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 46
 Session_ID = 46 :
     Negotiated_Cap_Entry = 1
```

Finally, to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2011
------------------------------------------------------------------------
 Client_ID = 2011,  Entity_ID = 1 :
    Message_Type = 1,  Version_Range = 2 ~ 2
          Message_Ver = 2,    Message_Mtu = 20
    Message_Type = 2,  Version_Range = 2 ~ 2
          Message_Ver = 2,    Message_Mtu = 20
    Message_Type = 3,  Version_Range = 2 ~ 2
          Message_Ver = 2,    Message_Mtu = 4
```

Verifying the ISSU Process for an MPLS VPN Client Example

This example shows how to verify the ISSU process for an MPLS VPN client.

The first command shows you whether the VPN client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2009
------------------------------------------------------------------------
Client_ID = 2009,  Entity_ID = 1 :
*** Session_ID = 39,  Session_Name = MPLS VPN ISSU Session :
   Peer   Peer  Negotiate  Negotiated   Cap      Msg     Session
 UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
    3       33   PASSIVE   COMPATIBLE    1        1         0
                          (no policy)
   Negotiation Session Info for This Message Session:
        Nego_Session_ID = 39
        Nego_Session_Name = MPLS VPN ISSU Session
        Transport_Mtu = 3980
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, in order to see the negotiated message version:

```
Router# show issu negotiated version 39
Session_ID = 39 :
    Message_Type = 1,  Negotiated_Version = 1,  Message_MTU = 32
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 39
Session_ID = 39 :
Negotiated_Cap_Entry = 1
```

Finally,= to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2009
------------------------------------------------------------------------
Client_ID = 2009,  Entity_ID = 1 :
   Message_Type = 1,  Version_Range = 1 ~ 1
         Message_Ver = 1,    Message_Mtu = 32
```

Verifying the ISSU Process for an MPLS VRF ("Table ID") Client Example

This example shows how to verify the ISSU process for an MPLS VRF ("Table ID") client.

The first command shows you whether the VRF client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2008
------------------------------------------------------------------------
 Client_ID = 2008,  Entity_ID = 1 :
 *** Session_ID = 19,  Session_Name = TABLEID ISSU CF :
    Peer   Peer  Negotiate  Negotiated   Cap      Msg     Session
  UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
     4       13   PRIMARY   COMPATIBLE    1        1         0
                           (no policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 19
         Nego_Session_Name = TABLEID ISSU CF
         Transport_Mtu = 3948

Router# show issu sessions 2008
------------------------------------------------------------------------
 Client_ID = 2008,  Entity_ID = 1 :
 *** Session_ID = 19,  Session_Name = TABLEID ISSU CF :
    Peer   Peer  Negotiate  Negotiated   Cap      Msg     Session
  UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
     4       13   PRIMARY   COMPATIBLE    1        1         0
                           (no policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 19
         Nego_Session_Name = TABLEID ISSU CF
         Transport_Mtu = 3948
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, in order to see the negotiated message version:

```
Router# show issu negotiated version 19
 Session_ID = 19 :
     Message_Type = 1,  Negotiated_Version = 1,  Message_MTU = 44
     Message_Type = 2,  Negotiated_Version = 1,  Message_MTU = 4
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 19
Session_ID = 19 :
Negotiated_Cap_Entry = 1
```

Finally, to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2008
------------------------------------------------------------------------
 Client_ID = 2008,  Entity_ID = 1 :
    Message_Type = 1,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 44
    Message_Type = 2,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 4
```

Verifying the ISSU Process for an MPLS LSD Label Manager HA Client Example

This example shows how to verify the ISSU process for an MPLS LSD Label Manager HA client.

The first command shows you whether the LSD client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2007
------------------------------------------------------------------------
 Client_ID = 2007,  Entity_ID = 1 :
 *** Session_ID = 40,  Session_Name = lsd_ha :
    Peer    Peer  Negotiate  Negotiated   Cap      Msg     Session
  UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
     4       30   PRIMARY   COMPATIBLE    1        1         0
                              (policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 40
         Nego_Session_Name = lsd_ha
         Transport_Mtu = 3948
         Compat_Result: raw_result = COMPATIBLE,  policy_result = COMPATIBLE
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, in order to see the negotiated message version:

```
Router# show issu negotiated version 40
Session_ID = 40 :
     Message_Type = 1,  Negotiated_Version = 2,  Message_MTU = 8
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 40
------------------------------------------------
  Client_ID = 2007,  Entity_ID = 1,  Session_ID = 40 :
      Negotiated_Cap_Entry = 1
```

Finally, to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2007
------------------------------------------------------------------------
 Client_ID = 2007,  Entity_ID = 1 :
    Message_Type = 1,  Version_Range = 1 ~ 2
          Message_Ver = 1,    Message_Mtu = 12
          Message_Ver = 2,    Message_Mtu = 8
```

Verifying the ISSU Process for an MPLS MFI Pull Client Example

This example shows how to verify the ISSU process for an MPLS MFI Pull client.

The first command shows you whether the MFI Pull client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2030
------------------------------------------------------------------------
Client_ID = 2030,  Entity_ID = 1 :
*** Session_ID = 131073,  Session_Name = MFI Pull         (6):
    Peer   Peer  Negotiate  Negotiated   Cap      Msg     Session
  UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
     7      35   PRIMARY    COMPATIBLE    1        1         0
                             (no policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 131073
         Nego_Session_Name = MFI Pull                       (6)
         Transport_Mtu = 4056
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, in order to see the negotiated message version:

```
Router# show issu negotiated version 131073
 Session_ID = 131073:
     Message_Type = 1006,  Negotiated_Version = 1,  Message_MTU = 4
 Message_Type = 3003,  Negotiated_Version = 1,  Message_MTU = 12
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 131073
 Session_ID = 131073 :
      Negotiated_Cap_Entry = 1
```

Finally to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2030
------------------------------------------------------------------------
 Client_ID = 2030,  Entity_ID = 1 :
 Message_Type = 1006,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 4
 Message_Type = 2004,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 12
```

Verifying the ISSU Process for an MPLS MFI Push Client Example

This example shows how to verify the ISSU process for an MPLS MFI Push client.

The first command shows you whether the MFI Push client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2031
------------------------------------------------------------------------
Client_ID = 2031,  Entity_ID = 1 :
*** Session_ID = 196646,  Session_Name = MFI Push         (6):
    Peer   Peer  Negotiate  Negotiated   Cap      Msg     Session
```

```
  UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
     7      36   PRIMARY    COMPATIBLE    1        1        0
                            (no policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 196646
         Nego_Session_Name = MFI Push                      (6)
         Transport_Mtu = 4056
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, in order to see the negotiated message version:

```
Router# show issu negotiated version 196646
Session_ID = 196646:
     Message_Type = 101,  Negotiated_Version = 1,  Message_MTU = 17
     Message_Type = 105,  Negotiated_Version = 1,  Message_MTU = 31
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 196646
Session_ID = 196646 :
      Negotiated_Cap_Entry = 1
```

Finally to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2031
------------------------------------------------------------------------
Client_ID = 2031,  Entity_ID = 1 :
        Message_Type = 5002,  Version_Range = 1 ~ 2
              Message_Ver = 1,    Message_Mtu = 10
        Message_Type = 5018,  Version_Range = 1 ~ 1
              Message_Ver = 1,    Message_Mtu = 39
```

Verifying the ISSU Process for an MPLS LSPV Push Client Example

This example shows how to verify the ISSU process for an MPLS LSVP Push client.

The first command shows you whether the LSPV Push client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2089
------------------------------------------------------------------------
Client_ID = 2089,  Entity_ID = 1 :
 *** Session_ID = 45,  Session_Name = MPLS LSPV Push (6 ):
    Peer    Peer  Negotiate  Negotiated   Cap      Msg     Session
  UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
      7     36   PRIMARY    COMPATIBLE    1        1         0
                           (no policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 45
         Nego_Session_Name = MPLS LSPV Push (6 )
         Transport_Mtu = 1438
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, in order to see the negotiated message version:

```
Router# show issu negotiated version 45
 Session_ID = 45:
```

```
 Message_Type = 0,  Negotiated_Version = 1,  Message_MTU = 74
 Message_Type = 1,  Negotiated_Version = 1,  Message_MTU = 120
 Message_Type = 2,  Negotiated_Version = 1,  Message_MTU = 120
 Message_Type = 3,  Negotiated_Version = 1,  Message_MTU = 5122
 Message_Type = 4,  Negotiated_Version = 1,  Message_MTU = 6
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 45
Session_ID = 45:
Cap_Type = 0    Cap_Result = 1     No cap value assigned
```

Finally to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2089
------------------------------------------------------------------------
 Client_ID = 2089,  Entity_ID = 1 :
    Message_Type = 0,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 74
    Message_Type = 1,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 120
 Message_Type = 2,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 120
    Message_Type = 3,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 5122
 Message_Type = 4,  Version_Range = 1 ~ 1
          Message_Ver = 1,    Message_Mtu = 6
```

Verifying the ISSU Process for an MPLS TE Client Example

This example shows how to verify the ISSU process for an MPLS TE client.

The first command shows you whether the TE client's old and new software versions are compatible, and therefore are able to make use of the ISSU opportunity:

```
Router# show issu sessions 2053
------------------------------------------------------------------------
 Client_ID = 2053,  Entity_ID = 1 :
 *** Session_ID = 84,  Session_Name = RSVP HA Session :
    Peer    Peer  Negotiate  Negotiated   Cap      Msg     Session
  UniqueID  Sid    Role       Result   GroupID  GroupID  Signature
    22       94   PRIMARY   COMPATIBLE    1        1         0
                          (no policy)
    Negotiation Session Info for This Message Session:
         Nego_Session_ID = 84
         Nego_Session_Name = RSVP HA Session
         Transport_Mtu = 1392
```

Now you can take the session ID displayed in the previous command's output and enter it into the next command, in order to see the negotiated message version:

```
Router# show issu negotiated version 84
Session_ID = 84 :
     Message_Type = 1,  Negotiated_Version = 2,  Message_MTU = 1024
```

Next you can enter the same session ID into the following command to display the capability negotiation result:

```
Router# show issu negotiated capability 84
Session_ID = 84 :
     Cap_Type = 0,     Cap_Result = 1     No cap value assigned
```

Finally to see which message types and versions are supported by this particular client, you enter the client ID into the following command:

```
Router# show issu message types 2053
------------------------------------------------------------------------
 Client_ID = 2053,  Entity_ID = 1 :
    Message_Type = 1,  Version_Range = 1 ~ 2
          Message_Ver = 1,    Message_Mtu = 1024
          Message_Ver = 2,    Message_Mtu = 1024
```

Additional References

The following sections provide references related to the ISSU MPLS Clients feature.

Related Documents

Related Topic	Document Title
ISSU process	• Cisco IOS XE In Service Software Upgrade Process • Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide
High availability commands	*Cisco IOS High Availability Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for ISSU MPLS Clients

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 82: Feature Information for ISSU MPLS Clients

Feature Name	Releases	Feature Information
ISSU MPLS--LDP	Cisco IOS XE Release 2.1	This feature allows In Service Software Upgrade (ISSU) support for the Label Distribution Protocol (LDP) and Multiprotocol Label Switching (MPLS) Forwarding. MPLS applications can be upgraded using the In Service Software Upgrade (ISSU) process. Thus, MPLS applications are considered ISSU's MPLS clients. The ISSU process allows Cisco IOS XE software to be updated or otherwise modified while packet forwarding continues. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.
		The following commands were introduced or modified: **show issu clients, show issu entities, show issu message types, show issu negotiated, show issu outage, show issu sessions.**

Feature Name	Releases	Feature Information
ISSU--MPLS VPN (Support for IPv4 VPNs)	Cisco IOS XE Release 2.1	This feature supports In Service Software Upgrade (ISSU) for Multiprotocol Label Switching (MPLS) Virtual Private networks (VPNs) for IPv4 address families only. In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. No commands were introduced or modified for this feature.
ISSU--MPLS TE	Cisco IOS XE Release 2.3	This feature allows upgrade or downgrade of compatible Cisco IOS XE software images on the back up Route Processor (RP) while the device is operational and passing traffic on Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels. In Cisco IOS XE Release 2.3, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. No commands were introduced or modified for this feature.

Glossary

IS--intermediate system.

ISSU--In Service Software Upgrade.

LACP--Link Aggregration Control Protocol.

LDP--Label Distribution Protocol.

MFI--Multiprotocol Label Switching Forwarding Infrastructure.

MPLS--Multiprotocol Label Switching.

OAM--Operation, Administration, and Management.

PagP--port aggregation Protocol.

PPP--Point to Point protocol.

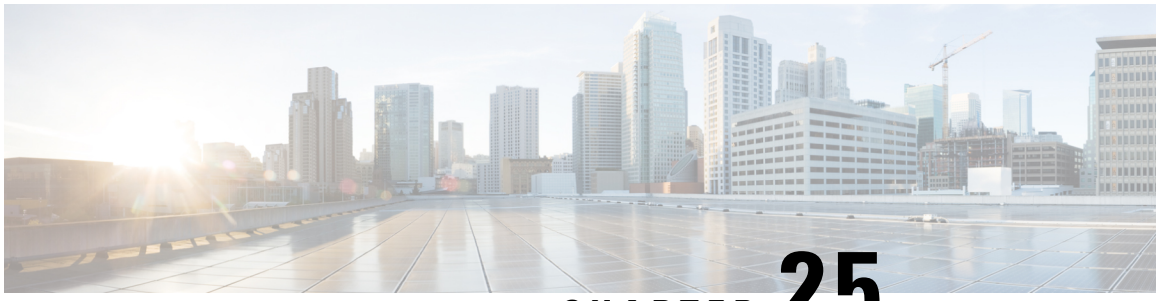
RP--Route Processor.

RSVP GR--Resource Reservation Protocol graceful restart.

TE--traffic engineering.

VPN--Virtual Private Network.

VRF--virtual routing and forwarding.



CHAPTER 25

MPLS Traffic Engineering--RSVP Graceful Restart

The MPLS Traffic Engineering--RSVP Graceful Restart feature allows a neighboring Route Processor (RP) to recover from disruption in control plane service (specifically, the Label Distribution Protocol [LDP] component) without losing its Multiprotocol Label Switching (MPLS) forwarding state.

Prerequisites for MPLS TE--RSVP Graceful Restart

Perform the following tasks on routers before configuring the MPLS Traffic Engineering--RSVP Graceful Restart feature:

- Configure the Resource Reservation Protocol (RSVP).
- Enable MPLS.
- Configure traffic engineering (TE).
- Enable graceful restart.

If you have many tunnels/LSPs (100 or more) or if you have a large-scale network, the following configuration is recommended:

```
ip rsvp signalling refresh reduction
ip rsvp signalling rate-limit period 50 burst 16 maxsize 3000 limit 37
ip rsvp signalling patherr state-removal
ip rsvp signalling initial-retransmit-delay 15000
```

Additional info about these RSVP commands can be found in the *Cisco IOS Quality of Service Command Reference*.

Restrictions for MPLS TE--RSVP Graceful Restart

- Graceful restart supports node failure only.
- Graceful restart does not support restart or recovery on Cisco nodes, but helps in recovering a neighbor that is restart capable. Cisco routers advertise a restart time of 5 milliseconds (ms) and a recovery time of 0 in hello messages.
- Unnumbered interfaces are not supported.

Information About MPLS TE--RSVP Graceful Restart

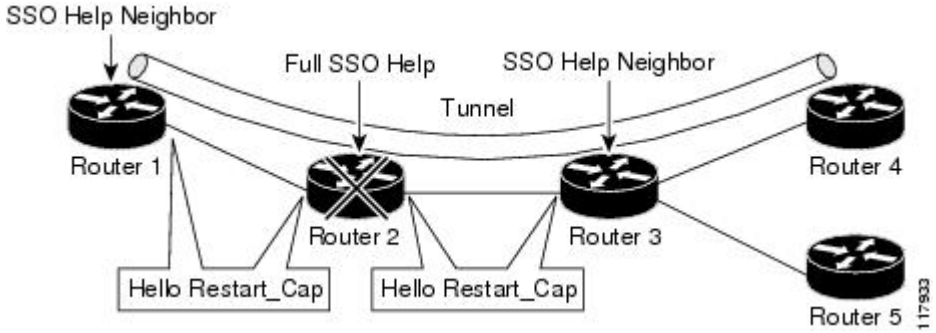
Graceful Restart

Graceful restart allows RSVP TE enabled nodes to start gracefully following a node failure in the network such that the RSVP state after the failure is restored as quickly as possible. The node failure may be completely transparent to other nodes in the network as far as the RSVP state is concerned.

Graceful restart preserves the label values and forwarding information and works with third-party or Cisco routers seamlessly.

Graceful restart depends on RSVP hello messages that include Hello Request or Hello Acknowledgment (ACK) objects between two neighbors.

The figure below shows the graceful restart extension to these messages that an object called Restart_Cap, which tells neighbors that a node, may be capable of restarting if a failure occurs. The time-to-live (TTL) in these messages is set to 255 so that adjacencies can be maintained through alternate paths even if the link between two neighbors goes down.



The Restart_Cap object has two values--the restart time, which is the sender's time to restart the RSVP_TE component and exchange hello messages after a failure; and the recovery time, which is the desired time that the sender wants the receiver to synchronize the RSVP and MPLS databases.

In the figure above, graceful restart is enabled on Router 1, Router 2, Router 3, and Router 4. For simplicity, assume that all routers are restart capable. A TE label switched path (LSP) is signaled from Router 1 to Router 4.

Router 2 and Router 3 exchange periodic graceful restart hello messages every 10,000 ms (10 seconds), and so do Router 2 and Router 1 and Router 3 and Router 4. Assume that Router 2 advertises its restart time as 60,000 ms (60 seconds) and its recovery time as 60,000 ms (60 seconds) as shown in the following example:

```
23:33:36: Outgoing Hello:
23:33:36:   version:1 flags:0000 cksum:883C ttl:255 reserved:0 length:32
23:33:36:   HELLO               type HELLO REQUEST length 12:
23:33:36:   Src_Instance: 0x6EDA8BD7, Dst_Instance: 0x00000000
23:33:36:   RESTART_CAP         type 1 length 12:
23:33:36:   Restart_Time: 0x0000EA60
, Recovery_Time: 0x0000EA60
```

Note The restart and recovery time are shown in **bold** in the last entry.

Router 3 records this into its database. Also, both neighbors maintain the neighbor status as UP. However, Router 3's control plane fails at some point (for example, a Primary Route Processor failure). As a result, RSVP and TE lose their signaling information and states although data packets continue to be forwarded by the line cards.

When four ACK messages are missed from Router 2 (40 seconds), Router 3 declares communication with Router 2 lost "indicated by LOST" and starts the restart time to wait for the duration advertised in Router 2's restart time previously and recorded (60 seconds). Router 1 and Router 2 suppress all RSVP messages to Router 3 except hellos. Router 3 keeps sending the RSVP Path and Resv refresh messages to Router 4 and Router 5 so that they do not expire the state for the LSP; however, Router 3 suppresses these messages for Router 2.

Note A node restarts if it misses four ACKs or its hello src_instance (last source instance sent to its neighbor) changes so that its restart time = 0.

Before the restart time expires, Router 2 restarts and loads its configuration and graceful restart makes the configuration of router 2 send the hello messages with a new source instance to all the data links attached. However, because Router 2 has lost the neighbor states, it does not know what destination instance it should use in those messages; therefore, all destination instances are set to 0.

When Router 3 sees the hello from Router 2, Router 3 stops the restart time for Router 2 and sends an ACK message back. When Router 3 sees a new source instance value in Router 2's hello message, Router 3 knows that Router 2 had a control plane failure. Router 2 gets Router 3's source instance value and uses it as the destination instance going forward.

Router 3 also checks the recovery time value in the hello message from Router 2. If the recovery time is 0, Router 3 knows that Router 2 was not able to preserve its forwarding information and Router 3 deletes all RSVP state that it had with Router 2.

If the recovery time is greater than 0, Router 1 sends Router 2 Path messages for each LSP that it had previously sent through Router 2. If these messages were previously refreshed in summary messages, they are sent individually during the recovery time. Each of these Path messages includes a Recovery_Label object containing the label value received from Router 2 before the failure.

When Router 3 receives a Path message from Router 2, Router 3 sends a Resv message upstream. However, Router 3 suppresses the Resv message until it receives a Path message.

Graceful Restart Benefits

- Graceful restart allows a node to recover state information from its neighbor when there is an RP failure or the device has undergone a stateful switchover (SSO).
- Graceful restart allows session information recovery with minimal disruption to the network.
- A node can perform a graceful restart to help a neighbor recover its state by keeping the label bindings and state information to provide a quick recovery of the failed node and not affect the traffic that is currently forwarded.

How to Configure MPLS TE--RSVP Graceful Restart

Enabling Graceful Restart

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart mode help-neighbor**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart mode help-neighbor** **Example:** `Router(config)# ip rsvp signalling hello graceful-restart mode help-neighbor`	Sets the number of DSCP hello messages on a neighboring router with restart capability.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

What to do next

Note

If you have many tunnels/LSPs (100 or more) or if you have a large-scale network, the following configuration is recommended:

```
ip rsvp signalling refresh reduction
ip rsvp signalling rate-limit period 50 burst 16 maxsize 3000 limit 37
ip rsvp signalling patherr state-removal
ip rsvp signalling initial-retransmit-delay 15000
```

Additional info about these RSVP commands can be found in the Cisco IOS Quality of Service Command Reference.

Setting a DSCP Value on a Router for MPLS TE Graceful Restart

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart dscp** *num*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart dscp** *num* **Example:** `Router(config)# ip rsvp signalling hello graceful-restart dscp 30`	Sets the number of DSCP hello messages on a graceful restart-enabled router.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Setting a Hello Refresh Interval for MPLS TE Graceful Restart

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart refresh interval** *interval-value*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart refresh interval** *interval-value* **Example:** `Router(config)# ip rsvp signalling hello graceful-restart refresh interval 5000`	Sets a hello refresh interval on a router with graceful restart enabled.
Step 4	**exit** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Setting a Missed Refresh Limit for MPLS TE Graceful Restart

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart refresh misses** *msg-count*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart refresh misses** *msg-count* **Example:** `Router(config)# ip rsvp signalling hello graceful-restart refresh misses 5`	Sets a refresh limit on a router with graceful restart enabled.
Step 4	**exit** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Verifying Graceful Restart Configuration

SUMMARY STEPS

1. **enable**
2. **show ip rsvp hello graceful-restart**
3. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip rsvp hello graceful-restart**

Use this command to display information about the status of graceful restart and related parameters. For example:

Example:

```
Router# show ip rsvp hello graceful-restart
Graceful Restart:Enabled (help-neighbor only)
  Refresh interval:10000 msecs
  Refresh misses:4
  DSCP:0x30
  Advertised restart time:0 secs
  Advertised recovery time:0 secs
  Maximum wait for recovery:3600000 secs
```

Step 3 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS TE--RSVP Graceful Restart

Example MPLS TE--RSVP Graceful Restart

In the following example, graceful restart is enabled, and related parameters, including a DSCP value, a refresh interval, and a missed refresh limit are set:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp signalling hello graceful-restart mode help-neighbor
Router(config)# ip rsvp signalling hello graceful-restart dscp 30
Router(config)# ip rsvp signalling hello graceful-restart refresh interval 10000
Router(config)# ip rsvp signalling hello graceful-restart refresh misses 4
Router(config)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Quality of service (QoS) features including signaling, classification, and congestion management	*Cisco IOS XE Quality of Service Solutions Configuration Guide*, Release 2
Stateful switchover	Stateful Switchover

Related Topic	Document Title
MPLS Label Distribution Protocol	MPLS Label Distribution Protocol (LDP)
Cisco nonstop forwarding	Cisco Nonstop Forwarding
Information on stateful switchover, Cisco nonstop forwarding, graceful restart	MPLS LDP: SSO/NSF Support and Graceful Restart
Hellos for state timeout	MPLS TE--RSVP Hello State Timer

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBS are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3209	RSVP-TE: Extensions to RSVP for LSP Tunnels
RFC 3473	*Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions*
RFC 3478	Graceful Restart Mechanism for Label Distribution

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering--RSVP Graceful Restart

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 83: Feature Information for MPLS Traffic Engineering--RSVP Graceful Restart

Feature Name	Releases	Feature Information
MPLS Traffic Engineering--RSVP Graceful Restart	Cisco IOS XE Release 2.3	The MPLS TE--RSVP Graceful Restart feature allows a neighboring Route Processor (RP) to recover from disruption in control plane service (specifically, the Label Distribution Protocol (LDP) component) without losing its MPLS forwarding state. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **ip rsvp signalling hello graceful-restart dscp**, **ip rsvp signalling hello graceful-restart mode help-neighbor**, **ip rsvp signalling hello graceful-restart refresh interval**, **ip rsvp signalling hello graceful-restart refresh misses**, **show ip rsvp counters**, **show ip rsvp counters state teardown**, **show ip rsvp hello**, **show ip rsvp hello client lsp detail**, **show ip rsvp hello client lsp summary**, **show ip rsvp hello client neighbor detail**, **show ip rsvp hello client neighbor summary**, **show ip rsvp hello graceful-restart**, **show ip rsvp hello instance detail**, **show ip rsvp hello instance summary**.

Glossary

autonomous system --A collection of networks that share the same routing protocol and that are under the same system administration.

ASBR --Autonomous System Boundary Router. A router that connects and exchanges information between two or more autonomous systems.

backup tunnel --A Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnel used to protect other (primary) tunnels' traffic when a link or node failure occurs.

DSCP --differentiated services code point. Six bits in the IP header, as defined by the Internet Engineering Task Force (IETF). These bits determine the class of service provided to the IP packet.

Fast Reroute --A mechanism for protecting Multiprotocol Label Switching (MPLS) traffic engineering (TE) label switched paths (LSPs) from link and node failure by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish end-to-end LSPs

to replace them. Fast Reroute (FRR) locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

graceful restart --A process for helping a neighboring Route Processor (RP) restart after a node failure has occurred.

headend --The router that originates and maintains a given label switched path (LSP). This is the first router in the LSP's path.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

instance --A mechanism that implements the Resource Reservation Protocol. (RSVP) hello extensions for a given router interface address and remote IP address. Active hello instances periodically send Hello Request messages, expecting Hello ACK messages in response. If the expected ACK message is not received, the active hello instance declares that the neighbor (remote IP address) is unreachable (that is, it is lost). This can cause label switched paths (LSPs) crossing this neighbor to be fast rerouted.

label --A short, fixed-length data identifier that tells switching nodes how to forward data (packets or cells).

LDP --Label Distribution Protocol. The protocol that supports Multiprotocol Label Switching (MPLS) hop-by-hop forwarding by distributing bindings between labels and network prefixes.

LSP --label switched path. A configured connection between two routers, in which Multiprotocol Label Switching (MPLS) is used to carry packets. A path created by the concatenation of one or more label switched hops, allowing a packet to be forwarded by swapping labels from an MPLS node to another MPLS node.

merge point --The tail of the backup tunnel.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. MPLS enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels.

PLR --point of local repair. The headend of the backup tunnel.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

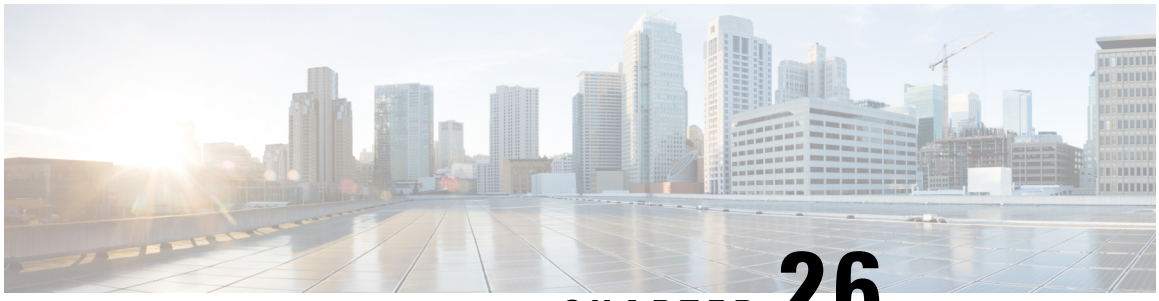
state --Information that a router must maintain about each label switched path (LSP). The information is used for rerouting tunnels.

tailend --The router upon which an label switched path (LSP) is terminated. This is the last router in the LSP's path.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

topology --The physical arrangement of network nodes and media within an enterprise networking structure.

tunnel --Secure communications path between two peers, such as two routers.



CHAPTER 26

NSF SSO--MPLS TE and RSVP Graceful Restart

The NSF/SSO--MPLS TE and RSVP Graceful Restart feature allows a Route Processor (RP) to recover from disruption in control plane service without losing its Multiprotocol Label Switching (MPLS) forwarding state.

Cisco nonstop forwarding (NSF) with stateful switchover (SSO) provides continuous packet forwarding, even during a network processor hardware or software failure. In a redundant system, the secondary processor recovers control plane service during a critical failure in the primary processor. SSO synchronizes the network state information between the primary and the secondary processor.

Prerequisites for NSF SSO--MPLS TE and RSVP Graceful Restart

- Configure Resource Reservation Protocol (RSVP) graceful restart in full mode.
- Configure RSVP graceful restart on all interfaces of the neighbor that you want to be restart-capable.
- Configure the redundancy mode as SSO. See the Stateful Switchover feature module for more information.
- Enable NSF on the routing protocols running among the provider routers (P), provider edge (PE) routers, and customer edge (CE) routers. The routing protocols are as follows:
  - Border Gateway Protocol (BGP)
  - Open Shortest Path First (OSPF)
  - Intermediate System-to-Intermediate System (IS-IS)

See the Cisco Nonstop Forwarding feature module for more information.

- Enable MPLS.
- Configure traffic engineering (TE).

Restrictions for NSF SSO--MPLS TE and RSVP Graceful Restart

- RSVP graceful restart supports node failure only.
- Unnumbered interfaces are not supported.
- You cannot enable RSVP fast reroute (FRR) hello messages and RSVP graceful restart on the same router.
- You cannot enable primary one-hop autotunnels, backup autotunnels, or autotunnel mesh groups on a router that is also configured with SSO and Route Processor Redundancy Plus (RPR+). This restriction does not prevent an MPLS TE tunnel that is automatically configured by TE autotunnel from being successfully recovered if any midpoint router along the label-switched path (LSP) of the router experiences an SSO.
- MPLS TE LSPs that are fast reroutable cannot be successfully recovered if the LSPs are FRR active and the Point of Local Repair (PLR) router experiences an SSO.
- When you configure RSVP graceful restart, you must use the neighbor's interface IP address.

Information About NSF SSO--MPLS TE and RSVP Graceful Restart

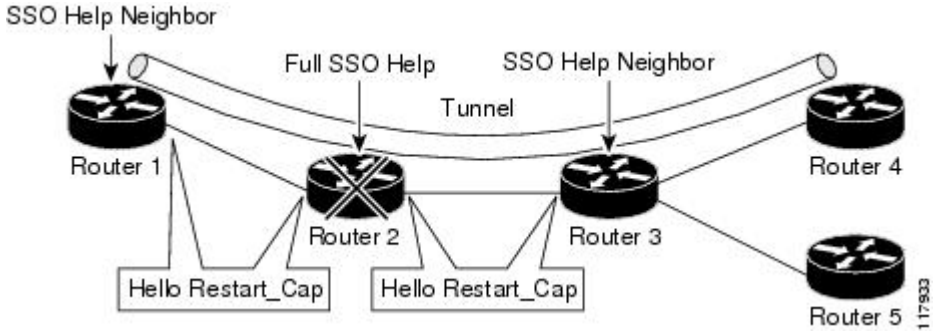
Overview of MPLS TE and RSVP Graceful Restart

RSVP graceful restart allows RSVP TE-enabled nodes to recover gracefully following a node failure in the network such that the RSVP state after the failure is restored as quickly as possible. The node failure may be completely transparent to other nodes in the network.

RSVP graceful restart preserves the label values and forwarding information and works with third-party or Cisco routers seamlessly.

RSVP graceful restart depends on RSVP hello messages to detect that a neighbor went down. Hello messages include Hello Request or Hello Acknowledgment (ACK) objects between two neighbors.

As shown in the figure below, the RSVP graceful restart extension to these messages adds an object called Hello Restart_Cap, which tells neighbors that a node may be capable of recovering if a failure occurs.



The Hello Restart_Cap object has two values: the restart time, which is the sender's time to restart the RSVP_TE component and exchange hello messages after a failure; and the recovery time, which is the desired time that the sender wants the receiver to synchronize the RSVP and MPLS databases.

In the figure above, RSVP graceful restart help neighbor support is enabled on Routers 1 and 3 so that they can help a neighbor recover after a failure, but they cannot perform self recovery. Router 2 has full SSO help support enabled, meaning it can perform self recovery after a failure or help its neighbor to recover. Router 2 has two RPs, one that is active and one that is standby (backup). A TE LSP is signaled from Router 1 to Router 4.

Router 2 performs checkpointing; that is, it copies state information from the active RP to the standby RP, thereby ensuring that the standby RP has the latest information. If an active RP fails, the standby RP can take over.

Routers 2 and 3 exchange periodic graceful restart hello messages every 10,000 milliseconds (ms) (10 seconds), and so do Routers 2 and 1 and Routers 3 and 4. Assume that Router 2 advertises its restart time = 60,000 ms (60 seconds) and its recovery time = 60,000 ms (60 seconds) as shown in the following example:

```
23:33:36: Outgoing Hello:
23:33:36:   version:1 flags:0000 cksum:883C ttl:255 reserved:0 length:32
23:33:36:  HELLO               type HELLO REQUEST length 12:
23:33:36:   Src_Instance: 0x6EDA8BD7, Dst_Instance: 0x00000000
23:33:36:  RESTART_CAP         type 1 length 12:
23:33:36:   Restart_Time: 0x0000EA60, Recovery_Time: 0x0000EA60
```

Router 3 records this into its database. Also, both neighbors maintain the neighbor status as UP. However, Router 3's control plane fails at some point (for example, a primary RP failure). As a result, RSVP and TE lose their signaling information and states although data packets continue to be forwarded by the line cards.

When Router 3 declares communication with Router 2 lost, Router 3 starts the restart time to wait for the duration advertised in Router 2's restart time previously recorded (60 seconds). Routers 1 and 2 suppress all RSVP messages to Router 3 except hellos. Router 3 keeps sending the RSVP PATH and RESV refresh messages to Routers 4 and 5 so that they do not expire the state for the LSP; however, Routers 1 and 3 suppress these messages for Router 2.

When Routers 1 and 3 receive the hello message from Router 2, Routers 1 and 3 check the recovery time value in the message. If the recovery time is 0, Router 3 knows that Router 2 was not able to preserve its forwarding information, and Routers 1 and 3 delete all RSVP state that they had with Router 2.

If the recovery time is greater than 0, Router 1 sends Router 2 PATH messages for each LSP that it had previously sent through Router 2. If these messages were previously refreshed in summary messages, they are sent individually during the recovery time. Each of these PATH messages includes a Recovery_Label object containing the label value received from Router 2 before the failure.

When Router 3 receives a PATH message from Router 2, Router 3 sends a RESV message upstream. However, Router 3 suppresses the RESV message until it receives a PATH message. When Router 2 receives the RESV message, it installs the RSVP state and reprograms the forwarding entry for the LSP.

Benefits of MPLS TE and RSVP Graceful Restart

State Information Recovery

RSVP graceful restart allows a node to perform self recovery or to help its neighbor recover state information when there is an RP failure or the device has undergone an SSO.

Session Information Recovery

RSVP graceful restart allows session information recovery with minimal disruption to the network.

Increased Availability of Network Services

A node can perform a graceful restart to help itself or a neighbor recover its state by keeping the label bindings and state information, thereby providing a faster recovery of the failed node and not affecting currently forwarded traffic.

How to Configure NSF SSO--MPLS TE and RSVP Graceful Restart

Enabling RSVP Graceful Restart Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart mode** (**help-neighbor**| **full**)
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart mode** (**help-neighbor**| **full**) **Example:** `Router(config)# ip rsvp signalling hello graceful-restart mode full`	Enables RSVP TE graceful restart capability on an RP. • Enter the **help-neighbor** keyword to enable a neighboring router to restart after a failure. • Enter the **full**keyword to enable a router to perform self recovery or to help a neighbor recover after a failure.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Returns to privileged EXEC mode.

Enabling RSVP Graceful Restart on an Interface

You must repeat this procedure for each of the neighbor router's interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*]
4. Repeat Step 3 as needed to configure additional interfaces.
5. **ip rsvp signalling hello graceful-restart neighbor** *ip-address*
6. Repeat Step 5 as needed to configure additional IP addresses on a neighbor router's interfaces.
7. **exit**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*] **Example:** `Router(config)# interface POS 1/0/0`	Configures the interface type and number and enters interface configuration mode.
Step 4	Repeat Step 3 as needed to configure additional interfaces.	(Optional) Configures additional interfaces.
Step 5	**ip rsvp signalling hello graceful-restart neighbor** *ip-address* **Example:** `Router(config-if)# ip rsvp signalling hello graceful-restart neighbor 10.0.0.0`	Enables support for RSVP graceful restart on routers helping their neighbors recover TE tunnels following SSO. **Note** The IP address must be that of the neighbor's interface.
Step 6	Repeat Step 5 as needed to configure additional IP addresses on a neighbor router's interfaces.	(Optional) Configures additional IP addresses on a neighbor router's interfaces.
Step 7	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	
Step 8	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Setting a DSCP Value for RSVP Graceful Restart

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart dscp num**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart dscp num** **Example:** `Router(config)# ip rsvp signalling hello graceful-restart dscp 30`	Sets a DSCP value on a router with RSVP graceful restart enabled.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Returns to privileged EXEC mode.

Setting a Value to Control the Refresh Interval for RSVP Hello Messages

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart refresh interval** *interval-value*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart refresh interval** *interval-value* **Example:** `Router(config)# ip rsvp signalling hello graceful-restart refresh interval 5000`	Sets the value to control the request interval in graceful restart hello messages. This interval represents the frequency at which RSVP hello messages are sent to a neighbor; for example, one hello message is sent per each interval. **Note** If you change the default value for this command and you also changed the RSVP refresh interval using the **ip rsvp signalling refresh interval** command, ensure that the value for the **ip rsvp signalling hello graceful-restart refresh interval**command **is less than the value for the ip rsvp signalling hello refresh interval** command. Otherwise, some or all of the label-switched paths (LSPs) may not be recovered after an SSO has occurred.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Returns to privileged EXEC mode.

Setting a Value to Control the Missed Refresh Limit for RSVP Graceful Restart Hello Acknowledgements

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart refresh misses** *msg-count*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart refresh misses** *msg-count* **Example:** `Router(config)# ip rsvp signalling hello graceful-restart refresh misses 5`	Specifies how many sequential RSVP TE graceful restart hello acknowledgments (ACKs) a node can miss before the node considers communication with its neighbor lost. **Note** If you change the default value for this command and you are also using the **ip rsvp signalling hello refresh misses**command, ensure that the value for the **ip rsvp signalling hello graceful-restart refresh misses**command **is less than the value for the ip rsvp signalling hello refresh misses** command. Otherwise, some or all of the LSPs may not be recovered after an SSO has occurred.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Returns to privileged EXEC mode.

Verifying the RSVP Graceful Restart Configuration

SUMMARY STEPS

1. **enable**
2. **show ip rsvp hello graceful-restart**
3. **exit**

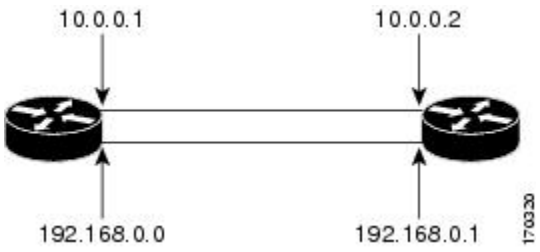
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp hello graceful-restart** **Example:** `Router# show ip rsvp hello graceful-restart`	Displays information about the status of RSVP graceful restart and related parameters.
Step 3	**exit** **Example:** `Router# exit`	(Optional) Returns to user EXEC mode.

Configuration Examples for NSF SSO--MPLS TE and RSVP Graceful Restart

Example Configuring NSF SSO--MPLS TE and RSVP Graceful Restart

In the following example, RSVP graceful restart is enabled globally and on a neighbor router's interfaces as shown in the figure below. Related parameters, including a DSCP value, a refresh interval, and a missed refresh limit are set.




```
enable
```

```
configure terminal
ip rsvp signalling hello graceful-restart mode full
interface POS 1/0/0
 ip rsvp signalling hello graceful-restart neighbor 10.0.0.1
 ip rsvp signalling hello graceful-restart neighbor 10.0.0.2
 exit
ip rsvp signalling hello graceful-restart dscp 30
ip rsvp signalling hello graceful-restart refresh interval 50000
ip rsvp signalling hello graceful-restart refresh misses 5
exit
```

Example Verifying the NSF SSO--MPLS TE and RSVP Graceful Restart Configuration

```
Router# show ip rsvp hello graceful-restart
Graceful Restart: Enabled (full mode)
  Refresh interval: 10000 msecs
  Refresh misses: 4
  DSCP:0x30
  Advertised restart time: 30000 msecs
  Advertised recovery time: 120000 msecs
  Maximum wait for recovery: 3600000 msecs
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Quality of service (QoS) classification	Classification Overview
Stateful switchover	Stateful Switchover
Cisco nonstop forwarding	Information about Cisco Nonstop Forwarding
RSVP hello state timer	MPLS Traffic Engineering: RSVP Hello State Timer

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBS are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3209	*RSVP-TE: Extensions to RSVP for LSP Tunnels*
RFC 3473	*Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions*
RFC 4558	*Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NSF SSO--MPLS TE and RSVP Graceful Restart

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 84: Feature Information for NSF/SSO--MPLS TE and RSVP Graceful Restart

Feature Name	Releases	Feature Information
NSF/SSO--MPLS TE and RSVP Graceful Restart	Cisco IOS XE Release 3.1S Cisco IOS XE Release 3.5S	The NSF/SSO--MPLS TE and RSVP Graceful Restart feature allows a Route Processor (RP) to recover from disruption in control plane service without losing its Multiprotocol Label Switching (MPLS) forwarding state. Cisco nonstop forwarding (NSF) with stateful switchover (SSO) provides continuous packet forwarding, even during a network processor hardware or software failure. In a redundant system, the secondary processor recovers control plane service during a critical failure in the primary processor. SSO synchronizes the network state information between the primary and the secondary processor. In Cisco IOS XE Release 3.1S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.
		The following commands were introduced or modified: **clear ip rsvp high-availability counters**, **debug ip rsvp high-availability**, **debug ip rsvp sso**, **debug mpls traffic-eng ha sso**, **ip rsvp signalling hello graceful-restart dscp**, **ip rsvp signalling hello graceful-restart mode**, **ip rsvp signalling hello graceful-restart mode help-neighbor**, **ip rsvp signalling hello graceful-restart neighbor**, **ip rsvp signalling hello graceful-restart refresh interval**, **ip rsvp signalling hello graceful-restart refresh misses**, **show ip rsvp counters**, **show ip rsvp counters state teardown**, **show ip rsvp hello**, **show ip rsvp hello client lsp detail**, **show ip rsvp hello client lsp summary**, **show ip rsvp hello client neighbor detail**, **show ip rsvp hello client neighbor summary**, **show ip rsvp hello graceful-restart**, **show ip rsvp hello instance detail**, **show ip rsvp hello instance summary**, **show ip rsvp high-availability counters**, **show ip rsvp high-availability database**, **show ip rsvp high-availability summary**.
MPLS TE--RSVP Graceful Restart 12.0S--12.2S Interop	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.
MPLS TE—Autotunnel/Automesh SSO Coexistence	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.

Glossary

DSCP --differentiated services code point. Six bits in the IP header, as defined by the Internet Engineering Task Force (IETF). These bits determine the class of service provided to the IP packet.

Fast Reroute --A mechanism for protecting Multiprotocol Label Switching (MPLS) traffic engineering (TE) label switched paths (LSPs) from link and node failure by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish end-to-end LSPs

to replace them. Fast reroute (FRR) locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

graceful restart --A process for helping a Route Processor (RP) restart after a node failure has occurred.

headend --The router that originates and maintains a given label switched path (LSP). This is the first router in the LSP's path.

hello instance --A mechanism that implements the Resource Reservation Protocol (RSVP) hello extensions for a given router interface address and remote IP address. Active hello instances periodically send hello request messages, expecting Hello ACK messages in response. If the expected ACK message is not received, the active hello instance declares that the neighbor (remote IP address) is unreachable (that is, it is lost). This can cause LSPs crossing this neighbor to be fast rerouted.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

ISSU --In Service Software Upgrade. Software upgrade without service interruption.

label --A short, fixed-length data identifier that tells switching nodes how to forward data (packets or cells).

LSP --label switched path. A configured connection between two routers, in which Multiprotocol Label Switching (MPLS) is used to carry packets.

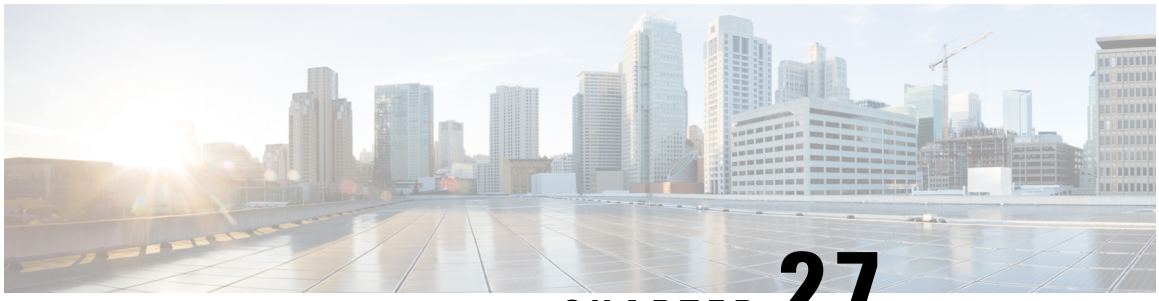
MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. MPLS enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

state --Information that a router must maintain about each label switched path (LSP). The information is used for rerouting tunnels.

tailend --The router upon which a label switched path (LSP) is terminated. This is the last router in the LSP's path.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.



CHAPTER 27

AToM Graceful Restart

The AToM Graceful Restart feature assists neighboring devices that have nonstop forwarding (NSF), stateful switchover (SSO), and graceful restart (GR) for Any Transport over Multiprotocol Label Switching (AToM) to recover gracefully from an interruption in service. AToM GR functions strictly in helper mode, which means it helps other devices that are enabled with the NSF/SSO—Any Transport over MPLS and AToM Graceful Restart feature to recover. If the device with AToM GR fails, its peers cannot help it recover. AToM GR is based on the MPLS Label Distribution Protocol (LDP) Graceful Restart feature.

Keep the following points in mind when reading this document:

- The AToM GR feature described in this document refers to helper mode.
- For brevity, the NSF/SSO—Any Transport over MPLS and AToM Graceful Restart feature is called AToM SSO/NSF in this document.

Prerequisites for AToM Graceful Restart

Any Transport over Multiprotocol Label Switching (AToM) must be configured.

Restrictions for AToM Graceful Restart

- Any Transport over Multiprotocol Label Switching (AToM) graceful restart (GR) is supported in strict helper mode.
- MPLS Label Distribution Protocol (LDP) GR cannot be configured on label-controlled ATM (LC-ATM) interfaces.
- On some hardware platforms, Tag Distribution Protocol (TDP) sessions are not supported. Only LDP sessions are supported.

Information About AToM Graceful Restart

How AToM Graceful Restart Works

Any Transport over Multiprotocol Label Switching Graceful Restart (AToM GR) works in strict helper mode, which means it helps a neighboring Route Processor (RP) that has AToM nonstop forwarding (NSF) and stateful switchover (SSO) to recover from a disruption in service without losing its MPLS forwarding state. The disruption in service could result from a TCP or User Datagram Protocol (UDP) event or the SSO of an RP. AToM GR is based on the MPLS Label Distribution Protocol (LDP) Graceful Restart feature, which preserves forwarding information for AToM circuits during an LDP session interruption. When the neighboring device establishes a new session, the LDP bindings and MPLS forwarding state are recovered.

How to Configure AToM Graceful Restart

Configuring AToM Graceful Restart

There is no Any Transport over Multiprotocol Label Switching (AToM)-specific configuration for AToM Graceful Restart (GR). You enable the Label Distribution Protocol (LDP) GR to assist a neighboring device configured with AToM nonstop forwarding (NSF) and stateful switchover (SSO) to maintain its forwarding state while the LDP session is disrupted.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **mpls ldp graceful-restart**
5. **exit**
6. **show mpls l2transport vc detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip cef distributed** **Example:** `Device(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding.
Step 4	**mpls ldp graceful-restart** **Example:** `Device(config)# mpls ldp graceful-restart`	Enables the device to protect the LDP bindings and MPLS forwarding state during a disruption in service. • AToM GR is enabled globally. When you enable AToM GR, it has no effect on existing LDP sessions. New LDP sessions that are established can perform AToM GR.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.
Step 6	**show mpls l2transport vc detail** **Example:** `Device# show mpls l2transport vc detail`	Displays detailed information about AToM virtual circuits (VCs).

Configuration Examples for AToM Graceful Restart

Example: Configuring AToM Graceful Restart

The following example shows a Fast Ethernet VLAN over Multiprotocol Label Switching (MPLS) configuration. PE1 is configured with Any Transport over MPLS Graceful Restart (AToM GR). PE2 is configured with AToM nonstop forwarding (NSF) and stateful switchover (SSO). The commands for configuring AToM GR and NSF/SSO are shown in bold.

PE1 with AToM GR	PE2 with AToM NSF/SSO
ip cef distributed ! mpls label protocol ldp **mpls ldp graceful-restart** mpls ldp router-id Loopback0 ! pseudowire-class atom encapsulation mpls ! interface Loopback0 ip address 10.1.1.2 255.255.255.255 ! interface FastEthernet2/1/1 no ip address ! interface FastEthernet2/1/1.2 description "xconnect to PE2" encapsulation dot1Q 2 native xconnect 10.2.2.2 1002 pw-class mpls ! ! IGP for MPLS router ospf 10 log-adjacency-changes auto-cost reference-bandwidth 1000 network 10.1.1.2 10.0.0.0 area 0 network 10.1.1.0 10.0.0.255 area 0	**redundancy** **mode sso** ip cef distributed ! mpls label protocol ldp **mpls ldp graceful-restart** mpls ldp router-id Loopback0 ! pseudowire-class atom encapsulation mpls ! interface Loopback0 ip address 10.2.2.2 255.255.255.255 ! interface FastEthernet0/3/2 no ip address ! interface FastEthernet0/3/2.2 description "xconnect to PE1" encapsulation dot1Q 2 xconnect 10.1.1.2 1002 pw-class mpls ! ! IGP for MPLS router ospf 10 log-adjacency-changes **nsf cisco enforce global** auto-cost reference-bandwidth 1000 network 10.2.2.2 10.0.0.0 area 0 network 10.1.1.0 10.0.0.255 area 0

Examples: Verifying AToM Graceful Restart Recovery from an LDP Session Disruption

The following examples show the output of the **show mpls l2transport vc** command during normal operation and when a Label Distribution Protocol (LDP) session is recovering from a disruption.

The following example shows the status of the virtual circuit (VC) on PE1 with Any Transport over Multiprotocol Label Switching Graceful Restart (AToM GR) during normal operation:

```
Device# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Fa2/1/1.2      Eth VLAN 2           10.2.2.2         1002       UP
```

The following example shows the status of the VC on PE1 with AToM GR while the VC is recovering from an LDP session disruption. The forwarding state for the circuit remains as it was before the disruption.

```
Device# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Fa2/1/1.2      Eth VLAN 2           10.2.2.2         1002       RECOVERING
```

The following example shows the status of the VC on PE1 with AToM GR after the LDP session disruption was cleared. The AToM label bindings were advertised within the allotted time and the status returned to UP.

```
Device# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Fa2/1/1.2      Eth VLAN 2           10.2.2.2         1002       UP
```

The following example shows the detailed status of the VC on PE1 with AToM GR during normal operation:

```
Device# show mpls l2transport vc detail

Local interface: Fa2/1/1.2 up, line protocol up, Eth VLAN 2 up
  Destination address: 10.2.2.2, VC ID: 1002, VC status: up
    Preferred path: not configured
    Default path: active
    Tunnel label: imp-null, next hop point2point
    Output interface: Se2/0/2, imposed label stack {16}
  Create time: 1d00h, last status change time: 1d00h
  Signaling protocol: LDP, peer 10.2.2.2:0 up
    MPLS VC labels: local 21, remote 16
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description: "xconnect to PE2"
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 3466, send 12286
    byte totals:   receive 4322368, send 5040220
    packet drops:  receive 0, send 0
```

The following example shows the detailed status of the VC on PE1 with AToM GR while the VC is recovering.

```
Device# show mpls l2transport vc detail

Local interface: Fa2/1/1.2 up, line protocol up, Eth VLAN 2 up
  Destination address: 10.2.2.2, VC ID: 1002, VC status: recovering
    Preferred path: not configured
    Default path: active
    Tunnel label: imp-null, next hop point2point
    Output interface: Se2/0/2, imposed label stack {16}
  Create time: 1d00h, last status change time: 00:00:03
  Signaling protocol: LDP, peer 10.2.2.2:0 down
    MPLS VC labels: local 21, remote 16
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description: "xconnect to PE2"
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 20040, send 28879
    byte totals:   receive 25073016, send 25992388
    packet drops:  receive 0, send 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS AToM and LDP commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS LDP graceful restart	"MPLS LDP Graceful Restart" module in the *MPLS: High Availability Configuration Guide* (part of the *Multiprotocol Label Switching Configuration Guide Library*)
Configuring AToM	"Any Transport over MPLS" module in the *MPLS: Layer 2 VPNs Configuration Guide* (part of the *Multiprotocol Label Switching Configuration Guide Library*)
Nonstop forwarding and stateful switchover for AToM	"NSF SSO Any Transport over MPLS and AToM Graceful Restart" module in the *MPLS: High Availability Configuration Guide* (part of the *Multiprotocol Label Switching Configuration Guide Library*)
High availability commands	*Cisco IOS High Availability Command Reference*

MIBs

MIBs	MIBs Link
MPLS Label Distribution Protocol MIB Version 8 Upgrade	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mib

RFCs

RFCs	Title
RFC 3036	*LDP Specification*
RFC 3478	*Graceful Restart Mechanism for Label Distribution*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

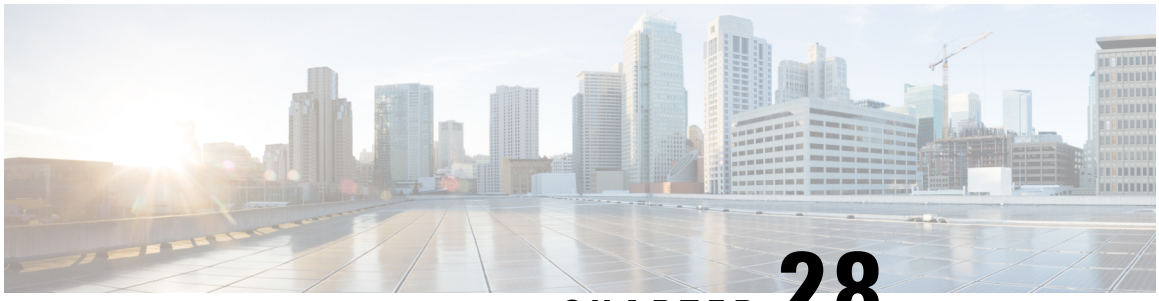
Feature Information for AToM Graceful Restart

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 85: Feature Information for AToM Graceful Restart

Feature Name	Releases	Feature Information
AToM Graceful Restart	12.0(29)S 12.2(33)SRA 12.2(33)SXH 12.4(11)T Cisco IOS XE Release 2.3	The AToM Graceful Restart feature assists neighboring devices that have nonstop forwarding (NSF), stateful switchover (SSO), and graceful restart (GR) for Any Transport over Multiprotocol Label Switching (AToM) to recover gracefully from an interruption in service. AToM GR functions strictly in helper mode, which means it helps other devices that are enabled with the NSF/SSO—Any Transport over MPLS and AToM Graceful Restart feature to recover. If the device with AToM GR fails, its peers cannot help it recover. AToM GR is based on the MPLS Label Distribution Protocol (LDP) Graceful Restart feature. In Cisco IOS Release 12.0(29)S, this feature was introduced. In Cisco IOS Release 12.2(33)SRA, support was added for the Cisco 7600 series routers. In Cisco IOS Release 12.2(33)SXH, this feature was integrated into the release. In Cisco IOS Release 12.4(11)T, this feature was integrated into the release. In Cisco IOS Release XE 2.3, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. This feature uses no new or modified commands.



CHAPTER 28

NSF SSO--Any Transport over MPLS and AToM Graceful Restart

The NSF/SSO--Any Transport over MPLS and AToM Graceful Restart feature allows Any Transport over MPLS (AToM) to use Cisco nonstop forwarding (NSF), stateful switchover (SSO), and Graceful Restart (GR) to allow a Route Processor (RP) to recover from a disruption in control plane service without losing its Multiprotocol Label Switching (MPLS) forwarding state.

NSF with SSO is effective at increasing availability of network services. Cisco NSF with SSO provides continuous packet forwarding, even during a network processor hardware or software failure. In a redundant system, the secondary processor recovers control plane service during a critical failure in the primary processor. SSO synchronizes the network state information between the primary and the secondary processor.

Note In this document, the NSF/SSO--Any Transport over MPLS and AToM Graceful Restart feature is referred to as AToM NSF for brevity.

In Cisco IOS XE software, AToM NSF supports the following attachment circuits:

- ATM
- Ethernet to Ethernet VLAN interworking

Prerequisites for AToM NSF

Before you can configure AToM NSF, make sure the following tasks have been completed:

- AToM virtual circuits (VCs) have been configured on the router. See the Any Transport over MPLS for information on configuring AToM. For configuring L2VPN Interworking, see the L2VPN Interworking feature module.
- SSO has been configured on the RPs. See the Stateful Switchover feature module for configuration information.
- Nonstop forwarding has been configured on the routers. You must enable nonstop forwarding on the routing protocols running between the P routers, PE routers, and CE routers. The routing protocols are the following:
  - Open Shortest Path First (OSPF),
  - Intermediate System-to-Intermediate System (IS-IS), and
  - Border Gateway Protocol (BGP).

See the Cisco Nonstop Forwarding feature module for configuration information.

- AToM NSF requires that neighbor networking devices be able to perform AToM GR.

Restrictions for AToM NSF

- AToM NSF cannot be configured on label-controlled ATM (LC-ATM) interfaces.
- AToM NSF supports AToM Layer 2 Virtual Private Network (L2VPN) Interworking. However, Layer 2 Tunnel Protocol Version 3 (L2TPv3) Interworking is not supported.
- AToM NSF interoperates with Layer 2 local switching. However, AToM NSF has no effect on interfaces configured for local switching.
- To allow distributed Cisco Express Forwarding to work on the interfaces, disable fair queueing on serial interfaces.

Information About AToM NSF

How AToM NSF Works

AToM NSF improves the availability of a service provider's network that uses AToM to provide Layer 2 VPN services to its customers. HA provides the ability to detect failures and handle them with minimal disruption to the service being provided. AToM NSF is achieved by SSO and NSF mechanisms. A standby RP provides control-plane redundancy. The control plane state and data plane provisioning information for the attachment circuits (ACs) and AToM pseudowires (PWs) are checkpointed to the standby RP to provide NSF for AToM L2VPNs.

AToM Information Checkpointing

Checkpointing is a function that copies state information from the active RP to the backup RP, thereby ensuring that the backup RP has the latest information. If the active RP fails, the backup RP can take over.

For the AToM NSF feature, the checkpointing function copies the active RP's information bindings to the backup RP. The active RP sends updates to the backup RP when information is modified.

To display checkpointing data, issue the **show acircuit checkpoint** command on the active and backup RPs. The active and backup RPs have identical copies of the information.

Checkpointing Troubleshooting Tips for AToM NSF

To help troubleshoot checkpointing errors, use the following commands:

- Use the **debug acircuit checkpoint** command to enable checkpointing debug messages for ACs.
- Use the **debug mpls l2transport checkpoint** command to enable checkpointing debug messages for AToM.
- Use the **show acircuit checkpoint** command to display the AC checkpoint information.
- Use the **show mpls l2transport checkpoint** command to display whether checkpointing is allowed, how many AToM VCs were bulk-synchronized (on the active RP), and how many AToM VCs have checkpoint data (on the standby RP).
- Use the **show mpls l2transport vc detail** command to display details of VC checkpointed information.

NSF SSO Support for Ethernet to Ethernet VLAN Interworking

The NSF/SS0--Ethernet to Ethernet VLAN Interworking features enables SSO and NSF capabilities for Ethernet to VLAN attachment circuits. Changes in the learned MAC address for interworking are reflected on the standby RP so that identical values exist on the active and standby RPs.

ISSU Support for AToM NSF

AToM NSF supports In Service Software Upgrade (ISSU) capability. Virtual Private LAN Services (VPLS) NSF/SSO and HA with ISSU work together to enable upgrades or downgrades of a Cisco IOS XE image without control and data plane outages. With ISSU, all message data structures that are used for checkpointing and exchanges between the active RP and standby RP are versioned.

How to Configure AToM NSF

There is no AToM-specific configuration for AToM NSF. Before you configure AToM NSF, you need to configure MPLS LDP Graceful Restart. You enable MPLS LDP Graceful Restart to assist a neighboring router configured with AToM NSF to maintain its forwarding state while the LDP session is disrupted. See the LDP Graceful Restart document for information about how MPLS LDP Graceful Restart works and how you can customize it for your network.

MPLS LDP Graceful Restart is enabled globally. When you enable MPLS LDP Graceful Restart, it has no effect on existing LDP sessions. MPLS LDP Graceful Restart is enabled for new sessions that are established after the feature has been globally enabled.

This section contains the following task:

Configuring MPLS LDP Graceful Restart

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **mpls ldp graceful-restart**
5. **interface** *type slot / subslot / port* [**.** *subinterface-number*]
6. **mpls ip**
7. **mpls label protocol ldp**
8. **exit**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding. **Note** In Cisco ASR 1000 Series Aggregation Services Routers, the **distributed** keyword is mandatory.
Step 4	**mpls ldp graceful-restart** **Example:** `Router (config)# mpls ldp graceful-restart`	Enables the router to protect the LDP bindings and MPLS forwarding state during a disruption in service.
Step 5	**interface** *type slot / subslot / port* [**.** *subinterface-number*] **Example:** `Router(config)# interface pos 0/3/0`	Specifies an interface and enters interface configuration mode.
Step 6	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Configures MPLS hop-by-hop forwarding for an interface.

	Command or Action	Purpose
Step 7	**mpls label protocol ldp** **Example:** `Router(config-if)# mpls label protocol ldp`	Configures the use of LDP for an interface. • You can also issue the **mpls label protocol ldp** command in global configuration mode, which enables LDP on all interfaces configured for MPLS.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for AToM NSF

Example Ethernet to VLAN Interworking with AToM NSF

The following example shows how to configure AToM NSF on two PE routers:

PE1

```
ip cef distributed
!
redundancy
mode sso
!
boot system flash disk2:rsp-pv-mz
!
mpls ldp graceful-restart
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp advertise-labels
!
pseudowire-class atom-eth
 encapsulation mpls
 interworking ethernet
!
interface Loopback0
 ip address 10.8.8.8 255.255.255.255
!
interface FastEthernet1/1/0
 xconnect 10.9.9.9 123 encap mpls pw-class atom-eth
interface POS0/1/0
 ip address 10.1.1.1 255.255.255.0
 mpls ip
 mpls label protocol ldp
 clock source internal
```

```
 crc 32
!
interface Loopback0
 ip address 10.8.8.8 255.255.255.255
 no shutdown
!
router ospf 10
 nsf
 network 10.8.8.8 0.0.0.0 area 0
 network 10.19.1.1 0.0.0.0 area 0
```

PE2

```
ip cef distributed
!
redundancy
mode sso
!
boot system flash disk2:rsp-pv-mz
mpls ldp graceful-restart
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp advertise-labels
!
pseudowire-class atom-eth
 encapsulation mpls
 interworking eth
!
interface Loopback0
 ip address 10.9.9.9 255.255.255.255
!
interface FastEthernet0/3/0
 ip route-cache cef
!
interface FastEthernet0/3/0.3
 encapsulation dot1Q 10
 xconnect 10.8.8.8 123 encap mpls pw-class atom-eth
interface POS1/0/0
 ip address 10.1.1.2 255.255.255.0
 mpls ip
 mpls label protocol ldp
 clock source internal
 crc 32
!
interface Loopback0
 ip address 10.9.9.9 255.255.255.255
!
router ospf 10
 nsf
 network 10.9.9.9 0.0.0.0 area 0
 network 10.1.1.2 0.0.0.0 area 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Stateful switchover	Stateful Switchover
MPLS Label Distribution Protocol	MPLS Label Distribution Protocol (LDP)
Cisco nonstop forwarding	Cisco Nonstop Forwarding
Any Transport over MPLS	Any Transport over MPLS
L2VPN Interworking configuration	L2VPN Interworking
MPLS AToM and LDP commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
High availability commands	*Cisco IOS High Availability Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
MPLS Label Distribution Protocol MIB Version 8 Upgrade	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3036	*LDP Specification*
RFC 3478	*Graceful Restart Mechanism for Label Distribution*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for AToM NSF

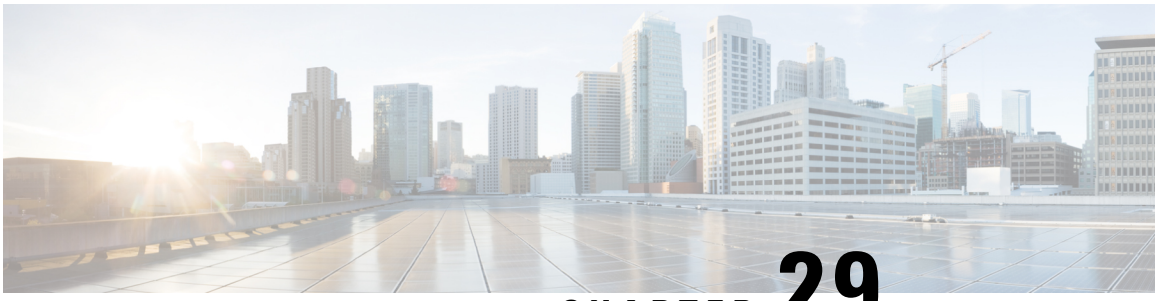
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 86: Feature Information for AToM NSF Any Transport over MPLS and AToM Graceful Restart

Feature Name	Releases	Feature Information
NSF/SSO--AToM ATM Attachment Circuit	Cisco IOS XE Release 2.3	This feature provides support for AToM NSF/SSO support for ATM over MPLS (ATMoMPLS), which allows ATMoMPLS to use Cisco nonstop forwarding (NSF), stateful switchover (SSO), and Graceful Restart (GR) to allow a Route Processor (RP) to recover from a disruption in control plane service without losing its Multiprotocol Label Switching (MPLS) forwarding state. In Cisco IOS XE Release 2.3, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **debug acircuit checkpoint**, **debug mpls l2transport checkpoint**, **show acircuit checkpoint**, **show mpls l2transport checkpoint**, **show mpls l2transport vc**.
ISSU--AToM ATM Attachment Circuit	Cisco IOS XE Release 2.3	This feature supports In Service Software Upgrade (ISSU) capability. Virtual Private LAN Services (VPLS) NSF/SSO and HA with ISSU work together to enable upgrades or downgrades of a Cisco IOS XE image without control and data plane outages. With ISSU, all message data structures that are used for checkpointing and exchanges between the active RP and standby RP are versioned. In Cisco IOS XE Release 2.3, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. No commands were introduced or modified for this feature.

Feature Name	Releases	Feature Information
NSF/SSO--Ethernet to Ethernet VLAN Interworking	Cisco IOS XE Release 2.4	The NSF/SS0--Ethernet to Ethernet VLAN Interworking features enables stateful switchover (SSO) and nonstop forwarding (NSF) capabilities for Ethernet to VLAN attachment circuits. Changes in the learned MAC address for interworking are reflected on the standby RP so that identical values exist on the Active and Standby RPs. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Routers. No commands were introduced or modified for this feature.



CHAPTER 29

Prerequisites for NSF SSO--MPLS VPN

- You must have a supported MPLS VPN network configuration. See Configuring MPLS VPNs for more information.
- The networking device that is to be configured for NSF must first be configured for stateful switchover (SSO). See Stateful Switchover for more information
- You must enable NSF on the routing protocols running between the provider (P) routers, provider edge (PE) routers, and customer edge (CE) routers. The supported routing protocols are Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), and Intermediate System-to-Intermediate System (IS-IS). See Configuring Nonstop Forwarding for more information.
- You must configure Cisco NSF support on the routers for Cisco Express Forwarding. See Configuring Nonstop Forwarding for more information.
- All neighbor networking devices must be NSF-aware. Peer routers must support the graceful restart of the protocol used to communicate with the NSF/SSO--MPLS VPN-capable router.

Restrictions for NSF SSO--MPLS VPN

- Tag Distribution Protocol (TDP) sessions are not supported. Only Label Distribution Protocol (LDP) sessions are supported.
- The NSF/SSO--MPLS VPN feature cannot be configured on label-controlled ATM (LC-ATM) interfaces.

Information About NSF SSO--MPLS VPN

Elements That Enable NSF SSO--MPLS VPN

VPN NSF requires several elements in order to work:

- VPN NSF uses the BGP Graceful Restart mechanisms to create MPLS forwarding entries for VPNv4 prefixes in NSF mode. The forwarding entries are preserved during a restart. BGP also saves prefix and corresponding label information and recovers the information after a restart.
- The NSF/SSO--MPLS VPN feature also uses NSF for the label distribution protocol in the core network (either MPLS Label Distribution Protocol, traffic engineering, or static labeling).
- The NSF/SSO--MPLS VPN feature uses NSF for the Interior Gateway Protocol (IGP) used in the core (OSPF or IS-IS).
- The NSF/SSO--MPLS VPN feature uses NSF for the routing protocols between the PE and CE routers.

How VPN Prefix Information Is Checkpointed to the Backup Route Processor

When BGP allocates local labels for prefixes, it checkpoints the local label binding in the backup RP. The checkpointing function copies state information from the active RP to the backup RP, thereby ensuring that the backup RP has an identical copy of the latest information. If the active RP fails, the backup RP can take over with no interruption in service. Checkpointing begins when the active RP does a bulk synchronization, which copies all of the local label bindings to the backup RP. After that, the active RP dynamically checkpoints individual prefix label bindings when a label is allocated or freed. This allows forwarding of labeled packets to continue before BGP reconverges.

How BGP Graceful Restart Preserves Prefix Information During a Restart

When a BGP Graceful Restart-capable router loses connectivity, it performs the following actions as the restarting router:

1. The restarting router establishes BGP sessions with other routers and relearns the BGP routes from other routers that are also capable of Graceful Restart. The restarting router waits to receive updates from the neighboring routers. When the neighboring routers send end-of-Routing Information Base (RIB) markers to indicate that they are done sending updates, the restarting router starts sending its own updates.
2. The restarting router accesses the checkpoint database to find the label that was assigned for each prefix. If it finds the label, it advertises it to the neighboring router. If it does not find the label, it allocates a new label and advertises it.
3. The restarting router removes any stale prefixes after a timer for stale entries expires.

A BGP Graceful Restart-capable peer router performs the following actions when it encounters a restarting router:

1. The peer router sends all the routing updates to the restarting router. When it has finished sending updates, the peer router sends an end-of-RIB marker to the restarting router.

2. The peer router does not immediately remove the BGP routes learned from the restarting router from its BGP routing table. As it learns the prefixes from the restarting router, the peer refreshes the stale routes if the new prefix and label information matches the old information.

If a router is not configured for the NSF/SSO--MPLS VPN feature and it attempts to establish a BGP session with a router that is configured with the NSF/SSO--MPLS VPN feature, the two routers create a normal BGP session but do not have the ability to perform the NSF/SSO--MPLS VPN feature.

How to Configure NSF SSO--MPLS VPN

Configuring NSF Support for Basic VPNs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **router bgp** *autonomous-system-number*
5. **bgp graceful-restart**
6. **bgp graceful-restart restart-time** *seconds*
7. **bgp graceful-restart stalepath-time** *seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** [**distributed**] **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding. • Use this command if Cisco Express Forwarding is not enabled by default on the router.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode.

	Command or Action	Purpose
Step 5	**bgp graceful-restart** **Example:** `Router(config-router)# bgp graceful-restart`	Enables BGP Graceful Restart on the router.
Step 6	**bgp graceful-restart restart-time** *seconds* **Example:** `Router(config-router)# bgp graceful-restart restart-time 200`	(Optional) Specifies the maximum time to wait for a graceful-restart-capable neighbor to come back up after a restart.
Step 7	**bgp graceful-restart stalepath-time** *seconds* **Example:** `Router(config-router)# bgp graceful-restart stalepath-time 400`	(Optional) Specifies the maximum time to hold on to the stale paths of a gracefully restarted peer. All stale paths are deleted after the expiration of this timer.
Step 8	**end** **Example:** `Router(config-router)# end`	Exits to privileged EXEC mode.

Verifying the Configuration

SUMMARY STEPS

1. **show ip bgp vpnv4 all labels**
2. **show ip bgp vpnv4 all neighbors**
3. **show ip bgp labels**
4. **show ip bgp neighbors**

DETAILED STEPS

Step 1 **show ip bgp vpnv4 all labels**

This command displays incoming and outgoing BGP labels for each route distinguisher. The following is sample output from the command:

Example:

```
Router# show ip bgp vpnv4 all labels

Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (vpn1)
   10.3.0.0/16      10.0.0.5        25/20
                    10.0.0.1        25/23
                    10.0.0.2        25/imp-null
```

```
   10.0.0.9/32      10.0.0.1        24/22
                    10.0.0.2        24/imp-null
```

Step 2 **show ip bgp vpnv4 all neighbors**

This command displays whether the BGP peers are capable of Graceful Restart. The following is sample output from the command:

Example:

```
Router# show ip bgp vpnv4 all neighbors
BGP neighbor is 10.0.0.1,  remote AS 100, internal link
  BGP version 4, remote router ID 10.0.0.1
  BGP state = Established, up for 02:49:47
  Last read 00:00:47, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family VPNv4 Unicast: advertised and received
    Graceful Restart Capabilty: advertised and received
      Remote Restart timer is 120 seconds
      Address families preserved by peer:
        VPNv4 Unicast
.
.
.
```

Step 3 **show ip bgp labels**

This command displays information about MPLS labels in the Exterior Border Gateway Protocol (EBGP) route table. The following is sample output from the command:

Example:

```
Router# show ip bgp labels
   Network          Next Hop      In label/Out label
   10.3.0.0/16      10.0.0.1        imp-null/imp-null
                    0.0.0.0         imp-null/nolabel
   10.0.0.9/32      10.0.0.1        21/29
   10.0.0.11/32     10.0.0.1        24/38
   10.0.0.13/32     0.0.0.0         imp-null/nolabel
   10.0.0.15/32     10.0.0.1        29/nolabel
                    10.0.0.1        29/21
```

Step 4 **show ip bgp neighbors**

This command displays whether the BGP peers are capable of Graceful Restart. The following is sample output from the command:

Example:

```
Router# show ip bgp neighbors
BGP neighbor is 10.0.0.1,  remote AS 100, external link
  BGP version 4, remote router ID 10.0.0.5
  BGP state = Established, up for 02:54:19
  Last read 00:00:18, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    ipv4 MPLS Label capability: advertised and received
    Graceful Restart Capabilty: advertised and received
      Remote Restart timer is 120 seconds
      Address families preserved by peer:
```

```
        IPv4 Unicast
.
.
.
```

Configuration Examples for NSF SSO--MPLS VPN

Example NSF SSO--MPLS VPN for a Basic MPLS VPN

The following sample output shows the configuration of the NSF/SSO--MPLS VPN feature on the CE and PE routers. SSO is enabled by default, and LDP is the default MPLS label protocol.

CE1 Router

```
ip cef
no ip domain-lookup
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
!
interface GigabitEthernet1/0/4
 ip address 10.0.0.1 255.0.0.0
 media-type 10BaseT
!
router ospf 100
 redistribute bgp 101
 nsf enforce global
 passive-interface GigabitEthernet1/0/4
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 101
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart network 10.0.0.0
 network 10.0.0.0
 neighbor 10.0.0.2 remote-as 100
```

PE1 Router

```
redundancy
mode sso
!
ip cef distributed
mpls ldp graceful-restart
mpls label protocol ldp
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
```

```
!
interface GigabitEthernet1/0/4
 ip vrf forwarding vpn1
 ip address 10.0.0.2 255.0.0.0
 !
 mpls ip
interface ATM3/0/0
 no ip address
!
interface ATM3/0/0.1 point-to-point
 ip unnumbered Loopback0
 mpls ip
!
router ospf 100
 passive-interface GigabitEthernet1/0/4
 nsf enforce global
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 10.14.14.14 remote-as 100
 neighbor 10.14.14.14 update-source Loopback0
!
address-family ipv4 vrf vpn1
 neighbor 10.0.0.1 remote-as 101
 neighbor 10.0.0.1 activate
 exit-address-family
!
address-family vpnv4
 neighbor 10.14.14.14 activate
 neighbor 10.14.14.14 send-community extended
 exit-address-family
```

PE2 Router

```
redundancy
mode sso
!
ip cef distributed
mpls ldp graceful-restart
mpls label protocol ldp
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
no mpls aggregate-statistics
!
!
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
!
interface ATM1/0
 no ip address
!
interface ATM1/0.1 point-to-point
 ip unnumbered Loopback0
 mpls ip
!
```

```
interface FastEthernet3/0/0
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 ip route-cache distributed
!
router ospf 100
 nsf enforce global
 passive-interface FastEthernet3/0/0
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 10.12.12.12 remote-as 100
 neighbor 10.12.12.12 update-source Loopback0
!
address-family ipv4 vrf vpn1
 neighbor 10.0.0.2 remote-as 102
 neighbor 10.0.0.2 activate
 exit-address-family
!
address-family vpnv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 exit-address-family
```

CE2 Router

```
ip cef
!
interface Loopback0
 ip address 10.13.13.13 255.255.255.255
!
interface FastEthernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip mroute-cache
!
router ospf 100
 redistribute bgp 102
 nsf enforce global
 passive-interface FastEthernet0/1
 network 10.0.0.0 0.255.255.255 area 100
!
router bgp 102
 no synchronization
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 network 10.0.0.0
 network 10.0.0.0
 neighbor 10.0.0.1 remote-as 100
```

Additional References

The following sections provide references related to the MPLS High Availability feature.

Related Documents

Related Topic	Document Title
MPLS VPNs Non Stop Forwarding	NSF/SSO—MPLS VPN
MPLS LDP Non Stop Forwarding	*NSF/SSO—MPLS LDP and LDP Graceful Restart*
AToM Non Stop Forwarding	NSF/SSO: Any Transport over MPLS and Graceful Restart
Cisco Express Forwarding	Cisco Express Forwarding: Command Changes
MIBs	• MPLS VPN: SNMP MIB Support • MPLS Label Distribution Protocol MIB Version 8 Upgrade • MPLS Label Switching Router MIB • MPLS Enhancements to Interfaces MIB • MPLS Traffic Engineering (TE) MIB
NSF/SSO	Cisco Nonstop Forwarding MPLS High Availability: Command Changes

Standards

Standard	Title
draft-ietf-mpls-bgp-mpls-restart.txt	Graceful Restart Mechanism for BGP with MPLS
draft-ietf-mpls-idr-restart.txt	Graceful Restart Mechanism for BGP

MIBs

MIB	MIBs Link
• MPLS VPN MIB • MPLS Label Distribution Protocol MIB Version 8 Upgrade	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3478	Graceful Restart Mechanism for Label Distribution

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for NSF SSO--MPLS VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 87: Feature Information for NSF/SSO--MPLS VPN

Feature Name	Releases	Feature Information
NSF/SSO--MPLS VPN	Cisco IOS XE Release 2.1	This feature allows a provider edge router to preserve data forwarding information in a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) when the primary Route Processor restarts.

CHAPTER 30

SSO and ISSU--MPLS VPN 6VPE and 6PE Support

This document provides information about configuring stateful switchover (SSO) and In Service Software Upgrade (ISSU) support for Cisco IOS XE VPN IPv6 provider edge (6VPE) and Cisco IOS XE IPv6 provider edge (6PE) routers over Multiprotocol Label Switching (MPLS).

Prerequisites for SSO and ISSU--MPLS VPN 6VPE and 6PE Support

- Your networking device must be running Cisco IOS XE 3.2S or a later release.
- Your network must be configured for a supported MPLS VPN. For information, see Configuring MPLS Layer 3 VPNs and Implementing IPv6 VPN over MPLS.
- SSO must be configured on the Route Processor (RP). For information, see Stateful Switchover.
- Your networking device must support the following:
  - IPv6 Cisco Express Forwarding (CEF)
  - IPv6 nonstop forwarding (NSF)
  - Label Distribution Protocol (LDP) Graceful Restart
- NSF must be enabled on the Border Gateway Protocol (BGP) and static routes that run between the provider (P), PE, and the customer edge (CE) routers. For configuration information, see Cisco Nonstop Forwarding.
- LDP Graceful Restart must be enabled if LDP is the protocol used in the MPLS core. For configuration information, see NSF/SSO-MPLS LDP and MPLS LDP Graceful Restart.

Restrictions for SSO and ISSU--MPLS VPN 6VPE and 6PE Support

- Only LDP sessions are supported.
- MPLS VPN 6VPE and 6PE Carrier Supporting Carrier (CSC) VPNs support only BGP. CSC configurations that use LDP are not supported.
- Only BGP and static routes are supported for 6VPE and 6PE.

Information About SSO and ISSU--MPLS VPN 6VPE and 6PE Support

Elements Supporting SSO and ISSU--MPLS VPN 6VPE and 6PE Support Features

The major elements supporting the functionality of the SSO and ISSU for Cisco IOS XE VPN 6vPE and 6PE features are the following:

- MPLS VPN--Forwards IP traffic using a VPN label that instructs the routers and switches in the network where to forward the packets based on preestablished IP routing information.
- BGP Graceful Restart--The BGP Graceful Restart feature is responsible for negotiating graceful restart capabilities, exchanging forwarding preservation states, and coordinating advertisements after session restarts. MPLS VPNs interact with BGP to exchange VPN routing and forwarding (VRF) routes and labels.
- IPv6 NSF--IPv6 NSF support enables IPv6 cache rebuilds during switchover using checkpointed Cisco Express Forwarding adjacencies.
- CEF/MFI--CEF and the MPLS Forwarding Infrastructure (MFI) are responsible for preserving forwarding entries and local labels across RP switchover.

Note If a router does not support the SSO and ISSU--MPLS VPN 6VPE and 6PE Support feature, prefix and label information is not preserved. After a switchover, BGP has to restart, relearn all routes, and install labels in the forwarding database. This can cause the loss of some network traffic.

How BGP Graceful Restart Works for MPLS VPN 6vPE and 6PE

BGP Graceful Restart behavior for IPv6 and VPNv6 is essentially the same as Graceful Restart behavior for IPv4 and VPNv4; the only difference is the addition of support for IPv6 and VPNv6 address families.

When you configure BGP Graceful Restart, BGP includes the Graceful Restart capability and negotiates the preservation states of address families, such as IPv4/VPNv4 and IPv6/VPNv6 address families.

Both BGP peers must agree on a Graceful Restart timer. After a BGP session comes up and finishes sending initial updates, each BGP peer sends an end-of-Routing Information Base (RIB) marker.

The SSO and ISSU--MPLS VPN 6VPE and 6PE Support features use the mechanisms defined in RFC 4724, *Graceful Restart Mechanism for BGP*.

How BGP Graceful Restart Preserves Prefix Information During a Restart

When a router that is capable of BGP Graceful Restart loses connectivity, the following happens to the restarting router:

1. The router establishes BGP sessions with other routers and relearns the BGP routes from other routers that are also capable of Graceful Restart. The restarting router waits to receive updates from the neighboring routers. When the neighboring routers send end-of-RIB markers to indicate that they are done sending updates, the restarting router starts sending its own updates.
2. The restarting router recovers labels from the MFI database for each prefix. If the router finds the label, it advertises the label to the neighboring router. If the router does not find the label, it allocates a new label from the database and advertises it.
3. The restarting router removes any stale prefixes after a timer for stale entries expires.

When a peer router that is capable of BGP Graceful Restart encounters a restarting router, it does the following:

1. The peer router sends all of the routing updates to the restarting router. When it has finished sending updates, the peer router sends an end-of RIB marker to the restarting router.
2. The peer router does not immediately remove the BGP routes learned from the restarting router from its BGP routing table. As it learns the prefixes from the restarting router, the peer refreshes the stale routes if the new prefix and label information matches the old information.

ISSU Support for MPLS VPN 6vPE and 6PE

The ISSU process allows Cisco IOS XE software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco IOS XE software to be modified while packet forwarding continues, which increases network availability and reduces downtime caused by planned software upgrades.

ISSU support for MPLS 6vPE and 6PE relies on 6vPE and 6PE NFS/SSO capability on the platform to minimize disruption on the forwarding plane.

SSO Support for MPLS VPN 6VPE and 6PE

SSO for 6VPE and 6PE supports the following configurations:

- NSF/SSO for IPv4 and VPNv4 coexistence
- Basic 6VPE and 6PE over MPLS core technology
- BGP multipath configuration

SSO for 6VPE supports the following configurations:

- Per-VRF label configuration
- Interautonomous systems (Inter-AS) topologies, including options B and C

- CSC when IPv6 + labels is configured on the PE-CE link

Because the SSO feature maintains stateful protocol and application information, user session information is maintained during a switchover and line cards continue to forward network traffic with no loss of sessions, providing improved network availability. SSO initializes and configures the standby RP and synchronizes state information, which can reduce the time required for routing protocols to converge. Network stability may be improved with the reduction in the number of route flaps created when routers in the network fail and lose their routing tables.

When RP switchover happens, forwarding information is preserved by MFI and Cisco Express Forwarding on both line cards and the standby RP. VPNv6 prefix and local label mapping is preserved in the forwarding database. When the standby RP becomes the new active RP, 6PE and 6vPE traffic continues to be forwarded with minimal interruption.

When a BGP session restarts on the new active RP, the new active RP does not have any prior state information about prefixes or labels. The new active RP must relearn VPNv6 prefixes from its peers. As the new active RP learns the VPNv6 prefixes, it tries to get new local labels the same way it does when it first comes up. If the MFI database has the preserved copy of the local label for a prefix, the MFI database gives the local label to BGP and BGP then maintains the same local label. If the MFI database does not have a preserved local label for the prefix, MFI allocates a new one.

BGP Graceful Restart Support for MPLS VPN Configurations

Graceful Restart Support for a Basic 6VPE Setup

For PE- to-CE external BGP (eBGP), Graceful Restart capability is supported for IPv6 address families. For PE-to-PE interior BGP (iBGP) sessions with or without a route reflector (RR) in the core, BGP Graceful Restart capability supports VPNv6 address families.

When the PE router resets, the connected CE router retains IPv6 prefixes that it received from the PE router and marks the prefixes as stale. If the eBGP session does not reestablish within the specified restart time or the session reestablishes but does not set the restart or forwarding state bit, the CE router removes the stale IPv6 routes. If the eBGP session reestablishes within the specified restart time and has both the forwarding and restart bits set, the CE router removes the stale state from the IPv6 routes when it receives the updates from PE router. After the CE router receives the end-of-RIB marker, it removes or withdraws the rest of the stale information, if any exists.

The restarting PE router waits for an end-of-RIB marker from all BGP-capable peers including iBGP peers and eBGP peers. The PE router begins to calculate the best path and send out initial updates only after receiving an end-of-RIB marker from all BGP capable peers.

Graceful Restart for 6VPE in Carrier Supporting Carrier and Interautonomous System Setups

The same Graceful Restart capabilities for route preservation that apply to a basic 6VPE setup apply to a CSC and Inter-AS setup. IPv6 or VPNv6 routes and labels are preserved during switchover.

In a CSC configuration, when send-labels are configured between a CSC-PE and CSC-CE eBGP connection, labels are preserved along with IPv6 BGP routes when one of the peers restarts.

In Inter-AS option B and options C setups, VPNv6 routes and labels are preserved on an Autonomous System Border Router (ASBR) or route reflector when the VPNv6 peer restarts.

How to Configure SSO and ISSU--MPLS VPN 6VPE and 6PE Support

Note Unlike SSO, which is a mode of operation for the device and a prerequisite for performing ISSU, the ISSU process is a series of steps performed while the router or switch is in operation. For information on performing ISSU upgrades on the Cisco ASR 1000 Series Aggregation Services Router, see the "In Service Software Upgrade (ISSU)" module in the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*.

Configuring SSO for a Basic MPLS 6VPE and 6PE Setup

Perform this task to configure SSO for a basic MPLS 6VPE and 6PE setup.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **ipv6 unicast-routing**
5. **ipv6 cef distributed**
6. **redundancy**
7. **mode sso**
8. **exit**
9. **router bgp** *autonomous-system-number*
10. **bgp graceful-restart restart-time** *seconds*
11. **bgp graceful-restart stalepath-time** *seconds*
12. **bgp graceful-restart**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding.
Step 4	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 5	**ipv6 cef distributed** **Example:** `Router(config)# ipv6 cef distributed`	Enables distributed Cisco Express Forwarding for IPv6.
Step 6	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.
Step 7	**mode sso** **Example:** `Router(red-config)# mode sso`	Sets the redundancy configuration mode to SSO.
Step 8	**exit** **Example:** `Router(red-config)# exit`	Exits to global configuration mode.
Step 9	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 1000`	Enters router configuration mode and configures the BGP routing process.
Step 10	**bgp graceful-restart restart-time** *seconds* **Example:** `Router(config-router)# bgp graceful-restart restart-time 180`	Enables the BGP graceful restart timer capability globally for all BGP neighbors and sets the maximum time period that the local router will wait for a graceful-restart-capable neighbor to return to normal operation after a restart event occurs.
Step 11	**bgp graceful-restart stalepath-time** *seconds* **Example:** `Router(config-router)# bgp graceful-restart stalepath-time 420`	Enables the BGP graceful restart stale path timer capability globally for all BGP neighbors and sets the maximum time period that the local router will hold stale paths for a restarting peer.

	Command or Action	Purpose
Step 12	**bgp graceful-restart** **Example:** `Router(config-router)# bgp graceful-restart`	Enables the BGP graceful restart capability globally for all BGP neighbors.
Step 13	**end** **Example:** `Router(config-router)# end`	Exits to privileged EXEC mode.

Verifying SSO and ISSU Support for 6VPE and 6PE

Perform this task to verify SSO and ISSU support for 6VPE and 6PE routers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbor**
3. **show ip bgp vpnv6 unicast vrf** *vrf-name*
4. **show ip bgp ipv6 unicast**
5. **show mpls forwarding**
6. **show ipv6 cef vrf** *vrf-name*

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip bgp neighbor**

Use this command to verify that the IPv6 address family and VPNv6 address family entries are preserved. For example:

Example:

```
Router# show ip bgp neighbor
BGP neighbor is 10.2.2.2, remote AS 100, internal link
  BGP version 4, remote router ID 10.2.2.2
  BGP state = Established, up for 00:02:42
  Last read 00:00:36, last write 00:00:36, hold time is 180, keepalive
.
.
.
  Neighbor capabilities:
.
```

```
.
.
    Graceful Restart Capability: advertised and received
      Remote Restart timer is 120 seconds
      Address families advertised by peer:
        IPv6 Unicast (was preserved), VPNv6 Unicast (was preserved)
```

IPv6 Unicast (was preserved), VPNv6 Unicast (was preserved) is displayed in the Graceful Restart Capability section of the output only after the peer restarts.

Step 3 **show ip bgp vpnv6 unicast vrf** *vrf-name*

Use this command to verify that VPNv6 entries are marked as stale during switchover. For example:

Example:

```
Router# show ip bgp vpnv6 unicast vrf vpn1
BGP table version is 10, local router ID is 10.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf vpn1)
S>iA::1/128         ::FFFF:10.2.2.2          0    100      0 200 ?
*> A::5/128         A::4:5:5                 0             0 200 ?
S>iA::1:2:0/112     ::FFFF:10.2.2.2          0    100      0 ?
*  A::4:5:0/112     A::4:5:5                 0             0 200 ?
```

Step 4 **show ip bgp ipv6 unicast**

Use this command to verify that VPNv6 entries are marked as stale during switchover. For example:

Example:

```
Router# show ip bgp ipv6 unicast
BGP table version is 9, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> A::1/128         ::                       0         32768 ?
S  A::1:2:0/112     A::1:2:2                 0             0 100 ?
*>                  ::                       0         32768 ?
S> A::4:5:0/112     A::1:2:2                               0 100 ?
Router#
```

Step 5 **show mpls forwarding**

Use this command to show preserved forwarding entries for IPv6 and VPNv6 prefixes. The sample output is from a PE router in a CSC configuration. Enter the command on the active and the standby router. Compare the sample output from the active router with the sample output from the standby router. Following is sample output from the active router;

Example:

```
Router# show mpls forwarding
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
18         Pop Label      10.3.3.3/32      0             FEt1/0/0      10.2.3.3
19         Pop Label      10.3.4.0/24      0             FEt1/0/0      10.2.3.3
20         17             10.4.4.4/32      0             FEt1/0/0      10.2.3.3
21         Pop Label      10.1.2.1/32[V]   0             FEt0/0/0      10.1.2.1
22         Pop Label      A::1:2:0/112[V]  0             aggregate/vpn1
```

```
23         Pop Label      A::1:2:1/128[V]  0             FEt0/0/0      A::1:2:1
24         Pop Label      10.1.2.0/24[V]   0             aggregate/vpn1
25         Pop Label      A::1:2:2/128[V]  0             aggregate/vpn1
26         18             A::1/128[V]      0             FEt0/0/0
FE80::A8BB:CCFF:FE03:2101
27         26             10.4.5.5/32[V]   0             FEt1/0/0      10.2.3.3
28         25             10.4.5.0/24[V]   0             FEt1/0/0      10.2.3.3
29         22             A::4:5:5/128[V]  0             FEt1/0/0      10.2.3.3
30         21             A::4:5:0/112[V]  0             FEt1/0/0      10.2.3.3
31         23             A::4:5:4/128[V]  0             FEt1/0/0      10.2.3.3
32         24             A::5/128[V]      0             FEt1/0/0      10.2.3.3
33         Pop Label      10.1.2.2/32[V]   0             aggregate/vpn1
34         Pop Label      10.1.1.1/32[V]   0             FEt0/0/0      10.1.2.1
35         27             10.4.5.4/32[V]   0             FEt1/0/0      10.2.3.3
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
36         28             10.5.5.5/32[V]   0             FEt1/0/0      10.2.3.3
```

Following is sample output from the standby router:

Example:

```
Standby-Router# show mpls forwarding
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
18         Pop Label      10.3.3.3/32      0             FEt1/0/0      10.2.3.3
19         Pop Label      10.3.4.0/24      0             FEt1/0/0      10.2.3.3
20         17             10.4.4.4/32      0             FEt1/0/0      10.2.3.3
21         Pop Label      10.1.2.1/32[V]   0             FEt0/0/0      10.1.2.1
22         Pop Label      A::1:2:0/112[V]  0             aggregate/vpn1
23         Pop Label      A::1:2:1/128[V]  0             FEt0/0/0      A::1:2:1
24         Pop Label      10.1.2.0/24[V]   0             aggregate/vpn1
25         Pop Label      A::1:2:2/128[V]  0             aggregate/vpn1
26         18             A::1/128[V]      0             FEt0/0/0
FE80::A8BB:CCFF:FE03:2101
27         26             10.4.5.5/32[V]   0             FEt1/0/0      10.2.3.3
28         25             10.4.5.0/24[V]   0             FEt1/0/0      10.2.3.3
29         22             A::4:5:5/128[V]  0             FEt1/0/0      10.2.3.3
30         21             A::4:5:0/112[V]  0             FEt1/0/0      10.2.3.3
31         23             A::4:5:4/128[V]  0             FEt1/0/0      10.2.3.3
32         24             A::5/128[V]      0             FEt1/0/0      10.2.3.3
33         Pop Label      10.1.2.2/32[V]   0             aggregate/vpn1
34         Pop Label      10.1.1.1/32[V]   0             FEt0/0/0      10.1.2.1
35         27             10.4.5.4/32[V]   0             FEt1/0/0      10.2.3.3
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
36         28             10.5.5.5/32[V]   0             FEt1/0/0      10.2.3.3
```

Step 6 **show ipv6 cef vrf** *vrf-name*

Use this command to show preserved forwarding entries for IPv6 and VPNv6 prefixes. This sample output is also from a PE router in a CSC configuration. Enter the command on the active and the standby router. Compare the sample output from the active router with the sample output from the standby router. Following is the output from the active router:

Example:

```
Router# show ipv6 cef vrf vrf1
::/0
  no route
::/127
  discard
A::1/128
  nexthop FE80::A8BB:CCFF:FE03:2101 FastEthernet0/0/0 label 18
```

```
A::5/128
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 24
A::1:2:0/112
  attached to FastEthernet0/0/0
A::1:2:1/128
  attached to FastEthernet0/0/0
A::1:2:2/128
  receive for FastEthernet0/0/0
A::4:5:0/112
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 21
A::4:5:4/128
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 23
A::4:5:5/128
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 22
FE80::/10
```

Following is sample output from the standby router:

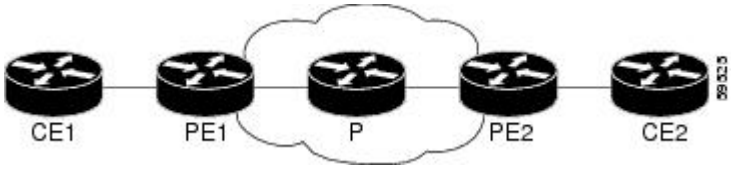
Example:

```
Standby-Router# show ipv6 cef vrf vrf1
::/0
  no route
::/127
  discard
A::1/128
  nexthop FE80::A8BB:CCFF:FE03:2101 FastEthernet0/0/0 label 18
A::5/128
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 24
A::1:2:0/112
  attached to FastEthernet0/0/0
A::1:2:1/128
  attached to FastEthernet0/0/0
A::1:2:2/128
  receive for FastEthernet0/0/0
A::4:5:0/112
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 21
A::4:5:4/128
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 23
A::4:5:5/128
  nexthop 10.2.3.3 FastEthernet1/0/0 label 17 22
FE80::/10
```

Configuration Examples for Configuring SSO and ISSU--MPLS VPN 6VPE and 6PE Support

The figure below illustrates a basic 6VPE or 6PE network configuration.

Figure 42: Sample Basic 6VPE/6PE Network Configuration



This section provides the following configuration examples for PE1 routers in a basic 6VPE or 6PE network configuration:

Example Configuring SSO for a Basic 6VPE Setup

The following is a configuration example for a PE1 router in a basic 6VPE setup (see the figure above) that includes VPNv6 and VPNv6 address families:

```
vrf definition vpn1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
!
ip cef distributed
!
ipv6 unicast-routing
ipv6 cef distributed
mpls ldp graceful-restart ! <==+ Command to configure LDP Graceful Restart
mpls label protocol ldp
redundancy
 mode sso
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
 ipv6 address A::2/128
!
interface FastEthernet0/0/0
 vrf forwarding vpn1
 ip address 10.1.2.2 255.255.255.0
 ipv6 address A::1:2:2/112
!interface FastEthernet1/0/0
 ip address 10.2.3.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!router ospf 10
 log-adjacency-changes
 nsf
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120      ! <=== This command,
 bgp graceful-restart stalepath-time 360    ! <=== this command, and
 bgp graceful-restart            ! <=== this command configures NFS/SSO for a 6VPE router.
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.4.4.4 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 10.4.4.4 activate
  neighbor 10.4.4.4 send-community extended
 exit-address-family
 !
 address-family vpnv6
```

```
 neighbor 10.4.4.4 activate
 neighbor 10.4.4.4 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1
 no synchronization
 redistribute connected
 redistribute static
 neighbor 10.1.2.1 remote-as 200
 neighbor 10.1.2.1 update-source FastEthernet0/0/0
 neighbor 10.1.2.1 activate
exit-address-family
!
address-family ipv6 vrf vpn1
 redistribute connected
 redistribute static
 no synchronization
 neighbor A::1:2:1 remote-as 200
 neighbor A::1:2:1 update-source FastEthernet0/0/0
 neighbor A::1:2:1 activate
exit-address-family
```

Example Configuring SSO for a Basic 6PE Setup

This following is a configuration example for the PE1 router in a basic 6PE setup (see the figure above):

```
ip cef distributed
!
ipv6 unicast-routing
ipv6 cef distributed
mpls ldp graceful-restart ! <=== Command to configure LDP Graceful Restart
mpls label protocol ldp
redundancy
 mode sso
interface Loopback0
 ip address 10.11.11.1 255.255.255.255
 ipv6 address BEEF:11::1/64
interface FastEthernet0/0/0
 ip address 10.50.1.2 255.255.255.0
 ipv6 address 4000::72B/64
 ipv6 address 8008::72B/64
!
interface FastEthernet1/0/0
 ip address 10.40.1.2 255.255.255.0
 mpls ip
!
router ospf
nsf
network 0.0.0.0 0.0.0.0 area 0
!
router bgp 100
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120               ! <=== This command,
 bgp graceful-restart stalepath-time 360            ! <=== this command, and
 bgp graceful-restart             ! <=== this command configures NFS/SSO for a 6PE router.

 neighbor 8008::72A remote-as 200
 neighbor 10.10.10.1 remote-as 100
 neighbor 10.10.10.1 update-source Loopback0
 !
 address-family ipv4
  no synchronization
```

```
 redistribute connected
 no neighbor 8008::72A activate
 neighbor 10.10.10.1 activate
 no auto-summary
exit-address-family
!
address-family ipv6
 redistribute connected
 no synchronization
 neighbor 8008::72A activate
 neighbor 10.10.10.1 activate
 neighbor 10.10.10.1 send-label
exit-address-family
```

Additional References

Related Documents

Related Topic	Document Title
6VPE over MPLS	Implementing IPv6 VPN over MPLS
6PE over MPLS	Implementing IPv6 over MPLS
Cisco IOS XE commands	Cisco IOS Master Command List, All Releases
Cisco IOS XE MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Cisco nonstop forwarding	Cisco Nonstop Forwarding
ISSU	• Cisco IOS XE In Service Software Upgrade Process • "In Service Software Upgrade (ISSU)" module in the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*
MPLS LDP NSF/SSO and Graceful Restart	NSF/SSO-MPLS LDP and MPLS LDP Graceful Restart
MPLS VPNs	Configuring MPLS Layer 3 VPNs
NFS/SSO for MPLS VPN	NSF/SSO--MPLS VPN
SSO	Stateful Switchover

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4659	BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN
RFC 4724	Graceful Restart Mechanism for BGP
RFC 4781	Graceful Restart Mechanism for BGP with MPLS
FRC 4798	Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SSO and ISSU--MPLS VPN 6VPE and 6PE Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 88: Feature Information for SSO and ISSU--MPLS VPN 6VPE and 6PE Support

Feature Name	Releases	Feature Information
ISSU--MPLS VPN 6VPE and 6PE ISSU Support	Cisco IOS XE 3.2S	This feature provides ISSU support for Cisco IOS XE VPN IPv6 provider edge router (6VPE) over MPLS and Cisco IOS XE IPv6 provider edge router (6PE) over MPLS. In Cisco IOS XE 3.2S, this feature was introduced for Cisco ASR 1000 Series Aggregation Services Routers. This feature introduced no new or modified commands.
SSO--MPLS VPN 6VPE and 6PE SSO Support	Cisco IOS XE 3.2S	This feature provides SSO support for Cisco IOS XE VPN IPv6 provider edge router (6VPE) over MPLS and Cisco IOS XE IPv6 provider edge router (6PE) over MPLS. In Cisco IOS XE 3.2S, this feature was introduced for Cisco ASR 1000 Series Aggregation Services Routers. This feature introduced no new or modified commands.

Glossary

6PE router --IPv6 provider edge (PE) router. A router running a Border Gateway Protocol (BGP)-based mechanism to interconnect IPv6 islands over a Multiprotocol Label Switching (MPLS)-enabled IPv4 cloud.

6VPE router --Provider edge router providing Border Gateway Protocol (BGP)-Multiprotocol Label Switching (MPLS) IPv6 Virtual Private Network (VPN) service over an IPv4-based MPLS core. It is a IPv6 VPN provider edge (PE), dual-stack router that implements 6PE concepts on the core-facing interfaces.

BGP --Border Gateway Protocol. An interdomain routing protocol designed for the global Internet. Exterior Border Gateway Protocols (eBGPs) communicate among different autonomous systems. Interior Border Gateway Protocols (iBGPs) communicate among routers within a single autonomous system.

CE router --customer edge router. A router that is part of a customer network and interfaces to a provider edge (PE) router.

Cisco Express Forwarding --An advanced Layer 3 IP switching technology. It optimizes network performance and scalability for all kinds of networks.

eBGP --external Border Gateway Protocol.

graceful restart --A process for helping an RP restart after a node failure has occurred.

iBGP --Interior Border Gateway Protocol.

ISSU --In Service Software Upgrade. Software upgrade without service interruption.

LDP --Label Distribution Protocol. A standard protocol between Multiprotocol Label Switching (MPLS)-enabled routers to negotiate the labels (addresses) used to forward packets.

MPLS --Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and switches in the network where to forward the packets based on preestablished IP routing information.

NSF --nonstop forwarding. The ability of a router to continue to forward traffic to a router that may be recovering from a failure. Also, the ability of a router recovering from a failure to continue to correctly forward traffic sent to it by a peer.

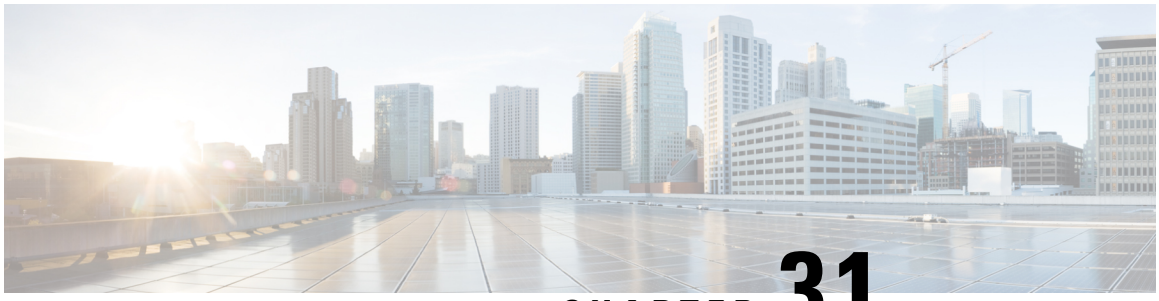
PE router --provider edge router. The PE router is the entry point into the service provider network. The PE router is typically deployed on the edge of the network and is administered by the service provider. The PE router is the redistribution point between EIGRP and BGP in PE to CE networking.

RIB --Routing Information Base. Also called the routing table.

SSO --stateful switchover. SSO refers to the implementation of Cisco IOS XE software that allows applications and features to maintain a defined state between an active and standby RP. When a switchover occurs, forwarding and sessions are maintained. Along with NSF, SSO makes an RP failure undetectable to the network.

VPN --Enables IP traffic to travel securely over a public TCP/IP network by encrypting traffic from one network to another. A VPN uses tunneling to encrypt all information at the IP level.

VRF --Virtual Private Network (VPN) routing and forwarding instance. A VRF consists of an IP routing table, a derived routing table, a set of interfaces that use the forwarding table. and a set of rules and routing information that defines a customer VPN site that is attached to a provider edge (PE) router.



CHAPTER 31

SSO Support for MPLS TE Autotunnel and Automesh

The SSO Support for MPLS TE Autotunnel and Automesh feature provides full stateful switchover (SSO), Cisco nonstop forwarding (NSF), and In Service Software Upgrade (ISSU) support for autotunnel primary and backup TE tunnels feature and for autotunnel mesh group TE tunnels feature.

The NSF with SSO provides continuous packet forwarding even during a network processor hardware or software failure. In a redundant system, the secondary processor recovers control plane service during a critical failure in the primary processor. SSO synchronizes the network state information between the primary and the secondary processor.

Note For brevity in this document, the Autotunnel Primary and Backup feature is called Autotunnel. The Autotunnel Mesh Groups feature is called Automesh.

Prerequisites for SSO Support for MPLS TE Autotunnel and Automesh

- The MPLS TE RSVP Graceful Restart feature must be enabled on the stateful switchover (SSO) device and its neighbor devices.
- NSF must be configured on the IGP that is configured for TE. You must specify either the **nsf cisco** or the **nsf ietf** router configuration command or the recovery of TE tunnels might fail.
- The MPLS TE Autotunnel feature must be configured.
- The MPLS TE Automesh feature must be configured.

Note The SSO Support for MPLS TE Autotunnel and Automesh feature obsoletes the MPLS TE Autotunnel and SSO Coexistence feature available with the MPLS TE Autotunnel feature and the MPLS TE Automesh feature.

Restrictions for SSO Support for MPLS TE Autotunnel and Automesh

- The SSO Support for MPLS TE Autotunnel and Automesh feature is supported only on hardware platforms with dual Route Processors (RPs) that support SSO and Cisco NSF.
- SSO and Fast Reroute (FRR) double failure cases are not supported.
- To keep the Autotunnel and Automesh configurations synchronized between the active and standby RPs, you can no longer modify an existing Autotunnel or Automesh interface by using the **interface tunnel** command. This action is prohibited by the software.
- You can no longer use the following commands as a way for disabling the Autotunnel or the Automesh feature:
  - **clear mpls traffic-eng auto-tunnel primary**
  - **clear mpls traffic-eng auto-tunnel backup**
  - **clear mpls traffic-eng auto-tunnel mesh**

  Instead, use the **no** form of these commands:
  - **no mpls traffic-eng auto-tunnel primary onehop**
  - **no mpls traffic-eng auto-tunnel backup**
  - **no mpls traffic-eng auto-tunnel mesh**

Information About SSO Support for MPLS TE Autotunnel and Automesh

Overview of SSO Support for MPLS TE Autotunnel and Automesh

With the SSO Support for MPLS TE Autotunnel and Automesh feature, once you enable the device for the Autotunnel feature or for the Automesh feature by using the **mpls traffic-eng auto-tunnel primary onehop**, **mpls traffic-eng auto-tunnel backup**, or the **mpls traffic-eng auto-tunnel mesh** commands, the device starts creating the specified type of autotunnel on both the active and standby RPs. No additional configuration is needed to implement the SSO Support for MPLS TE Autotunnel and Automesh feature.

When the **no** form of these commands is executed, the SSO feature is disabled on both the active and the standby RPs.

The Autotunnel feature enables a device to dynamically build backup tunnels and to dynamically create one-hop primary tunnels on all interfaces that have been configured with MPLS TE tunnels.

The Automesh feature allows a network administrator to configure TE label switched paths (LSPs). In a network topology where edge label switch routers (LSRs) are connected by core LSRs, the Automesh feature automatically constructs a mesh of TE LSPs among the provider edge (PE) devices.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Master Commands List, All Releases*
MPLS traffic engineering commands	*Multiprotocol Label Switching Command Reference*
MPLS traffic engineering—Autotunnel Mesh Groups feature	*MPLS Traffic Engineering Path Calculation and Setup Configuration Guide*
MPLS traffic engineering—Autotunnel Primary and Backup feature	*MPLS Traffic Engineering Path Link and Node Protection Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SSO Support for MPLS TE Autotunnel and Automesh

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 89: Feature Information for SSO Support for MPLS TE Autotunnel and Automesh

Feature Name	Releases	Feature Information
SSO Support for MPLS TE Autotunnel and Automesh	15.2(2)S Cisco IOS XE Release 3.6S	The SSO Support for MPLS TE Autotunnel and Automesh feature provides full stateful switchover (SSO), Cisco nonstop forwarding (NSF), and In Service Software Upgrade (ISSU) support for the autotunnel primary and backup TE tunnels and for the autotunnel mesh group TE tunnels. The following commands were introduced or modified: **clear mpls traffic-eng auto-tunnel backup tunnel**, **clear mpls traffic-eng auto-tunnel mesh tunnel**, **clear mpls traffic-eng auto-tunnel primary tunnel**, **debug mpls traffic-eng auto-tunnel backup**, **debug mpls traffic-eng auto-tunnel primary**, **debug mpls traffic-eng ha sso**, **mpls traffic-eng auto-tunnel backup**, **mpls traffic-eng auto-tunnel mesh**, **mpls traffic-eng auto-tunnel primary onehop**, **show ip rsvp high-availability counters**, **show ip rsvp high-availability database**, **show ip rsvp high-availability database summary**, **show ip rsvp high-availability summary**, **show mpls traffic-eng auto-tunnel primary**.

Glossary

backup tunnel—An MPLS traffic engineering tunnel used to protect another (primary) tunnel's traffic when a link or node failure occurs.

Fast Reroute—Fast Reroute (FRR) is a mechanism for protecting MPLS traffic engineering LSPs from link and node failure by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend devices attempt to establish end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

graceful restart—A process for helping an RP restart after a node failure has occurred.

ISSU—In Service Software Upgrade. Software upgrade without service interruption.

LSP—label switched path. A path that a labeled packet follows over several hops, starting at an ingress LSR and ending at an egress LSR.

LSR—label switch router. A Layer 3 device that forwards a packet based on the value of a label encapsulated in the packet.

mesh group—A set of label switch routers (LSRs) that are members of a full or partial network of traffic engineering label switched paths (LSPs).

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the devices in the network where to forward the packets based on preestablished IP routing information.

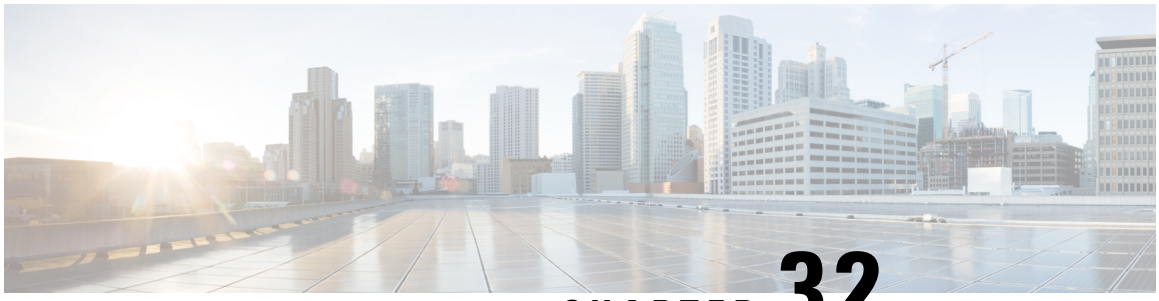
NSF—nonstop forwarding. The ability of a device to continue to forward traffic to a device that may be recovering from a failure. Also, the ability of a device recovering from a failure to continue to correctly forward traffic sent to it by a peer.

primary tunnel—An MPLS tunnel whose LSP can be fast-rerouted if there is a failure.

SSO—stateful switchover. SSO refers to the implementation of Cisco software that allows applications and features to maintain a defined state between an active and standby RP. When a switchover occurs, forwarding and sessions are maintained. Along with NSF, SSO makes an RP failure undetectable to the network.

TE—traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel—A secure communication path between two peers. A traffic engineering tunnel is a label switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than a normal Layer 3 device.



CHAPTER 32

MPLS Traffic Engineering Nonstop Routing Support

The MPLS Traffic Engineering Nonstop Routing Support feature assists the Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) routing devices to recover from an interruption in service. This feature also defines the checkpoint and recovery scheme for the devices.

Prerequisites for MPLS Traffic Engineering Nonstop Routing Support

Your network must support the following Cisco features before you enable Multiprotocol Label Switching (MPLS) Traffic Engineering (TE):

- MPLS
- Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)

Before enabling MPLS TE Nonstop Routing (NSR), a full-mode check needs to be done by the system to verify if the **mpls traffic-eng nsr** command is permitted or is restricted due to conflicts or user privileges.

Restrictions for MPLS Traffic Engineering Nonstop Routing Support

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) and Resource Reservation Protocol (RSVP) Graceful Restart (GR) are both mutually exclusive recovery mechanisms. Hence, MPLS TE NSR cannot be enabled when RSVP GR is enabled.

How to Configure MPLS Traffic Engineering Nonstop Routing Support

MPLS Traffic Engineering Nonstop Routing Support Overview

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) enables routing devices to recover from an interruption in service. The MPLS TE NSR functionality defines a checkpoint for the control plane of the routing devices. Resource Reservation Protocol (RSVP) Graceful Restart (GR) is another method for recovering and restarting interrupted services.

To avoid conflict and guarantee deterministic behavior, only one of the above mentioned recovery methods can be configured at a given time.

The MPLS TE NSR feature differs from the RSVP GR feature in the following ways:

- MPLS TE NSR devices are fully independent and do not rely on neighbor nodes for a stateful switchover (SSO) recovery.
- MPLS TE NSR supports the SSO recovery of Fast Reroute (FRR) active tunnels.
- MPLS TE NSR has an active standby mode. This helps with most of the recovery states being created before the SSO recovery actually happens, ensuring a faster recovery after SSO.
- MPLS TE NSR **show** commands display recovery information in standby mode as well.
- Label switched paths (LSPs) which are not fully signaled, do not resume signaling after an interruption and will go down on SSO.

How to Configure MPLS Traffic Engineering Nonstop Routing Support

Configuring MPLS Traffic Engineering Nonstop Routing Support

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip cef**
4. **mpls traffic-eng nsr**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Device(config)# ip cef`	Enables standard Cisco Express Forwarding operations.
Step 4	**mpls traffic-eng nsr** **Example:** `Device(config)# mpls traffic-eng nsr`	Enables the MPLS Traffic Engineering (TE) Non-Stop Routing (NSR) functionality on a device. **Note** Enabling the MPLS TE NSR functionality automatically enables the Resource Reservation Protocol (RSVP) NSR functionality as well.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying MPLS Traffic Engineering Nonstop Routing Support

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng nsr**
3. **show mpls traffic-eng nsr counters**
4. **show mpls traffic-eng nsr database**
5. **show mpls traffic-eng nsr oos**
6. **show mpls traffic-eng nsr summary**

7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**show mpls traffic-eng nsr**
	Example:
	`Device# show mpls traffic-eng nsr`
	`  counters  TE NSR counters`
	`  database  TE NSR check pointed data`
	`  oos  TE NSR out of sync database`
	`  summary  TE NSR summary`
	`  |  Output modifiers`
	`  <cr>`
		Displays options to obtain Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) configuration information such as the database status, counter numbers, devices which are out of sync, and the summary of all the devices.

Step 3	**show mpls traffic-eng nsr counters** **Example:** `Device# show mpls traffic-eng nsr counters`	Displays information about the data structures or states that are successfully created or removed, along with errors counts.
Step 4	**show mpls traffic-eng nsr database** **Example:** `Device# show mpls traffic-eng nsr database`	Displays information pertaining to the write and read databases supporting MPLS TE NSR. The write and read databases store the data that is used for recovering TE state on a standby device after stateful switchover (SSO).
Step 5	**show mpls traffic-eng nsr oos** **Example:** `Device# show mpls traffic-eng nsr oos`	Displays information pertaining to the out of sync databases supporting MPLS TE NSR. The out of sync databases indicate the devices whose states are not in sync with each other.
Step 6	**show mpls traffic-eng nsr summary** **Example:** `Device# show mpls traffic-eng nsr summary`	Displays a summary of MPLS TE NSR information such as the current TE NSR state (standby-hot / recovering / staling / active), recovery time, and the recovery result (success / failure).
Step 7	**end** **Example:** `Device(config)# end`	Exits privileged EXEC mode.

Configuration Examples for MPLS Traffic Engineering Nonstop Routing Support

Example: Configuring MPLS Traffic Engineering Nonstop Routing Support

The following example shows how to configure Multiprotocol (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) support on a device:

```
enable
 configure terminal
  ip cef
  mpls traffic-eng nsr
  end
```

Example: Verifying MPLS Traffic Engineering Nonstop Routing Support

Displaying MPLS Traffic Engineering Nonstop Routing Support Verification Options

The following example shows how to display the options that help you verify Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Nonstop Routing (NSR) information:

```
enable
 show mpls traffic-eng nsr ?
  counters  TE NSR counters
  database  TE NSR check pointed data
  oos       TE NSR out of sync database
  summary   TE NSR summary
  |         Output modifiers
  <cr>
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Counters

The following example shows how to verify information about the data structures or states that are successfully created or removed, along with errors counts:

```
enable
 show mpls traffic-eng nsr counters

State: Active

Bulk sync
  Last bulk sync was successful (entries sent: 24)
  initiated: 1

Send timer
  started: 7

Checkpoint Messages (Items) Sent
```

```
  Succeeded:      13  (101)
    Acks accepted:13  (101)
    Acks ignored:     (0)
    Nacks:        0  (0)
  Failed:         0  (0)
  Buffer alloc:   13
  Buffer freed:   13

ISSU:
  Checkpoint Messages Transformed:
    On Send:
      Succeeded:         13
      Failed:            0
      Transformations:   0
    On Recv:
      Succeeded:         0
      Failed:            0
      Transformations:   0

  Negotiation:
    Started:              1
    Finished:             1
    Failed to Start:      0
    Messages:
      Sent:
        Send succeeded:   5
        Send failed:      0
        Buffer allocated:        5
        Buffer freed:            0
        Buffer alloc failed:     0
      Received:
        Succeeded:        7
        Failed:           0
        Buffer freed:     7
  Init:
    Succeeded:            1
    Failed:               0

  Session Registration:
    Succeeded:            0
    Failed:               0

  Session Unregistration:
    Succeeded:            0
    Failed:               0

Errors:
  None
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Databases

The following example shows how to verify information pertaining to the write and read databases supporting MPLS TE NSR. The write and read databases store the data that is used for recovering TE state on a standby device after Stateful Switchover (SSO):

```
Device# show mpls traffic-eng nsr database if-autotun
IF_AUTOTUN WRITE DB

  Header:
    State: Checkpointed     Action: Add
```

```
    Seq #: 14               Flags: 0x0
  Data:
    te_nsr_seq_num: 28
    Tunnel ID: 100 (if_handle: 25), prot_if_handle: 3
    template_unit: n/a, dest: 10.2.0.1, flags=0x0

IF_AUTOTUN READ DB

Device# show mpls traffic-eng nsr database lsp-ac ?
  |  Output modifiers
  <cr>

Device# show mpls traffic-eng nsr database lsp-ac
 LM Tunnel WRITE DB:

Tun ID: 1   LSP ID: 11   (P2MP)
  SubGrp ID:   1
  SubGrp Orig: 10.1.0.1
  Dest:   10.2.0.1
  Sender: 10.1.0.1     Ext. Tun ID: 10.1.0.1
  Header:
    State: Checkpointed     Action: Add
    Seq #: 7                Flags: 0x0
    TE NSR Seq #: 14

LM Tunnel READ DB:

Device# show mpls traffic-eng nsr database internal

Write DB:
                    Checkpointed
   Entry Type       or Ack-Pending  Send-Pending
     PCALC Node                  0             0
     PCALC Link                  0             0
     PCALC Auto-Mes              0             0
     PCALC SRLG                  0             0
     lm_tunnel_t                 0             0
     NSR LSP FRR                 0             0
     nsr_if_autotun              0             0
     nsr_tspvif_set              0             0
     nsr_slsp_head               0             0

Read DB:
   Entry Type        Checkpointed
     PCALC Node                 5
     PCALC Link                12
     PCALC Auto-Mesh            0
     PCALC SRLG                 0
     lm_tunnel_t                5
     NSR LSP FRR                0
     nsr_if_autotun             0
     nsr_tspvif_setup           3
     nsr_slsp_head              5

TE NSR Sequence Bulk Sync List:
Entries: 0; next avail seq num: 132

TE NSR Sequence State Creation List:
Entries: 30; next expected seq num: 132
   Seq Num: 7  EntryPtr: 0x5A03B208
     Type: PCALC Node  Action: Add  Bundle Seq #: 1
   Seq Num: 8  EntryPtr: 0x5A0B8B38
     Type: PCALC Link  Action: Add  Bundle Seq #: 2
   Seq Num: 9  EntryPtr: 0x5A0B8DA0
```

```
    Type: PCALC Link  Action: Add  Bundle Seq #: 2
  Seq Num: 10  EntryPtr: 0x59FF1BB0
    Type: PCALC Node  Action: Add  Bundle Seq #: 1
  Seq Num: 11  EntryPtr: 0x5A0B9008
    Type: PCALC Link  Action: Add  Bundle Seq #: 2
  Seq Num: 32  EntryPtr: 0x586F2A50
    Type: PCALC Node  Action: Add  Bundle Seq #: 4
  Seq Num: 33  EntryPtr: 0x5949FC58
    Type: PCALC Link  Action: Add  Bundle Seq #: 5
  Seq Num: 34  EntryPtr: 0x5949FEC0
    Type: PCALC Link  Action: Add  Bundle Seq #: 5
  Seq Num: 60  EntryPtr: 0x5725BC30
    Type: lm_tunnel_t  Action: Add  Bundle Seq #: 12
  Seq Num: 61  EntryPtr: 0x5725BE00
    Type: nsr_tspvif_setup  Action: Add  Bundle Seq #: 12
  Seq Num: 62  EntryPtr: 0x59FC9E80
    Type: nsr_slsp_head  Action: Add  Bundle Seq #: 12
  Seq Num: 79  EntryPtr: 0x59296190
    Type: lm_tunnel_t  Action: Add  Bundle Seq #: 16
  Seq Num: 80  EntryPtr: 0x59296360
    Type: nsr_tspvif_setup  Action: Add  Bundle Seq #: 16
  Seq Num: 81  EntryPtr: 0x571EB7F8
    Type: nsr_slsp_head  Action: Add  Bundle Seq #: 16
  Seq Num: 98  EntryPtr: 0x5A04B770
    Type: lm_tunnel_t  Action: Add  Bundle Seq #: 20
  Seq Num: 99  EntryPtr: 0x59296108
    Type: nsr_tspvif_setup  Action: Add  Bundle Seq #: 20
  Seq Num: 100  EntryPtr: 0x57258670
    Type: nsr_slsp_head  Action: Add  Bundle Seq #: 20
  Seq Num: 101  EntryPtr: 0x5A060348
    Type: lm_tunnel_t  Action: Add  Bundle Seq #: 20
  Seq Num: 102  EntryPtr: 0x5A03B2B0
    Type: nsr_slsp_head  Action: Add  Bundle Seq #: 20
  Seq Num: 103  EntryPtr: 0x5B1F12B0
    Type: lm_tunnel_t  Action: Add  Bundle Seq #: 20
  Seq Num: 104  EntryPtr: 0x5A03B400
    Type: nsr_slsp_head  Action: Add  Bundle Seq #: 20
  Seq Num: 121  EntryPtr: 0x57258358
    Type: PCALC Node  Action: Add  Bundle Seq #: 21
  Seq Num: 122  EntryPtr: 0x59FAF080
    Type: PCALC Link  Action: Add  Bundle Seq #: 22
  Seq Num: 123  EntryPtr: 0x59502AC0
    Type: PCALC Link  Action: Add  Bundle Seq #: 23
  Seq Num: 124  EntryPtr: 0x594AE918
    Type: PCALC Link  Action: Add  Bundle Seq #: 21
  Seq Num: 125  EntryPtr: 0x59502120
    Type: PCALC Link  Action: Add  Bundle Seq #: 23
  Seq Num: 126  EntryPtr: 0x59FAFA20
    Type: PCALC Link  Action: Add  Bundle Seq #: 22
  Seq Num: 129  EntryPtr: 0x59FC9CC0
    Type: PCALC Node  Action: Add  Bundle Seq #: 24
  Seq Num: 130  EntryPtr: 0x5A060518
    Type: PCALC Link  Action: Add  Bundle Seq #: 24
  Seq Num: 131  EntryPtr: 0x59FAFC88
    Type: PCALC Link  Action: Add  Bundle Seq #: 24

Device# show mpls traffic-eng nsr database lsp-frr
LSP-FRR WRITE DB

Tun ID: 1   LSP ID: 10   (P2MP)
  SubGrp ID:   1
  SubGrp Orig: 10.1.0.1
  Dest:   10.2.0.1
  Sender: 10.1.0.1     Ext. Tun ID: 10.1.0.1
```

```
  Header:
    State: Checkpointed     Action: Add
    Seq #: 45               Flags: 0x0
  Data:
    te_nsr_seq_num: 164
    LSP Protected if_num: 3 (Ethernet0/0)
    LSP Next-Hop Info: rrr_id 10.2.0.1, address 10.2.0.1, label 17
    LSP Next-Next-Hop Info: rrr_id 0.0.0.0, address 0.0.0.0, label 16777216
    LSP Hold Priority: 7
    LSP bw_type: any pool
    LSP desired_bit_type: 0x0n    LSP Backup ERO address 10.1.2.2
    LSP advertise_bw: NO

LSP-FRR READ DB

Device# show mpls traffic-eng nsr database lsp-frr filter destination ?
  Hostname or A.B.C.D  IP addr or name of destination (tunnel tail)

Device# show mpls traffic-eng nsr database lsp-frr filter lsp-id ?
  <0-65535>  LSP ID

Device# show mpls traffic-eng nsr database lsp-frr filter source ?
  Hostname or A.B.C.D  IP addr or name of sender (tunnel head)

Device# show mpls traffic-eng nsr database lsp-frr filter tunnel-id ?
  <0-65535>  tunnel ID

Device# show mpls traffic-eng nsr database lsp-head
SLSP_HEAD WRITE DB

  Tun ID: 0  (P2P), lsp_id: 7
  Header:
    State: Checkpointed     Action: Add
    Seq #: 6                Flags: 0x0
  Data:
    te_nsr_seq_num: 18
    bandwidth: 5, thead_flags: 0x1, popt: 1
    feature flags: none
    output_if_num: 11, output_nhop: 10.1.3.2
    backup_output_if_num: 0
    output_tag: 19
    backup_output_tag: 16777218
    RRR path setup info
      Destination: 10.3.0.1, Id: 10.3.0.1 Router Node (ospf) flag:0x0
      IGP: ospf, IGP area: 0, Number of hops: 3, metric: 128
      Hop 0: 10.1.3.2, Id: 10.2.0.1 Router Node (ospf), flag:0x0
      Hop 1: 10.2.3.3, Id: 10.3.0.1 Router Node (ospf), flag:0x0
      Hop 2: 10.3.0.1, Id: 10.3.0.1 Router Node (ospf), flag:0x0

SLSP_HEAD READ DB

Device# show mpls traffic-eng nsr database lsp-head filter destination ?
  Hostname or A.B.C.D  IP addr or name of destination (tunnel tail)

Device# show mpls traffic-eng nsr database lsp-head filter lsp-id ?
  <0-65535>  LSP ID

Device# show mpls traffic-eng nsr database lsp-head filter source ?
  Hostname or A.B.C.D  IP addr or name of sender (tunnel head)

Device# show mpls traffic-eng nsr database lsp-head filter tunnel-id ?
  <0-65535>  tunnel ID

Device# show mpls traffic-eng nsr database pcalc auto-mesh
```

```
  PCALC Auto-Mesh WRITE DB:

  PCALC Auto-Mesh READ DB:

Device# show mpls traffic-eng nsr database pcalc nbr
 PCALC Link WRITE DB:
  Header:
    State: Checkpointed     Action: Add
    Seq #: 4                Flags: 0x0
    TE NSR Seq #: 26
    IGP Id:10.1.2.2     Area:0   Nbr IGP Id:10.1.2.2
    IP:10.1.2.1         Nbr IP:0.0.0.0  Framgment ID:1
    Intf ID    Local:0     Remote:0


 PCALC Link READ DB:

Device# show mpls traffic-eng nsr database pcalc node
 PCALC Node WRITE DB:
  Header:
    State: Checkpointed     Action: Add
    Seq #: 4                Flags: 0x0
    TE NSR Seq #: 25
    Router Id 10.1.0.1
    node_id 1
    num_links 2
    tlvs_len 0
    flags 0x6
    rid_frag_id 0
    bcid_mismatch 0
    incarnation  0

Device# show mpls traffic-eng nsr database pcalc srlg
 PCALC SRLGs WRITE DB:

 PCALC SRLGs READ DB:

Device# show mpls traffic-eng nsr database summary
MPLS-TE Non-Stop-Routing is ENABLED

Write DB Coalescing: INACTIVE
Write DB:
  Send-Pending:     0
  Ack-Pending :     0
  Checkpointed:    35
  Total       :    35

Read DB:
  Total       :     0

Device# show mpls traffic-eng nsr database tun-setup
TSPVIF_SETUP WRITE DB

  Tun ID: 0, lsp_id: 7
  Header:
    State: Checkpointed     Action: Add
    Seq #: 6                Flags: 0x0
  Data:
    te_nsr_seq_num: 17
    Setup Evt: allocating current tspsetup, chkpt_flags: 0x0

TSPVIF_SETUP READ DB
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Out-of-Sync Databases

The following example shows how to verify information pertaining to the out-of-sync databases supporting MPLS TE NSR. The out-of-sync databases indicate the **active and standby RSP** whose states are not in sync with each other:

```
enable
 show mpls traffic-eng nsr oos
  Tunnel: 4000
  Time created: 02/20/13-12:03:13
  Time synced: 02/20/13-12:03:14
  Key:
    Source:                    10.1.0.1
    Destination:               10.2.0.1
    ID:                        4000
    Ext Tun ID:                10.1.0.1
    Instance:                  4
    Slsp p2mp ID:              0
    Slsp p2mp subgroup ID:     0
    Slsp p2mp subgroup origin: 0

  RSVP States:
    Signal:       Unknown
    Fast-Reroute: Disabled
    Delete State: True

  TE States:
    Signal:       Unknown
    Fast-Reroute: Disabled
    Delete State: True

  Update History:
    Total number of updates: 2

      Update Time: 02/20/13-12:03:13
        Client Updating: RSVP
        Update State:
          Signal:       Unknown
          Fast-Reroute: Unknown
          Delete State: True

      Update Time: 02/20/13-12:03:14
        Client Updating: TE
        Update State:
          Signal:       Unknown
          Fast-Reroute: Unknown
          Delete State: True
```

Verifying MPLS Traffic Engineering Nonstop Routing Support Information Summary

The following example shows how to view a summary of MPLS TE NSR information such as the current TE NSR state (standby-hot / recovering / staling / active), recovery time, and the recovery result (success / failure):

```
enable
 show mpls traffic-eng nsr summary
  State:
```

```
Graceful-Restart: Disabled
HA state: Active
Checkpointing: Allowed
Messages:
 Send timer: not running (Interval: 1000 msec)
 Items sent per Interval: 200
 CF buffer size used: 3968
```

Additional References for MPLS Traffic Engineering Nonstop Routing Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Multiprotocol Label Switching High Availability Configuration Guide	Cisco IOS XE Multiprotocol Label Switching High Availability Configuration Guide
MPLS TE commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2205	*Resource Reservation Protocol (RSVP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

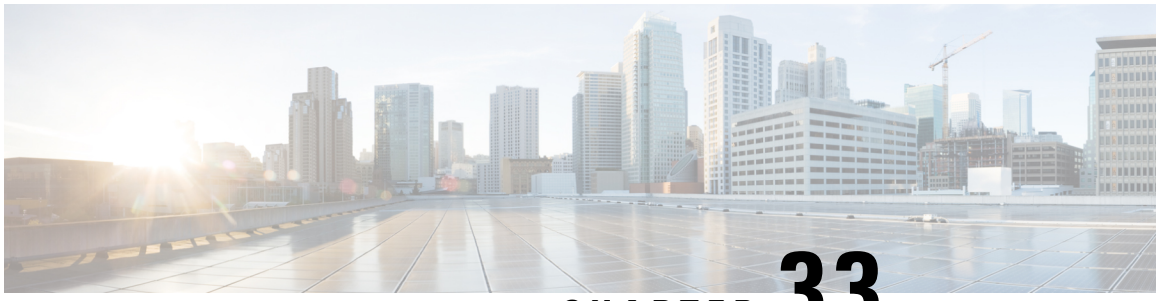
Feature Information for MPLS Traffic Engineering Nonstop Routing Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 90: Feature Information for MPLS Traffic Engineering Nonstop Routing Support

Feature Name	Releases	Feature Information
MPLS Traffic Engineering Nonstop Routing Support	Cisco IOS XE Release 3.10S, 3.13S	The MPLS Traffic Engineering Non-Stop Routing Support feature assists the Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) routing devices to recover from an interruption in service. The MPLS TE Nonstop Routing (NSR) support functionality also defines the checkpoint and recovery scheme for the devices. From Cisco IOS XE 3.13S, support was provided for ASR 903. The following commands were introduced: **mpls traffic-eng nsr** and **show mpls traffic-eng nsr**.



CHAPTER 33

NSR LDP Support

The NSR LDP Support feature allows the Label Distribution Protocol (LDP) to continue to operate across a Router Processor (RP) failure in redundant systems, without losing peer sessions. Before the introduction of nonstop routing (NSR), LDP sessions with peers reset if an RP failover (in a redundant system) or a Cisco In-Service Software Upgrade (ISSU) occurred. When peers reset, traffic is lost while the session is down. Protocol reconvergence occurs after the session is reestablished.

When NSR is enabled, RP failover and Cisco ISSU events are not visible to the peer device, and the LDP sessions that were established prior to failover do not flap. The protocol state learned from the peers persists across an RP failover or Cisco ISSU event and does not need to be relearned.

Prerequisites for NSR LDP Support

The Label Distribution Protocol (LDP) must be up and running on the standby Route Processor (RP) for NSR LDP Support to work.

Information About NSR LDP Support

Roles of the Standby Route Processor and Standby LDP

For the NSR LDP Support feature to work, the Label Distribution Protocol (LDP) must be up and running on the standby Route Processor (RP). The LDP component running on the active RP is called the active LDP, and the LDP component running on the standby RP is called the standby LDP.

When nonstop routing (NSR) is enabled, the standby LDP runs independently from the active LDP, but with the assistance of some software components. The standby LDP maintains LDP session states and database information, ready to take over for the active LDP if the failover occurs.

Standby LDP maintains its local database by querying or receiving notifications of interface status change, configuration changes from the CLI, and checkpoints from the active LDP for other information that is not directly available on the standby RP.

To keep the protocol and session-state information synchronized with the active LDP, the standby LDP depends on TCP to replicate all LDP messages on the active RP to the standby RP. The standby LDP processes all received messages, updates its state, but does not send any responses to its neighbors.

The standby LDP performs the following tasks:

- Processes LDP configuration on startup and during steady state
- Processes active LDP checkpoints of state and session information such as LDP adjacencies, remote addresses, remote bindings, and so forth
- Builds its database of local interfaces
- Processes interface change events
- Receives and processes all LDP messages replicated by TCP
- Updates remote address and label databases

After a switchover and notification that the RP has become active, the standby LDP takes over the role of the active LDP and performs the following tasks:

- Sends hello messages immediately to prevent neighbors from reaching the discovery timeout and bringing down the session
- Retransmits any protocol-level response that has not been sent by the previous active LDP
- Readvertises label bindings
- Refreshes all forwarding entries
- Processes and responds to any LDP message from its neighbor

When the NSR LDP Support feature is disabled, the active LDP performs the following tasks:

- Stops checkpointing to the standby LDP
- Continues to manage all existing sessions

The standby LDP performs the following tasks:

- Cleans up all session-state information
- Reverses to the behavior before NSR is enabled

LDP Operating States

When the NSR LDP Support feature is enabled, the Label Distribution Protocol (LDP) operates in the following states:

Initial State

In the initial state, the active Label Distribution Protocol (LDP) process sets up the standby LDP to be ready to support nonstop routing (NSR). The active LDP performs the following tasks:

- Replicates all TCP sessions used by LDP with the standby LDP
- Synchronizes all existing session-state information with the standby LDP
- Synchronizes the LDP database with the standby LDP

LDP could be in the initial state because of one of these conditions:

- NSR is enabled
- NSR was enabled and the standby Route Processor (RP) starts up (asymmetric startup)
- System boots up and NSR is configured (symmetric startup)

Steady State

In the steady state, the active and standby Label Distribution Protocol (LDP) databases are synchronized. The active and standby LDP process the same LDP messages and update their states independently. The standby LDP is ready to take over the active LDP role in a switchover event.

On the active Route Processor (RP), the active LDP performs the following tasks:

- Continues to manage all existing sessions and checkpoints any significant session event to the standby LDP (such as adjacency delete, session shutdown, timers)
- Notifies the standby LDP of new adjacencies and neighbors

On the standby RP, the standby LDP performs these tasks:

- Processes all received messages but does not send any messages to its neighbor
- Processes checkpoint information from the active LDP
- Manages session keepalive timers but does not bring down the session if a keepalive timer times out

Post Switchover

In the post switchover state, the standby Label Distribution Protocol (LDP) process takes over the active LDP role while the active Route Processor (RP) is reloading.

Supported NSR Scenarios

The NSR LDP Support feature is supported under the following scenarios:

- Route Processor (RP) failover or node failure

  The Label Distribution Protocol (LDP) keeps the session up during an RP or node failover because the LDP adjacency and session-state information between LDP on the active and standby RPs are synchronized. As sessions are created on the active RP, new adjacencies are synchronized to the standby RP. If a standby RP is brought online after sessions are already up (asymmetric startup), LDP synchronizes the existing session-state information from the active to the standby RP.

- Cisco In-Service Software Upgrade (ISSU)

  LDP supports Cisco ISSU negotiation between RPs when a standby RP comes online for the MPLS LDP IGP Synchronization feature. Current Cisco ISSU negotiation is not impacted by NSR. For NSR, LDP negotiates messages specific to NSR, which are checkpointed during initial synchronization (adjacency and session-state information).

How to Configure NSR LDP Support

Enabling NSR LDP Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp nsr**
4. **exit**
5. **show mpls ldp nsr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ldp nsr** **Example:** `Device(config)# mpls ldp nsr`	Enables nonstop routing (NSR) for all Label Distribution Protocol (LDP) sessions for both link and targeted.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show mpls ldp nsr** **Example:** `Device# show mpls ldp nsr`	Displays whether NSR is enabled.

Troubleshooting Tips for NSR LDP Support

Use the **debug mpls ldp nsr** command to enable the display of Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) nonstop routing (NSR) debugging events for all NSR sessions or for the specified peer.

Configuration Examples for NSR LDP Support

Example: NSR LDP Configuration

Additional References for NSR LDP Support

Related Documents

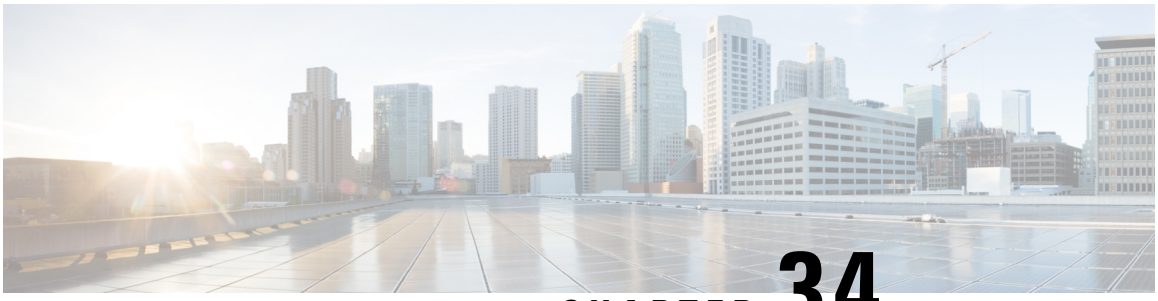
Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
LDP configuration tasks	*MPLS Label Distribution Protocol Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password	http://www.cisco.com/cisco/web/support/index.html

PART IV

MPLS LDP



CHAPTER 34

MPLS Label Distribution Protocol

MPLS Label Distribution Protocol (LDP) enables peer label switch routers (LSRs) in an Multiprotocol Label Switching (MPLS) network to exchange label binding information for supporting hop-by-hop forwarding in an MPLS network. This module explains the concepts related to MPLS LDP and describes how to configure MPLS LDP in a network.

Prerequisites for MPLS Label Distribution Protocol

Label switching on a device requires that Cisco Express Forwarding be enabled on that device.

Information About MPLS Label Distribution Protocol

Introduction to MPLS Label Distribution Protocol

MPLS Label Distribution Protocol (LDP) provides the means for label switch devices (LSRs) to request, distribute, and release label prefix binding information to peer devices in a network. LDP enables LSRs to discover potential peers and to establish LDP sessions with those peers for the purpose of exchanging label binding information.

Multiprotocol Label Switching (MPLS) LDP enables one LSR to inform another LSR of the label bindings it has made. Once a pair of devices communicate the LDP parameters, they establish a label switched path (LSP). MPLS LDP enables LSRs to distribute labels along normally routed paths to support MPLS forwarding. This method of label distribution is also called hop-by-hop forwarding. With IP forwarding, when a packet arrives at a device the device looks at the destination address in the IP header, performs a route lookup, and forwards the packet to the next hop. With MPLS forwarding, when a packet arrives at a device the device looks at the incoming label, looks up the label in a table, and then forwards the packet to the next hop. MPLS LDP is useful for applications that require hop-by-hop forwarding, such as MPLS VPNs.

MPLS Label Distribution Protocol Functional Overview

Cisco Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) provides the building blocks for MPLS-enabled applications, such as MPLS Virtual Private Networks (VPNs).

LDP provides a standard methodology for hop-by-hop, or dynamic label, distribution in an MPLS network by assigning labels to routes that have been chosen by the underlying Interior Gateway Protocol (IGP) routing protocols. The resulting labeled paths, called label switch paths (LIPS), forward label traffic across an MPLS backbone to particular destinations. These capabilities enable service providers to implement MPLS-based IP VPNs and IP+ATM services across multivendor MPLS networks.

Introduction to LDP Sessions

When you enable Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP), the label switch routers (LSRs) send out messages to try to find other LSRs with which they can create LDP sessions. The following sections explain the differences between directly connected LDP sessions and nondirectly connected LDP sessions.

Directly Connected MPLS LDP Sessions

If a label switch router (LSR) is one hop from its neighbor, it is directly connected to its neighbor. The LSR sends out Label Distribution Protocol (LDP) link Hello messages as User Datagram Protocol (UDP) packets to all the devices on the subnet (multicast). A neighboring LSR may respond to the link Hello message, allowing the two devices to establish an LDP session. This is called basic discovery.

To initiate an LDP session between devices, the devices determine which device will take the active role and which device will take the passive role. The device that takes the active role establishes the LDP TCP connection session and initiates the negotiation of the LDP session parameters. To determine the roles, the two devices compare their transport addresses. The device with the higher IP address takes the active role and establishes the session.

After the LDP TCP connection session is established, the LSRs negotiate the session parameters, including the method of label distribution to be used. Two methods are available:

- Downstream Unsolicited: An LSR advertises label mappings to peers without being asked to.
- Downstream on Demand: An LSR advertises label mappings to a peer only when the peer asks for them.

Nondirectly Connected MPLS LDP Sessions

If the label switch router (LSR) is more than one hop from its neighbor, it is nondirectly connected to its neighbor. For these nondirectly connected neighbors, the LSR sends out a targeted Hello message as a User Datagram Protocol (UDP) packet, but as a unicast message specifically addressed to that LSR. The nondirectly connected LSR responds to the Hello message and the two devices begin to establish a Label Distribution Protocol (LDP) session. This is called extended discovery.

A Multiprotocol Label Switching (MPLS) LDP targeted session is a label distribution session between devices that are not directly connected. When you create an MPLS traffic engineering tunnel interface, you need to establish a label distribution session between the tunnel headend and the tailend devices. You establish nondirectly connected MPLS LDP sessions by enabling the transmission of targeted Hello messages.

You can use the **mpls ldp neighbor targeted** command to set up a targeted session when other means of establishing targeted sessions do not apply, such as configuring **mpls ip** on a traffic engineering (TE) tunnel or configuring Any Transport over MPLS (AToM) virtual circuits (VCs). For example, you can use this

command to create a targeted session between directly connected MPLS LSRs when MPLS label forwarding convergence time is an issue.

The **mpls ldp neighbor targeted** command can improve label convergence time for directly connected neighbor LSRs when the links directly connecting them are down. When the links between the neighbor LSRs are up, both the link and targeted Hellos maintain the LDP session. If the links between the neighbor LSRs go down, and there is an alternate route between neighbors, the targeted Hellos would maintain the session, allowing the LSRs to retain labels learned from each other. When a link directly connecting the LSRs comes back up, the LSRs can immediately reinstall labels for forwarding use without having to reestablish their LDP session and exchange labels.

The exchange of targeted Hello messages between two nondirectly connected neighbors can occur in several ways, including the following:

- Device 1 sends targeted Hello messages carrying a response request to Device 2. Device 2 sends targeted Hello messages in response if its configuration permits. In this situation, Device 1 is considered to be active and Device 2 is considered to be passive.
- Device 1 and Device 2 both send targeted Hello messages to each other. Both devices are considered to be active. Both, one, or neither device can also be passive, if they have been configured to respond to requests for targeted Hello messages from each other.

The default behavior of an LSR is to ignore requests from other LSRs that send targeted Hello messages. You can configure an LSR to respond to requests for targeted Hello messages by issuing the **mpls ldp discovery targeted-hello accept** command.

The active LSR mandates the protocol that is used for a targeted session. The passive LSR uses the protocol of the received targeted Hello messages.

Introduction to LDP Label Bindings Label Spaces and LDP Identifiers

A Label Distribution Protocol (LDP) label binding is an association between a destination prefix and a label. The label used in a label binding is allocated from a set of possible labels called a label space.

LDP supports two types of label spaces:

- Interface-specific—An interface-specific label space uses interface resources for labels. For example, label-controlled ATM (LC-ATM) interfaces use virtual path identifiers/virtual circuit identifiers (VPIs/VCIs) for labels. Depending on its configuration, an LDP platform may support zero, one, or more interface-specific label spaces.
- Platform-wide—An LDP platform supports a single platform-wide label space for use by interfaces that can share the same labels. For Cisco platforms, all interface types, except LC-ATM, use the platform-wide label space.

LDP uses a 6-byte quantity called an LDP Identifier (or LDP ID) to name label spaces. The LDP ID is made up of the following components:

- The first four bytes, called the LPD router ID, identify the label switch router (LSR) that owns the label space.
- The last two bytes, called the local label space ID, identify the label space within the LSR. For the platform-wide label space, the last two bytes of the LDP ID are always both 0.

The LDP ID takes the following form:

<LDP router ID> : <local label space ID>

The following are examples of LPD IDs:

- 172.16.0.0:0
- 192.168.0.0:3

The device determines the LDP router ID as follows, if the **mpls ldp router-id** command is not executed,

1. The device examines the IP addresses of all operational interfaces.
2. If these IP addresses include loopback interface addresses, the device selects the largest loopback address as the LDP router ID.
3. Otherwise, the device selects the largest IP address pertaining to an operational interface as the LDP router ID.

The normal (default) method for determining the LDP router ID may result in a router ID that is not usable in certain situations. For example, the device might select an IP address as the LDP router ID that the routing protocol cannot advertise to a neighboring device. The **mpls ldp router-id** command allows you to specify the IP address of an interface as the LDP router ID. Make sure the specified interface is operational so that its IP address can be used as the LDP router ID.

When you issue the **mpls ldp router-id** command without the **force** keyword, the device select selects the IP address of the specified interface (provided that the interface is operational) the next time it is necessary to select an LDP router ID, which is typically the next time the interface is shut down or the address is configured.

When you issue the **mpls ldp router-id** command with the **force** keyword, the effect of the **mpls ldp router-id** command depends on the current state of the specified interface:

- If the interface is up (operational) and if its IP address is not currently the LDP router ID, the LDP router ID changes to the IP address of the interface. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.
- If the interface is down (not operational) when the **mpls ldp router-id** *interface* **force** command is issued, when the interface transitions to up, the LDP router ID changes to the IP address of the interface. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.

How to Configure MPLS Label Distribution Protocol

Enabling Directly Connected LDP Sessions

This procedure explains how to configure Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) sessions between two directly connected devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **mpls ip**
4. **mpls label protocol** [**ldp** | **tdp** | **both**]
5. **interface** *type number*
6. **mpls ip**
7. **exit**
8. **exit**
9. **show mpls interfaces** [*interface*] [**detail**]
10. **show mpls ldp discovery** [**all** | **vrf** *vpn-name*] [**detail**]
11. **show mpls ldp neighbor** [[**vrf** *vpn-name*] [*address* | *interface*] [**detail**] | **all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Configures MPLS hop-by-hop forwarding globally. • The **mpls ip** command is enabled by default; you do not have to specify this command. • Globally enabling MPLS forwarding does not enable it on the device interfaces. You must enable MPLS forwarding on the interfaces as well as for the device.
Step 4	**mpls label protocol** [**ldp**	**tdp**	**both**] **Example:** `Device(config)# mpls label protocol ldp`	Configures the use of LDP on all interfaces. • The keywords that are available depend on the hardware platform. • If you set all interfaces globally to LDP, you can override specific interfaces with either the **tdp** or **both** keyword by specifying the command in interface configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/3/0`	Specifies the interface to be configured and enters interface configuration mode.
Step 6	**mpls ip** **Example:**	Configures MPLS hop-by-hop forwarding on the interface.

	Command or Action	Purpose
	`Device(config-if)# mpls ip`	• You must enable MPLS forwarding on the interfaces as well as for the device.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 9	**show mpls interfaces** [*interface*] [**detail**] **Example:** `Device# show mpls interfaces`	Verifies that the interfaces have been configured to use LDP.
Step 10	**show mpls ldp discovery** [**all** | **vrf** *vpn-name*] [**detail**] **Example:** `Device# show mpls ldp discovery`	Verifies that the interface is up and is sending Discovery Hello messages.
Step 11	**show mpls ldp neighbor** [[**vrf** *vpn-name*] [*address* | *interface*] [**detail**] | **all**] **Example:** `Device# show mpls ldp neighbor`	Displays the status of LDP sessions.

Examples

The following **show mpls interfaces** command verifies that interfaces FastEthernet 0/3/0 and 0/3/1 have been configured to use LDP:

```
Device# show mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
FastEthernet0/3/0      Yes (ldp)     No       No  No     Yes
FastEthernet0/3/1      Yes           No       No  No     Yes
```

The following **show mpls ldp discovery** command verifies that the interface is up and is sending LDP Discovery Hello messages (as opposed to TDP Hello messages):

```
Device# show mpls ldp discovery
Local LDP Identifier:
    172.16.12.1:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/3/0 (ldp): xmit
```

The following example shows that the LDP session between devices was successfully established:

```
Device# show mpls ldp neighbor
Peer LDP Ident: 10.1.1.2:0; Local LDP Ident 10.1.1.1:0
TCP connection: 10.1.1.2.18 - 10.1.1.1.66
State: Oper; Msgs sent/rcvd: 12/11; Downstream
Up time: 00:00:10
LDP discovery sources:
FastEthernet0/1/0, Src IP addr: 10.20.10.2
Addresses bound to peer LDP Ident:
10.1.1.2    10.20.20.1    10.20.10.2
```

Establishing Nondirectly Connected MPLS LDP Sessions

This section explains how to configure nondirectly connected MPLS Label Distribution Protocol (LDP) sessions, which enable you to establish an LDP session between devices that are not directly connected.

Before you begin

- Multiprotocol Label Switching (MPLS) requires Cisco Express Forwarding.
- You must configure the devices at both ends of the tunnel to be active or enable one device to be passive with the **mpls ldp discovery targeted-hello accept** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol** [**ldp** | **tdp** | **both**]
5. **interface tunnel** *number*
6. **tunnel destination** *ip-address*
7. **mpls ip**
8. **exit**
9. **exit**
10. **show mpls ldp discovery** [**all** | **vrf** *vpn-name*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Configures MPLS hop-by-hop forwarding globally. • The **mpls ip** command is enabled by default; you do not have to specify this command. • Globally enabling MPLS forwarding does not enable it on the device interfaces. You must enable MPLS forwarding on the interfaces as well as for the device.
Step 4	**mpls label protocol** [**ldp** | **tdp** | **both**] **Example:** `Device(config)# mpls label protocol ldp`	Configures the use of LDP on all interfaces. • The keywords that are available depend on the hardware platform. • If you set all interfaces globally to LDP, you can override specific interfaces with either the **tdp** or **both** keyword by specifying the command in interface configuration mode.
Step 5	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 6	**tunnel destination** *ip-address* **Example:** `Device(config-if)# tunnel destination 172.16.1.1`	Assigns an IP address to the tunnel interface.
Step 7	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Configures MPLS hop-by-hop forwarding on the interface. • You must enable MPLS forwarding on the interfaces as well as for the device.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 10	**show mpls ldp discovery** [**all** | **vrf** *vpn-name*] [**detail**] **Example:** `Device# show mpls ldp discovery`	Verifies that the interface is up and is sending Discovery Hello messages.

Examples

The following example shows the output of the **show mpls ldp discovery** command for a nondirectly connected LDP session:

```
Device# show mpls ldp discovery
Local LDP Identifier:
          172.16.0.0:0
Discovery Sources:
Interfaces:
POS1/2/0 (ldp): xmit/recv
LDP Id: 172.31.255.255:0
Tunnel1 (ldp): Targeted -> 192.168.255.255
Targeted Hellos:
172.16.0.0 -> 192.168.255.255 (ldp): active, xmit/recv
LDP Id: 192.168.255.255:0
172.16.0.0 -> 192.168.0.0 (ldp): passive, xmit/recv
LDP Id: 192.168.0.0:0
```

This command output indicates that:

- The local label switch router (LSR) (172.16.0.0) sent LDP link Hello messages on interface POS1/2/0 and discovered neighbor 172.31.255.255.
- The local LSR sent LDP targeted Hello messages associated with interface Tunnel1 to target 192.168.255.255. The LSR was configured to use LDP.
- The local LSR is active for targeted discovery activity with 192.168.255.255; this means that the targeted Hello messages it sends to 192.168.255.255 carry a response request. The local LSR was configured to have an LDP session with the nondirectly connected LSR 192.168.255.255.
- The local LSR is not passive from the discovery activity with 192.168.255.255 for one of the following reasons:
  - The targeted Hello messages it receives from 192.168.255.255 do not carry a response request.
  - The local LSR has not been configured to respond to such requests.
- The local LSR sent Tag Distribution Protocol (TDP) directed Hello messages to the target LSR 192.168.0.0. This LSR uses TDP because the Hello messages received from the target LSR 192.168.0.0 were TDP directed Hello messages.
- The local LSR is passive in discovery activity with LSR 192.168.0.0. This means that the directed Hello messages it receives from LSR 192.168.0.0 carry a response request and that the local LSR has been configured with the **mpls ldp discovery targeted-hello accept** command to respond to such requests from LSR 192.168.0.0.
- The local LSR is not active in discovery activity with LSR 192.168.0.0, because no application that requires an LDP session with LSR 192.168.0.0 has been configured on the local LSR.

Saving Configurations MPLS Tag Switching Commands

In releases prior to Cisco IOS Release 12.4(2)T, some Multiprotocol Label Switching (MPLS) commands had both a tag-switching version and an MPLS version. For example, the two commands **tag-switching ip**

and **mpls ip** were the same. To support backward compatibility, the tag-switching form of the command was written to the saved configuration.

Starting in Cisco IOS Release 12.4(2)T, the MPLS form of the command is written to the saved configuration.

For example, if an ATM interface is configured using the following commands, which have both a tag-switching form and an MPLS form:

```
Device(config)# interface ATM 3/0
Device(config-if)# ip unnumbered Loopback0
Device(config-if)# tag-switching ip
Device(config-if)# mpls label protocol ldp
```

After you enter these commands and save this configuration or display the running configuration with the **show running-config** command, the commands saved or displayed appear as follows:

```
interface ATM 3/0
ip unnumbered Loopback0
mpls ip
mpls label protocol ldp
```

Specifying the LDP Router ID

The **mpls ldp router-id** command allows you to establish the IP address of an interface as the LDP router ID.

The following steps describe the normal process for determining the LDP router ID:

1. The device considers all the IP addresses of all operational interfaces.
2. If these addresses include loopback interface addresses, the device selects the largest loopback address. Configuring a loopback address helps ensure a stable LDP ID for the device, because the state of loopback addresses does not change. However, configuring a loopback interface and IP address on each device is not required.

The loopback IP address does not become the router ID of the local LDP ID under the following circumstances:

- • If the loopback interface has been explicitly shut down.
  • If the **mpls ldp router-id** command specifies that a different interface should be used as the LDP router ID.

If you use a loopback interface, make sure that the IP address for the loopback interface is configured with a /32 network mask. In addition, make sure that the routing protocol in use is configured to advertise the corresponding /32 network.

1. Otherwise, the device selects the largest interface address.

The device might select a router ID that is not usable in certain situations. For example, the device might select an IP address that the routing protocol cannot advertise to a neighboring device.

The device implements the router ID the next time it is necessary to select an LDP router ID. The effect of the command is delayed until the next time it is necessary to select an LDP router ID, which is typically the next time the interface is shut down or the address is deconfigured.

If you use the **force** keyword with the **mpls ldp router-id** command, the router ID takes effect more quickly. However, implementing the router ID depends on the current state of the specified interface:

- If the interface is up (operational) and its IP address is not currently the LDP router ID, the LDP router ID is forcibly changed to the IP address of the interface. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts Multiprotocol Label Switching (MPLS) forwarding activity associated with the bindings.
- If the interface is down, the LDP router ID is forcibly changed to the IP address of the interface when the interface transitions to up. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.

Before you begin

Make sure the specified interface is operational before assigning it as the Label Distribution Protocol (LDP) router ID.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol** [**ldp** | **tdp** | **both**]
5. **mpls ldp router-id** *interface* [**force**]
6. **exit**
7. **show mpls ldp discovery** [**all** | **detail** | **vrf** *vpn-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Configures MPLS hop-by-hop forwarding globally. • The **mpls ip** command is enabled by default; you do not have to specify this command. • Globally enabling MPLS forwarding does not enable it on the device interfaces. You must enable MPLS forwarding on the interfaces as well as for the device.
Step 4	**mpls label protocol** [**ldp**	**tdp**	**both**] **Example:**	Configures the use of LDP on all interfaces. • The keywords that are available depend on the hardware platform.

	Command or Action	Purpose
	`Device(config)# mpls label protocol ldp`	• If you set all interfaces globally to LDP, you can override specific interfaces with either the **tdp** or **both** keyword by specifying the command in interface configuration mode.
Step 5	**mpls ldp router-id** *interface* [**force**] **Example:** `Device(config)# mpls ldp router-id pos 2/0/0`	Specifies the preferred interface for determining the LDP router ID.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 7	**show mpls ldp discovery** [**all** | **detail** | **vrf** *vpn-name*] **Example:** `Device# show mpls ldp discovery`	Displays the LDP identifier for the local device.

Example

The following example assigns interface pos 2/0/0 as the LDP router ID:

```
Device> enable
Device# configure terminal
Device(config)# mpls ip
Device(config)# mpls label protocol ldp
Device(config)# mpls ldp router-id pos 2/0/0 force
```

The following example displays the LDP router ID (10.15.15.15):

```
Device# show mpls ldp discovery
Local LDP Identifier:
    10.15.15.15:0
Discovery Sources:
    Interfaces:
        FastEthernet0/3/0 (ldp): xmit/recv
            LDP Id: 10.14.14.14:0
```

Preserving QoS Settings with MPLS LDP Explicit Null

Normally, the Label Distribution Protocol (LDP) advertises an Implicit Null label for directly connected routes. The Implicit Null label causes the second last (penultimate) label switched router (LSR) to remove the Multiprotocol Label Switching (MPLS) header from the packet. In this case, the penultimate LSR and the last LSR do not have access to the quality of service (QoS) values that the packet carried before the MPLS header was removed. To preserve the QoS values, you can configure the LSR to advertise an explicit NULL

label (a label value of zero). The LSR at the penultimate hop forwards MPLS packets with a NULL label instead of forwarding IP packets.

Note An explicit NULL label is not needed when the penultimate hop receives MPLS packets with a label stack that contains at least two labels and penultimate hop popping is performed. In that case, the inner label can still carry the QoS value needed by the penultimate and edge LSR to implement their QoS policy.

When you issue the **mpls ldp explicit-null** command, Explicit Null is advertised in place of Implicit Null for directly connected prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol** [**ldp** | **tdp** | **both**]
5. **interface** *type number*
6. **mpls ip**
7. **exit**
8. **mpls ldp explicit-null** [**for** *prefix-acl* | **to** *peer-acl* | **for** *prefix-acl* **to** *peer-acl*]
9. **exit**
10. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | *next-hop address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vpn-name* [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Configures MPLS hop-by-hop forwarding globally. • The **mpls ip** command is enabled by default; you do not have to specify this command. • Globally enabling MPLS forwarding does not enable it on the device interfaces. You must enable MPLS forwarding on the interfaces as well as for the device.
Step 4	**mpls label protocol** [**ldp**	**tdp**	**both**]	Configures the use of LDP on all interfaces.

	Command or Action	Purpose
	Example: `Device(config)# mpls label protocol ldp`	• The keywords that are available depend on the hardware platform. • If you set all interfaces globally to LDP, you can override specific interfaces with either the **tdp** or **both** keyword by specifying the command in interface configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface atm 2/2/0`	Specifies the interface to be configured and enters interface configuration mode.
Step 6	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Configures MPLS hop-by-hop forwarding on the interface. • You must enable MPLS forwarding on the interfaces as well as for the device.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 8	**mpls ldp explicit-null** [**for** *prefix-acl* | **to** *peer-acl* | **for** *prefix-acl* **to** *peer-acl*] **Example:** `Device(config)# mpls ldp explicit-null`	Advertises an Explicit Null label in situations where it would normally advertise an Implicit Null label.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enter privileged EXEC mode.
Step 10	**show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | *next-hop address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vpn-name* [**detail**] **Example:** `Device# show mpls forwarding-table`	Verifies that MPLS packets are forwarded with an explicit-null label (value of 0).

Examples

Enabling explicit-null on an egress LSR causes that LSR to advertise the explicit-null label to all adjacent MPLS devices.

```
Device# configure terminal
Device(config)# mpls ldp explicit-null
```

If you issue the **show mpls forwarding-table** command on an adjacent device, the output shows that MPLS packets are forwarded with an explicit-null label (value of 0). In the following example, the second column shows that entries have outgoing labels of 0, where once they were marked “Pop label”.

```
Device# show mpls forwarding-table

Local  Outgoing    Prefix            Bytes label Outgoing   Next Hop
label  label or VC or Tunnel Id      switched   interface
19     Pop tag     10.12.12.12/32    0          Fa2/1/0    172.16.0.1
22     0           10.14.14.14/32    0          Fa2/0/0    192.168.0.2
23     0           172.24.24.24/32   0          Fa2/0/0    192.168.0.2
24     0           192.168.0.0/8     0          Fa2/0/0    192.168.0.2
25     0           10.15.15.15/32    0          Fa2/0/0    192.168.0.2
26     0           172.16.0.0/8      0          Fa2/0/0    192.168.0.2
27     25          10.16.16.16/32    0          Fa2/0/0    192.168.0.22
28     0           10.34.34.34/32    0          Fa2/0/0    192.168.0.2
```

Enabling explicit-null and specifying the **for** keyword with a standard access control list (ACL) changes all adjacent MPLS devices' tables to swap an explicit-null label for only those entries specified in the access-list. In the following example, an access-list is created that contains the 10.24.24.24/32 entry. Explicit null is configured and the access list is specified.

```
Device# configure terminal
Device(config)# mpls label protocol ldp
Device(config)# access-list 24 permit host 10.24.24.24
Device(config)# mpls ldp explicit-null for 24
```

If you issue the **show mpls forwarding-table** command on an adjacent device, the output shows that the only the outgoing labels for the addresses specified (172.24.24.24/32) change from Pop label to 0. All other Pop label outgoing labels remain the same.

```
Device# show mpls forwarding-table

Local  Outgoing    Prefix            Bytes label Outgoing   Next Hop
label  label or VC or Tunnel Id      switched   interface
19     Pop tag     10.12.12.12/32    0          Fa2/1/0    172.16.0.1
22     0           10.14.14.14/32    0          Fa2/0/0    192.168.0.2
23     0           172.24.24.24/32   0          Fa2/0/0    192.168.0.2
24     0           192.168.0.0/8     0          Fa2/0/0    192.168.0.2
25     0           10.15.15.15/32    0          Fa2/0/0    192.168.0.2
26     0           172.16.0.0/8      0          Fa2/0/0    192.168.0.2
27     25          10.16.16.16/32    0          Fa2/0/0    192.168.0.22
28     0           10.34.34.34/32    0          Fa2/0/0    192.168.0.2
```

Enabling explicit null and adding the**to** keyword and an access list enables you to advertise explicit-null labels to only those adjacent devices specified in the access-list. To advertise explicit-null to a particular device, you must specify the device's LDP ID in the access-list.

In the following example, an access-list contains the 10.15.15.15/32 entry, which is the LDP ID of an adjacent MPLS device. The device that is configured with explicit null advertises explicit-null labels only to that adjacent device.

```
Device# show mpls ldp discovery
```

```
Local LDP Identifier:
    10.15.15.15:0
Discovery Sources:
    Interfaces:
        FastEthernet2/0/0(ldp): xmit/recv
            TDP Id: 10.14.14.14:0
Device# configure terminal
Device(config)# mpls label protocol ldp
Device(config)# access-list 15 permit host 10.15.15.15
Device(config)# mpls ldp explicit-null to 15
```

If you issue the **show mpls forwarding-table** command, the output shows that explicit null labels are going only to the device specified in the access list.

```
Device# show mpls forwarding-table

Local  Outgoing    Prefix            Bytes label Outgoing   Next Hop
label  label or VC or Tunnel Id      switched   interface
19     Pop tag     10.12.12.12/32    0          Fa2/1/0    172.16.0.1
22     0           10.14.14.14/32    0          Fa2/0/0    192.168.0.2
23     0           172.24.24.24/32   0          Fa2/0/0    192.168.0.2
24     0           192.168.0.0/8     0          Fa2/0/0    192.168.0.2
25     0           10.15.15.15/32    0          Fa2/0/0    192.168.0.2
26     0           172.16.0.0/8      0          Fa2/0/0    192.168.0.2
27     25          10.16.16.16/32    0          Fa2/0/0    192.168.0.22
28     0           10.34.34.34/32    0          Fa2/0/0    192.168.0.2
```

Enabling explicit-null with both the **for** and **to** keywords enables you to specify which routes to advertise with explicit-null labels and to which adjacent devices to advertise these explicit-null labels.

```
Device# show access 15

Standard IP access list 15
     permit 10.15.15.15 (7 matches)
Device# show access 24

Standard IP access list 24
     permit 10.24.24.24 (11 matches)
Device# configure terminal
Device(config)# mpls label protocol ldp
Device(config)# mpls ldp explicit-null for 24 to 15
```

If you issue the **show mpls forwarding-table** command, the output shows that it receives explicit null labels for 10.24.24.24/32.

```
Device# show mpls forwarding-table

Local  Outgoing    Prefix            Bytes label Outgoing      Next Hop
label  label or VC or Tunnel Id      switched   interface
17     0  <---     10.24.24.24/32     0          Fe2/0/0        172.16.0.1
20     Pop tag     172.16.0.0/8       0          Fe2/0/0        172.16.0.1
21     20          10.12.12.12/32     0          Fe2/0/0        172.16.0.1
22     16          10.0.0.0/8         0          Fe2/0/0        172.16.0.1
23     21          10.13.13.13/32     0          Fe2/0/0        172.16.0.1
25     Pop tag     10.14.14.14/32     0          Fe2/0/0        172.16.0.1
27     Pop tag     192.168.0.0/8      0          Fe2/0/0        172.16.0.1
28     25          10.16.16.16/32     0          Fe2/0/0        172.16.0.1
29     Pop tag     192.168.34.34/32   0          Fe2/0/0        172.16.0.1
```

Protecting Data Between LDP Peers with MD5 Authentication

You can enable authentication between two Label Distribution Protocol (LDP) peers, which verifies each segment sent on the TCP connection between the peers. You must configure authentication on both LDP peers using the same password; otherwise, the peer session is not established.

Authentication uses the Message Digest 5 (MD5) algorithm to verify the integrity of the communication and authenticate the origin of the message.

To enable authentication, issue the **mpls ldp neighbor password** command. This causes the device to generate an MD5 digest for every segment sent on the TCP connection and check the MD5 digest for every segment received from the TCP connection.

When you configure a password for an LDP neighbor, the device tears down existing LDP sessions and establishes new sessions with the neighbor.

If a device has a password configured for a neighbor, but the neighboring device does not have a password configured, a message such as the following appears on the console who has a password configured while the two devices attempt to establish an LDP session. The LDP session is not established.

%TCP-6-BADAUTH: No MD5 digest from [peer's IP address](11003) to [local device's IP address](646)

Similarly, if the two devices have different passwords configured, a message such as the following appears on the console. The LDP session is not established.

%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address](11004) to [local device's IP address](646)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol** [**ldp** | **tdp** | **both**]
5. **mpls ldp neighbor** [**vrf** *vpn-name*] *ip-address* [**password** [**0**-**7**] *password-string*]
6. **exit**
7. **show mpls ldp neighbor** [[**vrf** *vpn-name*] [*address* | *interface*] [**detail**] | **all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:**	Configures MPLS hop-by-hop forwarding globally.

	Command or Action	Purpose
	`Device(config)# mpls ip`	• The **mpls ip** command is enabled by default; you do not have to specify this command. • Globally enabling MPLS forwarding does not enable it on the device interfaces. You must enable MPLS forwarding on the interfaces as well as for the device.
Step 4	**mpls label protocol** [**ldp** | **tdp** | **both**] **Example:** `Device(config)# mpls label protocol ldp`	Configures the use of LDP on all interfaces. • The keywords that are available depend on the hardware platform. • If you set all interfaces globally to LDP, you can override specific interfaces with either the **tdp** or **both** keyword by specifying the command in interface configuration mode.
Step 5	**mpls ldp neighbor** [**vrf** *vpn-name*] *ip-address* [**password** [**0**-**7**] *password-string*] **Example:** `Device(config)# mpls ldp neighbor 172.27.0.15 password onethirty9`	Specifies authentication between two LDP peers.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 7	**show mpls ldp neighbor** [[**vrf** *vpn-name*] [*address* | *interface*] [**detail**] | **all**] **Example:** `Device# show mpls ldp neighbor detail`	Displays the status of LDP sessions. If the passwords have been set on both LDP peers and the passwords match, the **show mpls ldp neighbor** command displays that the LDP session was successfully established.

Examples

The following example configures a device with the password cisco:

```
Device> enable
Device# configure terminal
Device(config)# mpls ip
Device(config)# mpls label protocol ldp
Device(config)# mpls ldp neighbor 10.1.1.1 password cisco
Device(config)# exit
```

The following example shows that the LDP session between devices was successfully established:

```
Device# show mpls ldp neighbor
```

```
Peer LDP Ident: 10.1.1.2:0; Local LDP Ident 10.1.1.1:0
TCP connection: 10.1.1.2.11118 - 10.1.1.1.646
State: Oper; Msgs sent/rcvd: 12/11; Downstream
Up time: 00:00:10
LDP discovery sources:
FastEthernet1/0/0, Src IP addr: 10.20.10.2
Addresses bound to peer LDP Ident:
10.1.1.2    10.20.20.1    10.20.10.2
```

The following **show mpls ldp neighbor detail** command shows that MD5 is used for the LDP session.

```
Device# show mpls ldp neighbor 10.0.0.21 detail

    Peer LDP Ident: 10.0.0.21:0; Local LDP Ident 10.0.0.22:0
        TCP connection: 10.0.0.21.646 - 10.0.0.22.14709; MD5 on
        State: Oper; Msgs sent/rcvd: 1020/1019; Downstream; Last TIB rev sent 2034
        Up time: 00:00:39; UID: 3; Peer Id 1;
        LDP discovery sources:
          FastEthernet1/1/0; Src IP addr: 172.16.1.1
            holdtime: 15000 ms, hello interval: 5000 ms
        Addresses bound to peer LDP Ident:
          10.0.0.21       10.0.38.28       10.88.88.2      172.16.0.1
          172.16.1.1
        Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
```

Configuration Examples for MPLS Label Distribution Protocol

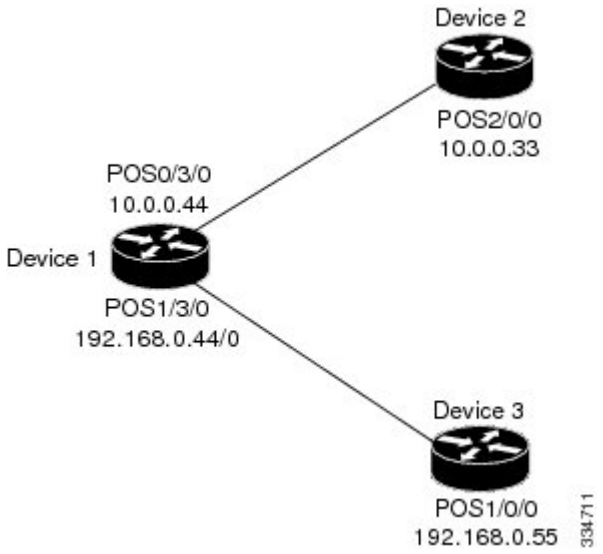
Example: Configuring Directly Connected MPLS LDP Sessions

The figure below shows a sample network for configuring directly connected Label Distribution Protocol (LDP) sessions.

This example configures the following:

- Multiprotocol Label Switching (MPLS) hop-by-hop forwarding for the POS links between Device 1 and Device 2 and between Device 1 and Device 3.
- LDP for label distribution between Device 1 and Device 2.
- LDP for label distribution between Device 1 and Device 3.
- A loopback interface and IP address for each LSR that can be used as the LDP router ID.

Figure 43: Configuration of MPLS LDP



Note The configuration examples below show only the commands related to configuring LDP for Device 1, Device 2, and Device 3 in the sample network shown in the figure above.

Device 1 Configuration

```
ip cef distributed                              !Assumes R1 supports distributed CEF
interface Loopback0                             !Loopback interface for LDP ID.
ip address 172.16.0.11 255.255.255.255
!
interface POS0/3/0
ip address 10.0.0.44 255.0.0.0
mpls ip                                          !Enable hop-by-hop MPLS forwarding
mpls label protocol ldp
!
interface POS1/3/0
ip address 192.168.0.44 255.0.0.0
mpls ip                                          !Enable hop-by-hop MPLS forwarding
mpls label protocol ldp
```

Device 2 Configuration

```
ip cef distributed                               !Assumes R2 supports distributed CEF
!
interface Loopback0                              !Loopback interface for LDP ID.
ip address 172.16.0.22 255.255.255.255
!
interface POS2/0/0
ip address 10.0.0.33 255.0.0.0
mpls ip                                          !Enable hop-by-hop MPLS forwarding
mpls label protocol ldp
```

Device 3 Configuration

```
ip cef                                                  !Assumes R3 does not support dCEF
!
interface Loopback0                                     !Loopback interface for LDP ID.
ip address 172.16.0.33 255.255.255.255
!
interface POS1/0/0
ip address 192.168.0.55 255.0.0.0
mpls ip                                                 !Enable hop-by-hop MPLS forwarding
mpls label protocol ldp
```

The LDP configuration for Device 1 uses the **mpls label protocol ldp** command in interface configuration mode. To specify LDP for all interfaces, use the **mpls label protocol ldp** command in global configuration mode without any interface **mpls label protocol** commands.

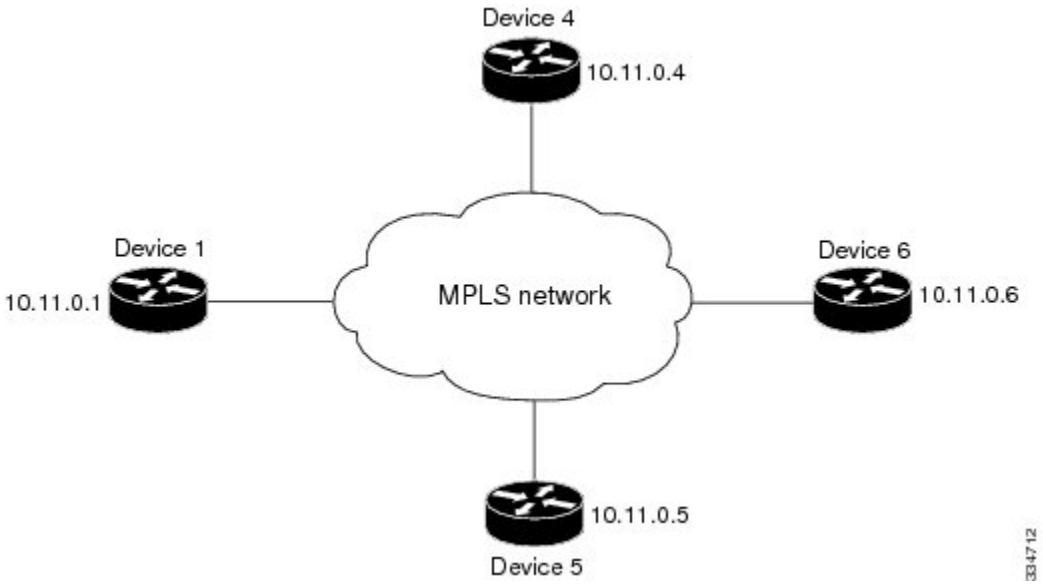
The configuration of Device 2 also uses the **mpls label protocol ldp** command in interface configuration mode. To specify LDP for all interfaces, use the **mpls label protocol ldp** command in global configuration mode without any interface **mpls label protocol** commands.

Configuring the **mpls ip** command on an interface triggers the transmission of discovery Hello messages for the interface.

Example: Establishing Nondirectly Connected MPLS LDP Sessions

The following examples illustrate the configuration of platforms for Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) nondirectly connected sessions using the sample network shown in the figure below. Note that Devices 1, 4, 5, and 6 in this sample network are not directly connected to each other.

Figure 44: Sample Network for Configuring LDP for Targeted Sessions



The configuration example shows the following:

- Targeted sessions between Devices 1 and 4 use LDP. Devices 1 and 4 are both active.
- Targeted sessions between Devices 1 and 6 use LDP. Device 1 is active and Device 6 is passive.

- Targeted sessions between Devices 1 and 5 use LDP. Device 5 is active.

These examples assume that the active ends of the nondirectly connected sessions are associated with tunnel interfaces, such as MPLS traffic engineering tunnels. They show only the commands related to configuring LDP targeted sessions. The examples do not show configuration of the applications that initiate the targeted sessions.

Device 1 Configuration

Tunnel interfaces Tunnel14 and Tunnel16 specify LDP for targeted sessions associated with these interfaces. The targeted session for Device 5 requires LDP. The **mpls label protocol ldp** command in global configuration mode makes it unnecessary to explicitly specify LDP as part of the configuration from the Tunnel14 and Tunnel16.

```
ip cef distributed             !Device1 supports distributed CEF
mpls label protocol ldp       !Use LDP for all interfaces
interface Loopback0            !Loopback interface for LDP ID.
ip address 10.25.0.11 255.255.255.255
interface Tunnel14             !Tunnel to Device 4 requiring label distribution
tunnel destination 10.11.0.4  !Tunnel endpoint is Device 4
mpls ip                        !Enable hop-by-hop forwarding on the interface
interface Tunnel15             !Tunnel to Device 5 requiring label distribution
tunnel destination 10.11.0.5  !Tunnel endpoint is Device 5
mpls label protocol ldp        !Use LDP for session with Device 5
mpls ip                        !Enable hop-by-hop forwarding on the interface
interface Tunnel16             !Tunnel to Device 6 requiring label distribution
tunnel destination 10.11.0.6  !Tunnel endpoint is Device 6
mpls ip                        !Enable hop-by-hop forwarding on the interface
```

Device 4 Configuration

The **mpls label protocol ldp** command in global configuration mode makes it unnecessary to explicitly specify LDP as part of the configuration for the Tunnel41 targeted session with Device 1.

```
ip cef distributed           !Device 4 supports distributed CEF
mpls label protocol ldp       !Use LDP for all interfaces
interface Loopback0           !Loopback interface for LDP ID.
ip address 10.25.0.44 255.255.255.255
interface Tunnel41            !Tunnel to Device 1 requiring label distribution
tunnel destination 10.11.0.1 !Tunnel endpoint is Device 1
mpls ip                        !Enable hop-by-hop forwarding on the interface
```

Device 5 Configuration

Device 5 uses LDP for all targeted sessions. Therefore, its configuration includes the **mpls label protocol ldp** command.

```
ip cef                       !Device 5 supports CEF
mpls label protocol ldp       !Use LDP for all interfaces
interface Loopback0          !Loopback interface for LDP ID.
ip address 10.25.0.55 255.255.255.255
interface Tunnel51             !Tunnel to Device 1 requiring label distribution
tunnel destination 10.11.0.1  !Tunnel endpoint is Device 1
mpls ip                        !Enable hop-by-hop forwarding on the interface
```

Device 6 Configuration

By default, a device cannot be a passive neighbor in targeted sessions. Therefore, Device 1, Device 4, and Device 5 are active neighbors in any targeted sessions. The **mpls ldp discovery targeted-hello accept** command permits Device 6 to be a passive target in targeted sessions with Device 1. Device 6 can also be an active neighbor in targeted sessions, although the example does not include such a configuration.

```
ip cef distributed                          !Device 6 supports distributed CEF
interface Loopback0                         !Loopback interface for LDP ID.
ip address 10.25.0.66 255.255.255.255
mpls ldp discovery targeted-hellos accept from LDP_SOURCES
                                            !Respond to requests for targeted hellos
                                            !from sources permitted by acl LDP_SOURCES
ip access-list standard LDP_SOURCES         !Define acl for targeted hello sources.
permit 10.11.0.1                           !Accept targeted hello request from Device 1.
deny any                                    !Deny requests from other sources.
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Configures LDP on every interface associated with a specified IGP instance.	"MPLS LDP Autoconfiguration" module in the *MPLS Label Distribution Protocol Configuration Guide*
Ensures that LDP is fully established before the IGP path is used for switching.	"MPLS LDP IGP Synchronization" module in the *MPLS Label Distribution Protocol Configuration Guide*
Allows ACLs to control the label bindings that an LSR accepts from its peer LSRs.	"MPLS LDP Inbound Label Binding Filtering" module in the *MPLS Label Distribution Protocol Configuration Guide*
Enables standard, SNMP-based network management of the label switching features.	"MPLS Label Distribution Protocol MIB Version 8 Upgrade" module in the *MPLS Embedded Management and MIBs Configuration Guide*

MIBs

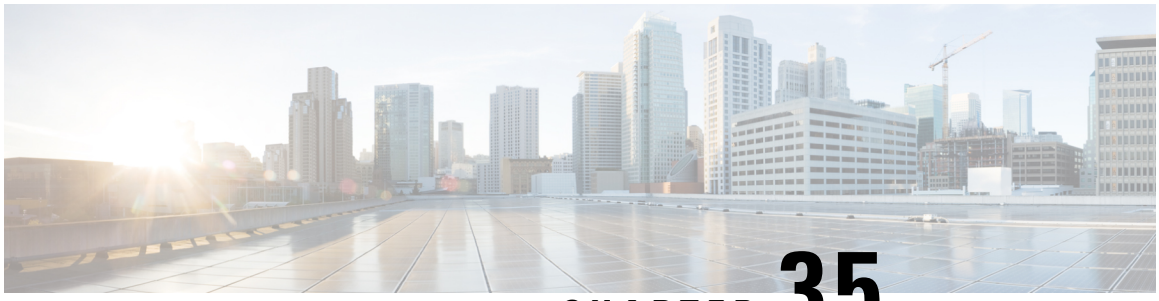
MIB	MIBs Link
• MPLS Label Distribution Protocol MIB (draft-ietf-mpls-ldp-mib-08.txt) • SNMP-VACM-MIB The View-based Access Control Model (ACM) MIB for SNMP	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mib

RFCs

RFC	Title
RFC 3036	*LDP Specification*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 35

MPLS LDP Session Protection

The MPLS LDP Session Protection feature provides faster Label Distribution Protocol (LDP) convergence when a link recovers following an outage. MPLS LDP Session Protection protects an LDP session between directly connected neighbors or an LDP session established for a traffic engineering (TE) tunnel.

Prerequisites for MPLS LDP Session Protection

Label switch routers (LSRs) must be able to respond to Label Distribution Protocol (LDP) targeted hellos. Otherwise, the LSRs cannot establish a targeted adjacency. All devices that participate in MPLS LDP Session Protection must be enabled to respond to targeted hellos. Both neighbor devices must be configured for session protection or one device must be configured for session protection and the other device must be configured to respond to targeted hellos.

Restrictions for MPLS LDP Session Protection

The MPLS LDP Session Protection feature is not supported under the following circumstances:

- With extended access lists
- With LC-ATM devices
- With Tag Distribution Protocol (TDP) sessions

Information About MPLS LDP Session Protection

How MPLS LDP Session Protection Works

MPLS LDP Session Protection maintains Label Distribution Protocol (LDP) bindings when a link fails. MPLS LDP sessions are protected through the use of LDP hello messages. When you enable Multiprotocol Label Switching (MPLS) LDP, the label switch routers (LSRs) send messages to find other LSRs with which they can create LDP sessions.

If the LSR is one hop from its neighbor, it is directly connected to its neighbor. The LSR sends out LDP Hello messages as User Datagram Protocol (UDP) packets to all the devices on the subnet. The hello message is called an LDP Link Hello. A neighboring LSR responds to the hello message, and the two devices begin to establish an LDP session.

If the LSR is more than one hop from its neighbor, it is not directly connected to its neighbor. The LSR sends out a directed hello message as a UDP packet but as a unicast message specifically addressed to that specific LSR. The hello message is called an LDP Targeted Hello. The nondirectly connected LSR responds to the Hello message and the two devices establish an LDP session. (If the path between two LSRs has been traffic engineered and has LDP enabled, the LDP session between them is called a targeted session.)

MPLS LDP Session Protection uses LDP Targeted Hellos to protect LDP sessions. For example, two directly connected devices have LDP enabled and can reach each other through alternate IP routes in the network. An LDP session that exists between two devices is called an LDP Link Hello Adjacency. When MPLS LDP Session Protection is enabled, an LDP Targeted Hello Adjacency is also established for the LDP session. If the link between the two devices fails, the LDP Link Adjacency also fails. However, if the LDP peer is still reachable through IP, the LDP session stays up, because the LDP Targeted Hello Adjacency still exists between the devices. When the directly connected link recovers, the session does not need to be reestablished, and LDP bindings for prefixes do not need to be relearned.

MPLS LDP Session Protection Customization

You can modify MPLS LDP Session Protection by using keywords in the **mpls ldp session protection** command. The following sections explain how to customize the feature:

How Long an LDP Targeted Hello Adjacency Should Be Retained

The default behavior of the **mpls ldp session protection** command allows a Label Distribution Protocol (LDP) Targeted Hello Adjacency to exist indefinitely following the loss of an LDP Link Hello Adjacency. You can issue the **duration** keyword to specify the number of seconds that the LDP Targeted Hello Adjacency is retained after the loss of the LDP Link Hello Adjacency. When the link is lost, a timer starts. If the timer expires, the LDP Targeted Hello Adjacency is removed.

Which Devices Should Have MPLS LDP Session Protection

The default behavior of the **mpls ldp session protection** command allows MPLS LDP Session Protection for all neighbor sessions. You can issue either the **vrf** or **for** keyword to limit the number of neighbor sessions that are protected:

- You can use the **vrf** keyword to select which virtual routing and forwarding (VRF) instance is to be protected if the device is configured with at least one virtual private network (VPN) VRF instance. You

cannot specify more than one VRF with the **mpls ldp session protection** command. To specify multiple VRFs, issue the command multiple times.

- You can create an access list that includes several peer devices. You can specify that access list with the **for** keyword to enable LDP Session Protection for the peer devices in the access control list.

How to Configure MPLS LDP Session Protection

Enabling MPLS LDP Session Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface loopback** *number*
5. **ip address** *prefix mask*
6. **exit**
7. **interface** *type number*
8. **mpls ip**
9. **mpls label protocol** [**ldp** | **tdp** | **both**]
10. **exit**
11. **mpls ldp session protection** [**vrf** *vpn-name*] [**for** *acl*] [**duration** {**infinite** | *seconds*}]
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** [**distributed**] **Example:** `Device(config)# ip cef distributed`	Configures distributed Cisco Express Forwarding or Cisco Express Forwarding.
Step 4	**interface loopback** *number* **Example:**	Configures a loopback interface and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface Loopback 0`	
Step 5	**ip address** *prefix mask* **Example:** `Device(config-if)# ip address 10.25.0.11 255.255.255.255`	Assigns an IP address to the loopback interface.
Step 6	**exit** **Example:** `Device(config-if) exit`	Returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface POS 0/3/0`	Specifies the interface to configure and enters interface configuration mode.
Step 8	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Configures MPLS hop-by-hop forwarding for the specified interface.
Step 9	**mpls label protocol** [**ldp** | **tdp** | **both**] **Example:** `Device(config-if)# mpls label protocol ldp`	Configures the use of LDP on a specific interface or on all interfaces. • The keywords that are available depend on the hardware platform. • If you set all interfaces globally to LDP, you can override specific interfaces with either the **tdp** or **both** keyword by specifying the command in interface configuration mode.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 11	**mpls ldp session protection** [**vrf** *vpn-name*] [**for** *acl*] [**duration** {**infinite** | *seconds*}] **Example:** `Device(config)# mpls ldp session protection`	Enables MPLS LDP session protection. • The **vrf** *vpn-name* keyword and argument protects Label Distribution Protocol (LDP) sessions for a specified virtual routing and forwarding (VRF) interface. • The **for** *acl* keyword and argument specifies a standard IP access control list (ACL) of prefixes to be protected.

	Command or Action	Purpose
		• The **duration** keyword specifies how long the device should retain the LDP Targeted Hello Adjacency following the loss of the LDP Link Hello Adjacency. • The **infinite** keyword specifies that the LDP Targeted Hello Adjacency should be retained forever after a link is lost. • The *seconds* argument specifies the time in seconds that the LDP Targeted Hello Adjacency should be retained after a link is lost. The range is 30 to 2,147,483 seconds. The **mpls ldp session protection** command entered without a keyword protects all LDP sessions.
Step 12	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Troubleshooting Tips

Use the **clear mpls ldp neighbor** command if you need to terminate a Label Distribution Protocol (LDP) session after a link goes down. This is useful for situations where the link needs to be taken out of service or needs to be connected to a different neighbor.

To enable the display of events related to MPLS LDP Session Protection, use the **debug mpls ldp session protection** command.

Verifying MPLS LDP Session Protection

SUMMARY STEPS

1. **enable**
2. **show mpls ldp discovery**
3. **show mpls ldp neighbor**
4. **show mpls ldp neighbor detail**
5. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password, if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show mpls ldp discovery**

Verifies that the output contains the term xmit/recv for the peer device.

Example:

```
Device# show mpls ldp discovery

 Local LDP Identifier:
    10.0.0.5:0
    Discovery Sources:
    Interfaces:
        ATM50/1/0.5 (ldp): xmit/recv
            LDP Id: 10.0.0.1:0
    Targeted Hellos:
        10.0.0.5 -> 10.0.0.3 (ldp): active, xmit/recv
            LDP Id: 10.0.0.3:0
```

Step 3 **show mpls ldp neighbor**

Verifies that the targeted hellos are active.

Example:

```
Device# show mpls ldp neighbor

Peer LDP Ident: 10.0.0.3:0; Local LDP Ident 10.0.0.5:0
TCP connection: 10.0.0.3.646 - 10.0.0.5.11005
State: Oper; Msgs sent/rcvd: 1453/1464; Downstream
Up time: 21:09:56
LDP discovery sources:
 Targeted Hello 10.0.0.5 -> 10.0.0.3, active
Addresses bound to peer LDP Ident:
 10.3.104.3       10.0.0.2        10.0.0.3
```

Step 4 **show mpls ldp neighbor detail**

Verifies that the MPLS LDP Session Protection state is Ready or Protecting. If the second last line of the output shows Incomplete, the Targeted Hello Adjacency is not up yet.

Example:

```
Device# show mpls ldp neighbor detail

    Peer LDP Ident: 10.16.16.16:0; Local LDP Ident 10.15.15.15:0
        TCP connection: 10.16.16.16.11013 - 10.15.15.15.646
        State: Oper; Msgs sent/rcvd: 53/51; Downstream; Last TIB rev sent 74
        Up time: 00:11:32; UID: 1; Peer Id 0;
        LDP discovery sources:
          Targeted Hello 10.15.15.15 -> 10.16.16.16, active, passive;
            holdtime: infinite, hello interval: 10000 ms
        Addresses bound to peer LDP Ident:
          10.0.0.2        10.16.16.16     10.101.101.101 11.0.0.1
        Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
        Clients: Dir Adj Client
        LDP Session Protection enabled, state: Protecting
            duration: infinite
```

Step 5 **exit**

Returns to user EXEC mode.

Example:

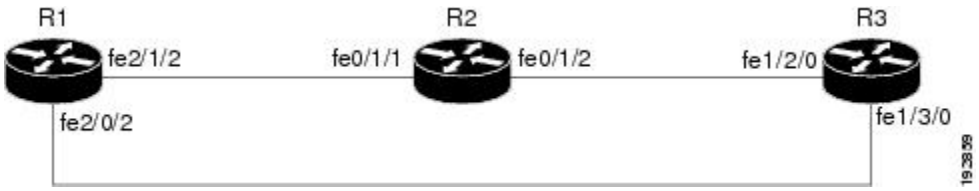
```
Device# exit
Device>
```

Configuration Examples for MPLS LDP Session Protection

Example: Configuring MPLS LDP Session Protection

The figure below shows a sample configuration for MPLS LDP Session Protection.

Figure 45: MPLS LDP Session Protection Example



The following configuration examples for R1, R2, and R3 are based on the figure above.

R1

```
redundancy
 no keepalive-enable
 mode hsa
!
ip cef distributed
no ip domain-lookup
multilink bundle-name both
mpls label protocol ldp
mpls ldp session protection
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.0.0.1 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Multilink4
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 load-interval 30
 ppp multilink
 multilink-group 4
!
interface FastEthernet1/0/0
 ip address 10.3.123.1 255.255.0.0
 no ip directed-broadcast
```

```
!
interface FastEthernet2/0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface FastEthernet2/0/1
 description --  ip address 10.0.0.2 255.255.255.0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface FastEthernet2/0/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet2/1/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet2/2/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 redistribute connected
 network 10.0.0.1 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
ip classless
```

R2

```
redundancy
 no keepalive-enable
 mode hsa
!
ip subnet-zero
ip cef distributed
mpls label protocol ldp
mpls ldp session protection
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.0.0.3 255.255.255.255
 no ip directed-broadcast
!
interface FastEthernet0/1/0
 no ip address
 no ip directed-broadcast
 shutdown
 full-duplex
```

```
!
interface FastEthernet0/1/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 full-duplex
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet0/1/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 ip load-sharing per-packet
 full-duplex
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet0/2/0
 ip address 10.3.123.112 255.255.0.0
 no ip directed-broadcast
!
router ospf 100
 log-adjacency-changes
 redistribute connected
 network 10.0.0.3 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
ip classless
```

R3

```
ip cef distributed
no ip domain-lookup
mpls label range 200 100000 static 16 199
mpls label protocol ldp
no mpls traffic-eng auto-bw timers frequency 0
mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.0.0.5 255.255.255.255
 no ip directed-broadcast
!
interface FastEthernet1/0/0
 no ip address
 no ip directed-broadcast
 shutdown
 half-duplex
!
interface FastEthernet1/2/0
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 full-duplex
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet1/3/0
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 full-duplex
 mpls label protocol ldp
 mpls ip
!
router ospf 100
```

```
 log-adjacency-changes
 redistribute connected
 network 10.0.0.5 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
ip classless
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS LDP	"MPLS Label Distribution Protocol" module in the *MPLS Label Distribution Protocol Configuration Guide*
MPLS LDP IGP synchronization	"MPLS LDP IGP Synchronization" module in the *MPLS Label Distribution Protocol Configuration Guide*
MPLS LDP Autoconfiguration	"MPLS LDP Autoconfiguration" module in the *MPLS Label Distribution Protocol Configuration Guide*

MIBs

MIBs	MIBs Link
MPLS LDP MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mib

RFCs

RFCs	Title
RFC 3036	LDP Specification
RFC 3037	LDP Applicability

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP Session Protection

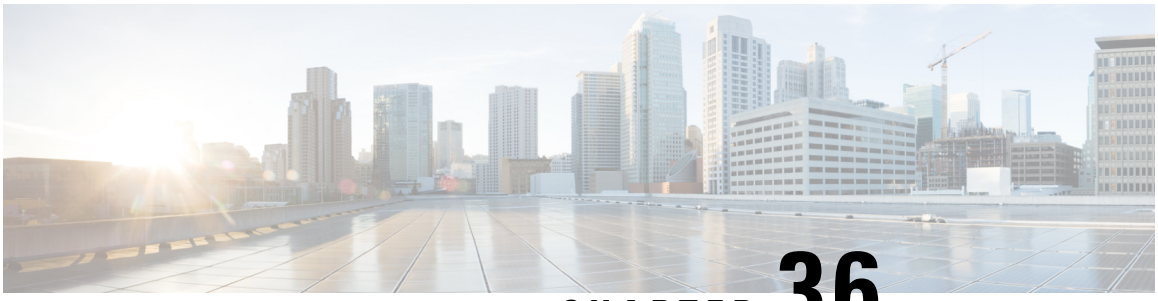
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Information for MPLS LDP Session Protection

Table 91: Feature Information for MPLS LDP Session Protection

Feature Name	Releases	Feature Information
MPLS LDP Session Protection		The MPLS LDP Session Protection feature provides faster Label Distribution Protocol (LDP) convergence when a link recovers following an outage. MPLS LDP Session Protection protects an LDP session between directly connected neighbors or an LDP session established for a traffic engineering (TE) tunnel.



CHAPTER 36

MPLS LDP Autoconfiguration

The MPLS LDP Autoconfiguration feature enables you to globally configure Label Distribution Protocol (LDP) on every interface associated with a specified Interior Gateway Protocol (IGP) instance.

Restrictions for MPLS LDP Autoconfiguration

- If the Label Distribution Protocol (LDP) is disabled globally, the **mpls ldp autoconfig** command fails and generates a console message explaining that LDP must first be enabled globally by using the **mpls ip** global configuration command.
- If the **mpls ldp autoconfig** command is configured for an IGP instance, you cannot enter the **no mpls ip** global configuration command. To disable LDP, you must first issue the **no mpls ldp autoconfig** command.
- For interfaces running Intermediate System-to-Intermediate System (IS-IS) processes, you can enable Multiprotocol Label Switching (MPLS) for each interface, using the router mode command **mpls ldp autoconfig** or the **mpls ldp igp autoconfig** interface configuration command.
- You specify that the default label distribution protocol is LDP for a device or for an interface. Tag Distribution Protocol (TDP) is not supported.
- The MPLS LDP Autoconfiguration feature is not supported on traffic engineering tunnel interfaces.

Information About MPLS LDP Autoconfiguration

MPLS LDP Autoconfiguration on OSPF and IS-IS Interfaces

The MPLS LDP Autoconfiguration feature enables you to globally enable Label Distribution Protocol (LDP) on every interface associated with an Interior Gateway Protocol (IGP) instance. This feature is supported on Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) IGPs. It provides a means to block LDP from being enabled on interfaces that you do not want enabled. The goal of the MPLS LDP Autoconfiguration feature is to make configuration easier, faster, and error free.

You issue the **mpls ldp autoconfig** command to enable LDP on each interface that is running an OSPF or IS-IS process. If you do not want some of the interfaces to have LDP enabled, you can issue the **no mpls ldp igp autoconfig** command on those interfaces.

How to Configure MPLS LDP Autoconfiguration

Configuring MPLS LDP Autoconfiguration with OSPF Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol ldp**
5. **interface** *type number*
6. **ip address** *prefix mask*
7. **mpls ip**
8. **exit**
9. **router ospf** *process-id*
10. **network** *ip-address wildcard-mask* **area** *area-id*
11. **mpls ldp autoconfig** [**area** *area-id*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Globally enables hop-by-hop forwarding.
Step 4	**mpls label protocol ldp** **Example:** `Device(config)# mpls label protocol ldp`	Specifies the Label Distribution Protocol (LDP) as the default protocol.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 6	**ip address** *prefix mask* **Example:** `Device(config-if)# ip address 10.25.0.11 255.255.255.255`	Assigns an IP address to the interface.
Step 7	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables hop-by-hop forwarding on the interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 9	**router ospf** *process-id* **Example:** `Device(config)# router ospf 1`	Enables Open Shortest Path First (OSPF) routing, and enters router configuration mode.
Step 10	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Device(config-router)# network 10.0.0.0 0.255.255.255 area 3`	Defines an interface on which OSPF runs and defines the area ID for that interface.
Step 11	**mpls ldp autoconfig** [**area** *area-id*] **Example:**	Enables the MPLS LDP Autoconfiguration feature to enable LDP on interfaces belonging to the OSPF process.

	Command or Action	Purpose
	`Device(config-router)# mpls ldp autoconfig area 3`	• If no area is specified, the command applies to all interfaces associated with the OSPF process. If an area ID is specified, then only interfaces associated with that OSPF area are enabled with LDP.
Step 12	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Disabling MPLS LDP Autoconfiguration from Selected OSPF Interfaces

When you issue the **mpls ldp autoconfig** command, all the interfaces that belong to an Open Shortest Path First (OSPF) area are enabled for the Label Distribution Protocol (LDP). To remove LDP from some interfaces, use the **no mpls ldp igp autoconfig** command on those interfaces. The following configuration steps show how to disable LDP from some of the interfaces after they were configured with the MPLS LDP Autoconfiguration feature with the **mpls ldp autoconfig** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no mpls ldp igp autoconfig**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface POS 3/0`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**no mpls ldp igp autoconfig** **Example:**	Disables LDP for that interface.

	Command or Action	Purpose
	`Device(config-if)# no mpls ldp igp autoconfig`	
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying MPLS LDP Autoconfiguration with OSPF

SUMMARY STEPS

1. **enable**
2. **show mpls interfaces** [*type number* | **vrf** *vpn-name*] [**all**] [**detail**] [**internal**]
3. **show mpls ldp discovery** [**vrf** *vpn-name* | **all**] [**detail**]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **show mpls interfaces** [*type number* | **vrf** *vpn-name*] [**all**] [**detail**] [**internal**]

Displays the method used to enable the Label Distribution Protocol (LDP) on an interface:

- If LDP is enabled by the **mpls ldp autoconfig** command, the output displays:

Example:

```
IP labeling enabled (ldp):
   IGP config
```

- If LDP is enabled by the **mpls ip** command, the output displays:

Example:

```
IP labeling enabled (ldp):
  Interface config
```

- If LDP is enabled by the **mpls ip** command and the **mpls ldp autoconfig** command, the output displays:

Example:

```
IP labeling enabled (ldp):
  Interface config
  IGP config
```

The following example shows that LDP was enabled on the interface by both the **mpls ip** and **mpls ldp autoconfig** commands:

Example:

Step 3 **show mpls ldp discovery** [**vrf** *vpn-name* | **all**] [**detail**]

Displays how LDP was enabled on the interface. In the following example, LDP was enabled by both the **mpls ip** and **mpls ldp autoconfig** commands:

Example:

Configuring MPLS LDP Autoconfiguration with IS-IS Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *prefix mask*
5. **ip router isis**
6. **exit**
7. **mpls ip**
8. **mpls label protocol ldp**
9. **router isis**
10. **mpls ldp autoconfig** [**level-1** | **level-2**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface POS 0/2`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**ip address** *prefix mask* **Example:** `Device(config-if)# ip address 10.50.72.4 255.0.0.0`	Assigns an IP address to the interface.

	Command or Action	Purpose
Step 5	**ip router isis** **Example:** `Device(config-if)# ip router isis`	Enables the Intermediate System-to-Intermediate System (IS-IS) for IP on the interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 7	**mpls ip** **Example:** `Device(config)# mpls ip`	Globally enables hop-by-hop forwarding.
Step 8	**mpls label protocol ldp** **Example:** `Device(config)# mpls label protocol ldp`	Specifies the Label Distribution Protocol (LDP) as the default protocol.
Step 9	**router isis** **Example:** `Device(config)# router isis`	Enables an IS-IS process on the device and enters router configuration mode.
Step 10	**mpls ldp autoconfig** [**level-1** | **level-2**] **Example:** `Device(config-router)# mpls ldp autoconfig`	Enables the LDP for interfaces that belong to an IS-IS process.
Step 11	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Disabling MPLS LDP Autoconfiguration from Selected IS-IS Interfaces

When you issue the **mpls ldp autoconfig** command, all the interfaces that belong to an Intermediate System-to-Intermediate System (IS-IS) process are enabled for the Label Distribution Protocol (LDP). To remove LDP from some interfaces, use the **no mpls ldp igp autoconfig** command on those interfaces. The following configuration steps show how to disable LDP from some of the interfaces after they were configured with the MPLS LDP Autoconfiguration feature with the **mpls ldp autoconfig** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **no mpls ldp igp autoconfig**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface POS 3/0`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**no mpls ldp igp autoconfig** **Example:** `Device(config-if)# no mpls ldp igp autoconfig`	Disables LDP for that interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying MPLS LDP Autoconfiguration with IS-IS

SUMMARY STEPS

1. **enable**
2. **show isis mpls ldp**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **show isis mpls ldp**

Shows that the Intermediate System-to-Intermediate System (IS-IS) is configured on the interface and that the Label Distribution Protocol (LDP) is enabled:

Example:

```
Device# show isis mpls ldp

Interface: POS0/2; ISIS tag null enabled
 ISIS is UP on interface
 AUTOCONFIG Information :
   LDP enabled: YES
 SYNC Information :
   Required: NO
```

The output shows:

- IS-IS is up.
- LDP is enabled.

If the MPLS LDP Autoconfiguration feature is not enabled on an interface, the output looks like the following:

Example:

```
Interface: Ethernet0; ISIS tag null enabled
 ISIS is UP on interface
 AUTOCONFIG Information :
   LDP enabled: NO
 SYNC Information :
   Required: NO
```

Troubleshooting Tips

You can use the **debug mpls ldp autoconfig** command to display events that are related to the MPLS LDP Autoconfiguration feature.

Configuration Examples for MPLS LDP Autoconfiguration

Example: MPLS LDP Autoconfiguration with OSPF

The following configuration commands enable the Label Distribution Protocol (LDP) for Open Shortest Path First (OSPF) process 1 area 3. The **mpls ldp autoconfig area 3** command and the OSPF **network** commands enable LDP on POS interfaces 0/0, 0/1, and 1/1. The **no mpls ldp igp autoconfig** command on POS interface 1/0 prevents LDP from being enabled on POS interface 1/0, even though OSPF is enabled for that interface.

```
configure terminal
 interface POS 0/0
  ip address 10.0.0.1 255.0.0.0
!
 interface POS 0/1
  ip address 10.0.1.1 255.0.0.1
!
 interface POS 1/1
```

```
  ip address 10.1.1.1 255.255.0.0
!
 interface POS 1/0
  ip address 10.1.0.1 0.1.0.255
  exit
!
 router ospf 1
  network 10.0.0.0 0.0.255.255 area 3
  network 10.1.0.0 0.0.255.255 area 3
  mpls ldp autoconfig area 3
  end
 interface POS 1/0
  no mpls ldp igp autoconfig
```

Example: MPLS LDP Autoconfiguration with IS-IS

The following example shows the configuration of the MPLS LDP Autoconfiguration feature on POS0/2 and 0/3 interfaces, which are running Intermediate System-to-Intermediate System (IS-IS) processes:

```
configure terminal
 interface POS 0/2
  ip address 10.0.0.1 255.0.0.1
  ip router isis
!
 interface POS 0/3
  ip address 10.1.1.1 255.0.1.0
  ip router isis
  exit
mpls ip
mpls label protocol ldp
router isis
mpls ldp autoconfig
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS LDP	"MPLS Label Distribution Protocol" module in the *MPLS Label Distribution Protocol Configuration Guide*
The MPLS LDP IGP Synchronization feature	"MPLS LDP IGP Synchronization" module in the *MPLS Label Distribution Protocol Configuration Guide*
The MPLS LDP Session Protection feature	"MPLS LDP Session Protection" module in the *MPLS Label Distribution Protocol Configuration Guide*
Configuring integrated IS-IS	"Integrated IS-IS Routing Protocol Overview" module in the *IP Routing: ISIS Configuration Guide*

MIBs

MIB	MIBs Link
MPLS LDP MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mib

RFCs

RFC	Title
RFC 3036	LDP Specification
RFC 3037	LDP Applicability

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP Autoconfiguration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 37

MPLS LDP IGP Synchronization

The MPLS LDP IGP Synchronization feature ensures that the Label Distribution Protocol (LDP) is fully established before the Interior Gateway Protocol (IGP) path is used for switching.

Prerequisites for MPLS LDP IGP Synchronization

- This feature is supported only on interfaces running Open Shortest Path First (OSPF) or Intermediate System-to-System (IS-IS) processes.
- This feature works when LDP is enabled on interfaces with either the **mpls ip** or **mpls ldp autoconfig** command.

Restrictions for MPLS LDP IGP Synchronization

- This feature is not supported on tunnel interfaces or LC-ATM interfaces.
- This feature is not supported with interface-local label space or downstream-on-demand (DoD) requests.
- This feature does not support targeted Label Distribution Protocol (LDP) sessions. Therefore, Any Transport over MPLS (AToM) sessions are not supported.
- The Tag Distribution Protocol (TDP) is not supported. You must specify that the default label distribution protocol is LDP for a device or for an interface.

Information About MPLS LDP IGP Synchronization

How MPLS LDP IGP Synchronization Works

Packet loss can occur because the actions of the Interior Gateway Protocol (IGP) and the Label Distribution Protocol (LDP) are not synchronized. Packet loss can occur in the following situations:

- When an IGP adjacency is established, the device begins forwarding packets using the new adjacency before the LDP label exchange completes between the peers on that link.
- If an LDP session closes, the device continues to forward traffic using the link associated with the LDP peer rather than an alternate pathway with a fully synchronized LDP session.

The MPLS LDP IGP Synchronization feature does the following:

- Provides a means to synchronize LDP and IGPs to minimize Multiprotocol Label Switching (MPLS) packet loss.
- Enables you to globally enable LDP IGP synchronization on each interface associated with an IGP Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS) process.
- Provides a means to disable LDP IGP synchronization on interfaces that you do not want enabled.
- Prevents MPLS packet loss due to synchronization conflicts.
- Works when LDP is enabled on interfaces using either the **mpls ip** or **mpls ldp autoconfig** command.

To enable LDP IGP synchronization on each interface that belongs to an OSPF or IS-IS process, enter the **mpls ldp sync** command. If you do not want some of the interfaces to have LDP IGP synchronization enabled, issue the **no mpls ldp igp sync** command on those interfaces.

If 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 **no mpls ldp igp sync holddown** command. If 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.

MPLS LDP IGP Synchronization with Peers

When the MPLS LDP IGP Synchronization feature is enabled on an interface, the Label Distribution Protocol (LDP) determines if any peer connected by the interface is reachable by looking up the peer's transport address in the routing table. If a routing entry (including longest match or default routing entry) for the peer exists, LDP assumes that LDP Interior Gateway Protocol (IGP) synchronization is required for the interface and notifies the IGP to wait for LDP convergence.

LDP IGP synchronization with peers requires that the routing table be accurate for the peer's transport address. If the routing table shows there is a route for the peer's transport address, that route must be able to reach the peer's transport address. However, if the route is a summary route, a default route, or a statically configured route, it may not the correct route for the peer. You must verify that the route in the routing table can reach the peer’s transport address.

When the routing table has an inaccurate route for the peer's transport address, LDP cannot set up a session with the peer, which causes the IGP to wait for LDP convergence unnecessarily for the sync hold-down time.

MPLS LDP IGP Synchronization Delay Timer

The MPLS LDP IGP Synchronization feature provide the option to configure a delay time for Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) and Interior Gateway Protocol (IGP) synchronization on an interface-by-interface basis. If you want to configure a delay time on an interface, use the **mpls ldp igp sync delay** *delay-time* command in interface configuration mode. To remove the delay timer from a specified interface, enter the **no mpls ldp igp sync delay** command. This command sets the delay time to 0 seconds, but leaves MPLS LDP IGP synchronization enabled.

When LDP is fully established and synchronized, LDP checks the delay timer:

- If you configured a delay time, LDP starts the timer. When the timer expires, LDP checks that synchronization is still valid and notifies the Open Shortest Path First (OSPF) process.
- If you did not configure a delay time, if synchronization is disabled or down, or if an interface was removed from an IGP process, LDP stops the timer and immediately notifies the OSPF process.

If you configure a new delay time while a timer is running, LDP saves the new delay time but does not reconfigure the running timer.

MPLS LDP IGP Synchronization Incompatibility with IGP Nonstop Forwarding

The MPLS LDP IGP Synchronization feature is not supported during the startup period if the Interior Gateway Protocol (IGP) nonstop forwarding (NSF) is configured. The MPLS LDP IGP Synchronization feature conflicts with IGP NSF when the IGP is performing NSF during startup. After the NSF startup is complete, the MPLS LDP IGP Synchronization feature is supported.

MPLS LDP IGP Synchronization Compatibility with LDP Graceful Restart

LDP Graceful Restart protects traffic when a Label Distribution Protocol (LDP) session is lost. If an interface that supports a Graceful Restart-enabled LDP session fails, MPLS LDP IGP synchronization is still achieved on the interface while it is protected by Graceful Restart. MPLS LDP IGP synchronization is eventually lost under the following circumstances:

- If LDP fails to restart before the LDP Graceful Restart reconnect timer expires.
- If an LDP session restarts through other interfaces, but the LDP session on the protected interface fails to recover when the LDP Graceful Restart recovery timer expires.

How to Configure MPLS LDP IGP Synchronization

Configuring MPLS LDP IGP Synchronization with OSPF Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol ldp**
5. **interface** *type number*
6. **ip address** *prefix mask*
7. **mpls ip**
8. **exit**
9. **router ospf** *process-id*
10. **network** *ip-address wildcard-mask* **area** *area-id*
11. **mpls ldp sync**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Globally enables hop-by-hop forwarding.
Step 4	**mpls label protocol ldp** **Example:** `Device(config)# mpls label protocol ldp`	Specifies the Label Distribution Protocol (LDP) as the default protocol.
Step 5	**interface** *type number* **Example:**	Specifies the interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface POS 3/0`	
Step 6	**ip address** *prefix mask* **Example:** `Device(config-if)# ip address 10.0.0.11 255.255.255.255`	Assigns an IP address to the interface.
Step 7	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables hop-by-hop forwarding on the interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 9	**router ospf** *process-id* **Example:** `Device(config)# router ospf 1`	Enables Open Shortest Path First (OSPF) routing, and enters router configuration mode.
Step 10	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Device(config-router)# network 10.0.0.0 0.0.255.255 area 3`	Specifies the interface on which OSPF runs and defines the area ID for that interface.
Step 11	**mpls ldp sync** **Example:** `Device(config-router)# mpls ldp sync`	Enables the Multiprotocol Label Switching (MPLS) Interior Gateway Protocol (IGP) synchronization for interfaces belonging for an OSPF or an Intermediate System-to-Intermediate System (IS-IS) process.
Step 12	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Disabling MPLS LDP IGP Synchronization from Some OSPF Interfaces

When you issue the **mpls ldp sync** command, all of the interfaces that belong to an Open Shortest Path First (OSPF) process are enabled for Label Distribution Protocol (LDP) Interior Gateway Protocol (IGP) synchronization. To remove LDP IGP synchronization from some interfaces, use the **no mpls ldp igp sync** command on those interfaces.

Perform the following task to disable LDP IGP synchronization from some OSPF interfaces after they are configured with LDP IGP synchronization through the **mpls ldp sync** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no mpls ldp igp sync**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface POS 0/3/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 4	**no mpls ldp igp sync** **Example:** `Device(config-if)# no mpls ldp igp sync`	Disables MPLS LDP IGP synchronization for that interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying MPLS LDP IGP Synchronization with OSPF

After you configure the interfaces for the Label Distribution Protocol (LDP), Open Shortest Path First (OSPF), and LDP Interior Gateway Protocol (IGP) synchronization, verify that the configuration is working correctly by using the **show mpls ldp igp sync** and **show ip ospf mpls ldp interface** commands.

SUMMARY STEPS

1. **enable**

2. **show mpls ldp igp sync**
3. **show ip ospf mpls ldp interface**
4. **exit**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 show mpls ldp igp sync

Shows that the Multiprotocol Label Switching (MPLS) LDP IGP synchronization is configured correctly because LDP is configured and the SYNC status shows that synchronization is enabled.

Example:

```
Device# show mpls ldp igp sync

FastEthernet0/0/0:
 LDP configured;  SYNC enabled.
 SYNC status: sync achieved; peer reachable.
 IGP holddown time: infinite.
 Peer LDP Ident: 10.0.0.1:0
 IGP enabled: OSPF 1
```

If MPLS LDP IGP synchronization is not enabled on an interface, the output appears as follows:

Example:

```
FastEthernet0/3/1:
 LDP configured; LDP-IGP Synchronization not enabled.
```

Step 3 show ip ospf mpls ldp interface

Shows that the interfaces are properly configured.

Example:

```
Device# show ip ospf mpls ldp interface

FastEthernet0/3/1
  Process ID 1, Area 0
  LDP is configured through LDP autoconfig
  LDP-IGP Synchronization: Yes
  Holddown timer is not configured
  Timer is not running
FastEthernet0/0/2
  Process ID 1, Area 0
  LDP is configured through LDP autoconfig
  LDP-IGP Synchronization: Yes
  Holddown timer is not configured
  Timer is not running
```

Step 4 **exit**

Returns to user EXEC mode.

Example:

```
Device# exit
Device>
```

Configuring MPLS LDP IGP Synchronization with IS-IS Interfaces

Configuring MPLS LDP IGP Synchronization on All IS-IS Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol ldp**
5. **router isis** *process-name*
6. **mpls ldp sync**
7. **interface** *type number*
8. **ip address** *prefix mask*
9. **ip router isis** *process-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Globally enables hop-by-hop forwarding.
Step 4	**mpls label protocol ldp** **Example:**	Specifies the Label Distribution Protocol (LDP) as the default label distribution protocol.

	Command or Action	Purpose
	`Device(config)# mpls label protocol ldp`	
Step 5	**router isis** *process-name* **Example:** `Device(config)# router isis ISIS`	Enables the Intermediate System-to-Intermediate System (IS-IS) protocol on the device, specifies an IS-IS process, and enters router configuration mode.
Step 6	**mpls ldp sync** **Example:** `Device(config-router)# mpls ldp sync`	Enables Multiprotocol Label Switching (MPLS) LDP Interior Gateway Protocol (IGP) synchronization on interfaces belonging to an IS-IS process.
Step 7	**interface** *type number* **Example:** `Device(config-router)# interface POS 0/3/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 8	**ip address** *prefix mask* **Example:** `Device(config-if)# ip address 10.25.25.11 255.255.255.0`	Assigns an IP address to the interface.
Step 9	**ip router isis** *process-name* **Example:** `Device(config-if)# ip router isis ISIS`	Enables IS-IS.
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring MPLS LDP IGP Synchronization on an IS-IS Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *prefix mask*
5. **ip router isis**
6. **exit**
7. **router isis**
8. **mpls ldp sync**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface POS 0/2/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 4	**ip address** *prefix mask* **Example:** `Device(config-if)# ip address 10.50.72.4 255.0.0.0`	Assigns an IP address to the interface.
Step 5	**ip router isis** **Example:** `Device(config-if)# ip router isis`	Enables the Intermediate System-to-Intermediate System (IS-IS) protocol for IP on the interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 7	**router isis** **Example:** `Device(config)# router isis`	Enters router configuration mode, and enables an IS-IS process on the device.
Step 8	**mpls ldp sync** **Example:** `Device(config-router)# mpls ldp sync`	Enables Label Distribution Protocol (LDP) Interior Gateway Protocol (IGP) synchronization for interfaces belonging to an IS-IS process.
Step 9	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Disabling MPLS LDP IGP Synchronization from Some IS-IS Interfaces

When you issue the **mpls ldp sync** command, all of the interfaces that belong to an Intermediate System-to-Intermediate System (IS-IS) process are enabled for Label Distribution Protocol (LDP) Interior Gateway Protocol (IGP) synchronization. To remove LDP IGP synchronization from some interfaces, use the **no mpls ldp igp sync** command on those interfaces.

Perform the following task to disable LDP IGP synchronization from some IS-IS interfaces after they are configured with LDP IGP synchronization through the **mpls ldp sync** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no mpls ldp igp sync**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface POS 0/3/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 4	**no mpls ldp igp sync** **Example:** `Device(config-if)# no mpls ldp igp sync`	Disables Multiprotocol Label Switching (MPLS) LDP IGP synchronization for that interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

Use the **debug mpls ldp igp sync** command to display events related to MPLS LDP IGP synchronization.

Configuration Examples for MPLS LDP IGP Synchronization

Example: MPLS LDP IGP Synchronization with OSPF

The following task shows how to enable the Label Distribution Protocol (LDP) for Open Shortest Path First (OSPF) process 1. The **mpls ldp sync** and the OSPF **network** commands enable LDP on interfaces POS0/0/0, POS0/1/0, and POS1/1/0, respectively. The **no mpls ldp igp sync** command on interface POS1/0/0 prevents LDP from being enabled on interface POS1/0/0, even though OSPF is enabled for that interface.

```
Device# configure terminal
Device(config)# interface POS0/0/0
Device(config-if)# ip address 10.0.0.1
Device(config-if)# mpls ip
!
Device(config)# interface POS0/1/0
Device(config-if)# ip address 10.0.1.1
Device(config-if)# mpls ip
!
Device(config)# interface POS1/1/0
Device(config-if)# ip address 10.1.1.1
Device(config-if)# mpls ip
!
Device(config)# interface POS1/0/0
Device(config-if)# ip address 10.1.0.1
Device(config-if)# mpls ip
!
Device(config)# router ospf 1
Device(config-router)# network 10.0.0.0 0.0.255.255 area 3
Device(config-router)# network 10.1.0.0 0.0.255.255 area 3
Device(config-router)# mpls ldp sync
Device(config-router)# exit
Device(config)# interface POS1/0/0
Device(config-if)# no mpls ldp igp sync
```

Example: MPLS LDP IGP Synchronization with IS-IS

The following examples show the configuration commands you can use to configure MPLS LDP IGP synchronization on interfaces POS0/2 /0 and POS0/3/0, which are running IS-IS processes:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface POS0/2/0
Device(config-if)# ip router isis
Device(config-if)# exit
Device(config)# router isis
Device(config-router)# mpls ldp sync
Device(config-router)# exit
.
.
.
Device(config)# interface POS0/3/0
Device(config-if)# ip router isis
Device(config-if)# exit
```

```
Device(config)# router isis
Device(config-router)# mpls ldp sync
Device(config-router)# exit
Device(config) exit
Device#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS LDP commands	Cisco IOS Multiprotocol Label Switching Command Reference
LDP autoconfiguration	"MPLS LDP Autoconfiguration" module in the *MPLS Label Distribution Protocol Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 3037	LDP Applicability
RFC 5036	LDP Specification

MIBs

MIBs	MIBs Link
MPLS LDP MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

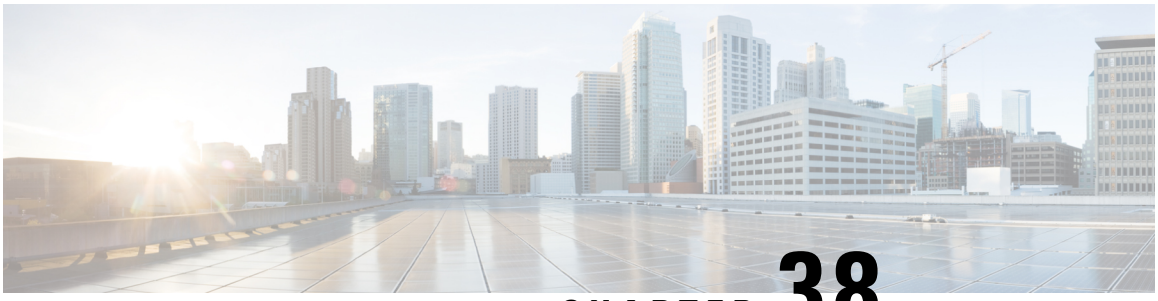
Feature Information for MPLS LDP IGP Synchronization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 92: Feature Information for MPLS LDP IGP Synchronization

Feature Name	Releases	Feature Information
MPLS LDP IGP Synchronization		The MPLS LDP IGP Synchronization feature ensures that LDP is fully established before the IGP path is used for switching.



CHAPTER 38

MPLS LDP Inbound Label Binding Filtering

Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) supports inbound label binding filtering. You can use the MPLS LDP Inbound Label Binding Filtering feature to configure access control lists (ACLs) for controlling the label bindings a label switch router (LSR) accepts from its peer LSRs.

Restrictions for MPLS LDP Inbound Label Binding Filtering

Inbound label binding filtering does not support extended access control lists (ACLs); it only supports standard ACLs.

Information about MPLS LDP Inbound Label Binding Filtering

Overview of MPLS LDP Inbound Label Binding Filtering

The MPLS LDP Inbound Label Binding Filtering feature can be used to control the amount of memory used to store Label Distribution Protocol (LDP) label bindings advertised by other devices. For example, in a simple Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) environment, the VPN provider edge (PE) devices might require label switched paths (LSPs) only to their peer PE devices (that is, they do not need LSPs to core devices). Inbound label binding filtering enables a PE device to accept labels only from other PE devices.

How to Configure MPLS LDP Inbound Label Binding Filtering

Configuring MPLS LDP Inbound Label Binding Filtering

Perform this task to configure a device for inbound label filtering. The following configuration allows the device to accept only the label for prefix 25.0.0.2 from the Label Distribution Protocol (LDP) neighbor device 10.12.12.12.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list standard** *access-list-number*
4. **permit** {*source* [*source-wildcard*] | **any**} [**log**]
5. **exit**
6. **mpls ldp neighbor** [**vrf** *vpn-name*] *nbr-address* **labels accept** *acl*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list standard** *access-list-number* **Example:** `Device(config)# ip access-list standard 1`	Defines a standard IP access list with a number.
Step 4	**permit** {*source* [*source-wildcard*] | **any**} [**log**] **Example:** `Device(config-std-nacl)# permit 10.0.0.0`	Specifies one or more prefixes permitted by the access list.
Step 5	**exit** **Example:** `Device(config-std-nacl)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 6	**mpls ldp neighbor** [**vrf** *vpn-name*] *nbr-address* **labels accept** *acl* **Example:** `Device(config)# mpls ldp neighbor 10.12.12.12 labels accept 1`	Specifies the access control list (ACL) to be used to filter label bindings for the specified LDP neighbor.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying that MPLS LDP Inbound Label Bindings are Filtered

If inbound filtering is enabled, perform the following tasks to verify that inbound label bindings are filtered.

SUMMARY STEPS

1. **enable**
2. **show mpls ldp neighbor** [**vrf** *vpn-name*] [*address* | *interface*] [**detail**]
3. **show ip access-list** [*access-list-number* | *access-list-name*]
4. **show mpls ldp bindings**
5. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show mpls ldp neighbor** [**vrf** *vpn-name*] [*address* | *interface*] [**detail**]

Shows the status of the Label Distribution Protocol (LDP) session, including the name or number of the access control list (ACL) configured for inbound filtering.

Note To display information about inbound label binding filtering, you must enter the **detail** keyword.

Example:

```
Device# show mpls ldp neighbor 10.12.12.12 detail
 Peer LDP Ident: 10.12.12.12:0; Local LDP Ident 10.13.13.13:0
   TCP connection: 10.12.12.12.646 - 10.13.13.13.12592
   State: Oper; Msgs sent/rcvd: 49/45; Downstream; Last TIB rev sent 1257
   Up time: 00:32:41; UID: 1015; Peer Id 0;
   LDP discovery sources:
```

```
    Serial1/0/0; Src IP addr: 192.168.1.1
     holdtime: 15000 ms, hello interval: 5000 ms
   Addresses bound to peer LDP Ident:
    10.0.0.129       10.12.12.12     192.168.1.1
   Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
   LDP inbound filtering accept acl: 1
```

Step 3 **show ip access-list** [*access-list-number* | *access-list-name*]

Displays the contents of all current IP access lists or of a specified access list.

Note It is important that you enter this command to see how the access list is defined; otherwise, you cannot verify inbound label binding filtering.

The following command output shows the contents of IP access list 1:

Example:

```
Device# show ip access 1
Standard IP access list 1
  permit 10.0.0.0, wildcard bits 0.0.0.255 (1 match)
```

Step 4 **show mpls ldp bindings**

Verifies that the label switch router (LSR) has remote bindings only from a specified peer for prefixes permitted by the access list.

Example:

```
Device# show mpls ldp bindings
 tib entry: 10.0.0.0/8, rev 4
     local binding:  tag: imp-null
 tib entry: 10.2.0.0/16, rev 1137
     local binding:  tag: 16
 tib entry: 10.2.0.0/16, rev 1139
     local binding:  tag: 17
 tib entry: 10.12.12.12/32, rev 1257
     local binding:  tag: 18
 tib entry: 10.13.13.13/32, rev 14
     local binding:  tag: imp-null
 tib entry: 10.10.0.0/16, rev 711
     local binding:  tag: imp-null
 tib entry: 10.0.0.0/8, rev 1135
     local binding:  tag: imp-null
     remote binding: tsr: 10.12.12.12:0, tag: imp-null
 tib entry: 10.0.0.0/8, rev 8
     local binding:  tag: imp-null
```

Step 5 **exit**

Returns to user EXEC mode.

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS LDP Inbound Label Binding Filtering

Examples: MPLS LDP Inbound Label Binding Filtering Configuration

In the following example, the **mpls ldp neighbor labels accept** command is configured with an access control list to filter label bindings received on sessions with the neighbor 10.110.0.10.

Label bindings for prefixes that match 10.b.c.d are accepted, where b is less than or equal to 63, and c and d can be any integer between 0 and 128. Other label bindings received from 10.110.0.10 are rejected.

```
Device# configure terminal
Device(config)# access-list 1 permit 10.63.0.0 0.63.255.255
Device(config)# mpls ldp neighbor 10.110.0.10 labels accept 1
Device(config)# end
```

In the following example, the **show mpls ldp bindings neighbor** command displays label bindings that were learned from 10.110.0.10. This example verifies that the LIB does not contain label bindings for prefixes that have been excluded.

```
Device# show mpls ldp bindings neighbor 10.110.0.10

tib entry: 10.2.0.0/16, rev 4
    remote binding: tsr: 10.110.0.10:0, tag: imp-null
tib entry: 10.43.0.0/16, rev 6
    remote binding: tsr: 10.110.0.10:0, tag: 16
tib entry: 10.52.0.0/16, rev 8
    remote binding: tsr: 10.110.0.10:0, tag: imp-null
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS Label Distribution Protocol (LDP)	"MPLS Label Distribution Protocol" module in the *MPLS Label Distribution Protocol Configuration Guide*

MIBs

MIB	MIBs Link
LDP Specification, draft-ietf-mpls-ldp-08.txt	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mib

RFCs

RFC	Title
RFC 3036	LDP Specification
RFC 3037	LDP Applicability

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP Inbound Label Binding Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

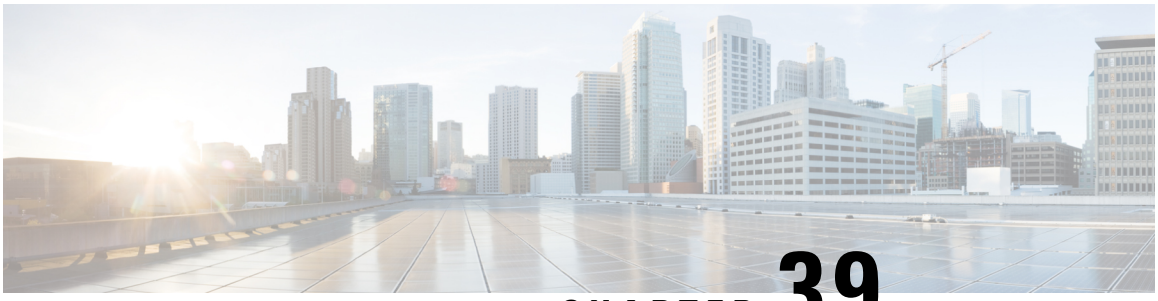
carrier supporting carrier—A situation where one service provider allows another service provider to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

CE device—customer edge device. A device that is part of a customer network and that interfaces to a provider edge (PE) device.

inbound label binding filtering—Allows label switch routers (LSRs) to control which label bindings it will accept from its neighboring LSRs. Consequently, an LSR does not accept or store some label bindings that its neighbors advertise.

label—A short fixed-length identifier that tells switching nodes how to forward data (packets or cells).

label binding—An association between a destination prefix and a label.



CHAPTER 39

MPLS LDP Local Label Allocation Filtering

The MPLS LDP Local Label Allocation Filtering feature introduces CLI commands to modify the way in which Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) handles local label allocation. This MPLS LDP feature enhancement enables the configuration of filtering policies for selective local label binding assignments by LDP to improve LDP scalability and convergence.

Prerequisites for MPLS LDP Local Label Allocation Filtering

The MPLS LDP Local Label Allocation Filtering feature requires the MPLS Forwarding Infrastructure (MFI).

Restrictions for MPLS LDP Local Label Allocation Filtering

- This feature does not support access lists; it supports prefix lists.
- Label Distribution Protocol (LDP) local label allocation configuration for prefix list or host routes is supported only in the global routing table.
- LDP and Routing Information Base (RIB) restart handling does not apply.
- Wildcard Forwarding Equalence Class (FEC) requests are not supported.
- Remote bindings are retained for LDP table entries that are filtered.

Information About MPLS LDP Local Label Allocation Filtering

MPLS LDP Local Label Allocation Filtering Overview

The Label Distribution Protocol (LDP) allocates a local label for every route learned from the Interior Gateway Protocol (IGP). In the absence of inbound and outbound label filtering, these local labels are advertised to and learned by all peers.

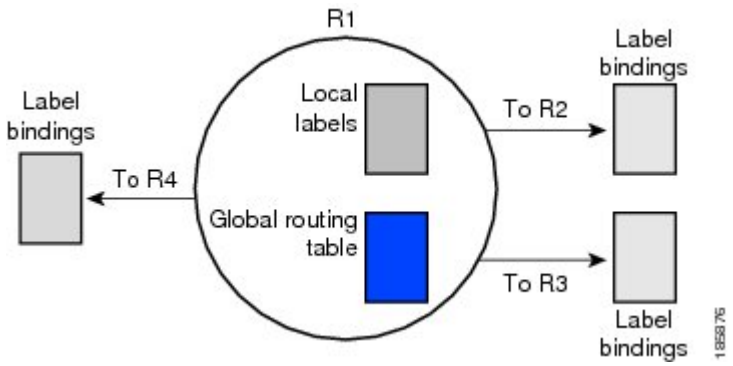
In most Layer 3 Virtual Private Network (VPN) configurations only the label switched paths (LSPs) created to reach the /32 host routes or Border Gateway Protocol (BGP) next hops between the provider edge (PE) devices carry traffic and are relevant to the Layer 3 VPNs. LSPs between the PE devices that are not members of a VPN use more memory and create additional processing in LDP across the core.

With the load increases in the service provider domain in the last decade (1997-2007), scalability has become more important in the service provider networks. Controlling the local label allocation could off-load LDP processing of non-VPN LSPs in the service provider network core devices.

The MPLS LDP Local Label Allocation Filtering feature introduces the **mpls ldp label** and **allocate** commands that allow you to configure LDP to selectively allocate local labels for a subset of the prefixes learned from the IGP. You can select that LDP allocate local labels for prefixes configured in a prefix list in the global table or for host routes in the global table.

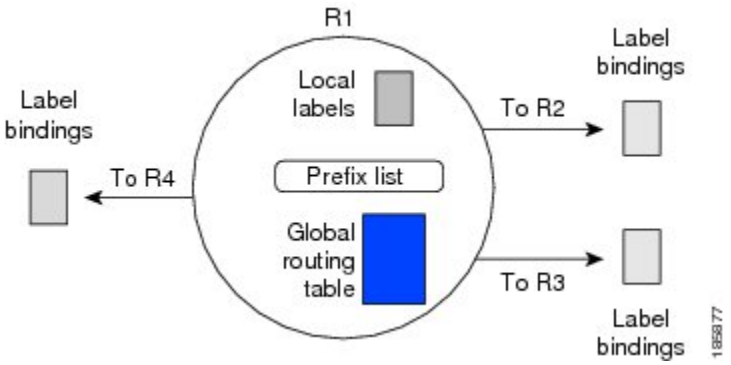
Local label allocation filtering reduces the number of local labels allocated and therefore the number of messages exchanged with peers. This improves LDP scalability and convergence. The two figures below show how controlling local label allocation can reduce local label space size and greatly reduce the number of advertisements to peers. The first figure below shows default LDP label allocation behavior. LDP allocates a local label for every route and advertises a label binding for every route learned from the IGP.

Figure 46: Default LDP Local Label Allocation Behavior



The figure below shows LDP behavior with local label allocation control configured. The size of the local label space and the number of label binding advertisements are reduced with local label allocation filtering through the use of a prefix list. The decrease in the number of local labels and label binding advertisement messages reduces the amount of memory use and improves convergence time for LDP. The MPLS LDP Local Label Allocation Filtering feature also allows for more efficient use of the label space.

Figure 47: LDP Behavior with Local Label Allocation Controls



The figure above shows that device R1 learns a number of routes from its IGP neighbors on devices R2, R3, and R4. A prefix list defined on device R1 specifies the prefixes for which LDP allocates a local label.

Note In general, the number of Label Information Base (LIB) entries remains the same regardless of the kind of label filtering. This is because the remote label bindings for the prefixes that are filtered are kept in the LIB. Memory use is reduced because local label filtering decreases the number of local labels allocated and the number of label bindings advertised to and stored by the peers of a label switch router (LSR).

Prefix Lists for MPLS LDP Local Label Allocation Filtering Benefits and Description

The MPLS LDP Local Label Allocation Filtering feature allows you to configure the Label Distribution Protocol (LDP) to allocate local labels for a subset of the learned prefixes. LDP accepts the prefix and allocates a local label if the prefix is permitted by a prefix list. If the prefix list is not defined, LDP accepts all prefixes and allocates local labels based on its default mode of operation.

The benefits of using prefix lists for LDP local label allocation filtering are as follows:

- Prefix lists provide more flexibility for specifying a subset of prefixes and masks.
- Prefix lists use a tree-based matching technique. This technique is more efficient than evaluating prefixes or host routes sequentially.
- Prefix lists are easy to modify.

You configure a prefix list for the MPLS LDP Local Label Allocation Filtering feature with the **ip prefix-list** command.

Local Label Allocation Changes and LDP Actions

The MPLS LDP Local Label Allocation Filtering enhancement modifies the Label Distribution Protocol's (LDP's) local label allocation handling. The feature supports local label allocation filtering through the specification of a prefix list or host routes.

With the introduction of this feature, LDP needs to determine whether a prefix filter is already configured to control the local label allocation on the local node. If a prefix list exists, the local label allocation is confined to the list of prefixes permitted by the configured prefix list.

LDP also needs to respond to local label allocation configuration changes and to configuration changes that affect the prefix list that LDP is using. Any of the following configuration changes can trigger LDP actions:

- Creating a local label allocation configuration
- Deleting or changing a local label allocation configuration
- Creating a new prefix list for a local label allocation configuration
- Deleting or changing a prefix list for a local label allocation configuration

LDP responds to local label allocation configuration changes by updating the Label Information Database (LIB) and the forwarding table in the global routing table. To update the LIB after a local label filter configuration change without a session reset, LDP keeps all remote bindings.

If you create a local label allocation configuration without defining a prefix list, no LDP action is required. The local label allocation configuration has no effect because the prefix list is created and permits all prefixes.

If you create or change a prefix list and prefixes that were previously allowed are rejected, LDP goes through a label withdraw and release procedure before the local labels for these prefixes are deallocated.

If you delete a prefix, LDP goes through the label withdraw and release procedure for the LIB local label. If the associated prefix is one for which no LIB entry should be allocated, LDP bypasses this procedure.

The LDP default behavior is to allocate local labels for all non-BGP prefixes. This default behavior does not change with the introduction of this feature and the **mpls ldp label** and **allocate** commands.

Note The local label allocation filtering has no impact on inbound label filtering because both provide LDP filtering independently. The LDP Inbound Label Binding Filtering feature controls label bindings that a label switch router (LSR) accepts from its peer LSRs through the use of access control lists (ACLs). The MPLS LDP Local Label Allocation Filtering feature controls the allocation of local labels through the use of prefix lists or host routes.

LDP Local Label Filtering and BGP Routes

The Label Distribution Protocol (LDP) default behavior is to allocate local labels for all non-Border Gateway Protocol (BGP) prefixes.

LDP does not apply the configured local label filter to redistributed BGP routes in the global table for which BGP allocates local label, but LDP does the advertisements (Inter-AS Option C). LDP neither forwards these entries nor releases the local labels allocated by BGP.

How to Configure MPLS LDP Local Label Allocation Filtering

Creating a Prefix List for MPLS LDP Local Label Allocation Filtering

Perform the following task to create a prefix list for the Label Distribution Protocol (LDP) local label allocation filtering. A prefix list allows LDP to selectively allocate local labels for a subset of the routes learned from the Interior Gateway Protocol {IGP). The decrease in the number of local labels in the LDP Label Information Base (LIB) and the number of label mapping advertisements reduces the amount of memory use and improves convergence time for LDP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** {*list-name* | *list-number*} [**seq** *number*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-length*] [**le** *le-length*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** {*list-name* | *list-number*} [**seq** *number*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-length*] [**le** *le-length*] **Example:** `Device(config)# ip prefix-list list1 permit 192.168.0.0/16 le 20`	Creates a prefix list or adds a prefix-list entry. • The *list-name* argument configures a name to identify the prefix list. • The *list-number* argument configures a number to identify the prefix list. • The **seq** *number* keyword and argument apply a sequence number to a prefix-list entry. The range of sequence numbers is 1 to 4294967294. If a sequence number is not entered when this command is configured, a default sequence numbering is applied to the prefix list. The number 5 is applied to the first prefix entry, and subsequent unnumbered entries are incremented by 5.

	Command or Action	Purpose
		• The **deny** keyword denies access for a matching condition. • The **permit** keyword permits access for a matching condition. • The *network*/*length* arguments and keyword configure the network address, and the length of the network mask in bits. The network number can be any valid IP address or prefix. The bit mask can be a number from 0 to 32. • The **ge** *ge-length* keyword and argument specify the lesser value of a range (the "from" portion of the range description) by applying the *ge-length* argument to the range specified. The *ge-length* argument represents the minimum prefix length to be matched. The **ge** keyword represents the greater than or equal to operator. • The **le** *le-length* keyword and argument specify the greater value of a range (the "to" portion of the range description) by applying the *le-length* argument to the range specified. The *le-length* argument represents the maximum prefix length to be matched. The **le** keyword represents the less than or equal to operator.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring MPLS LDP Local Label Allocation Filtering

Perform the following task to configure the Label Distribution Protocol (LDP) local allocation filtering. Configuring filtering policies for selective local label binding assignments by LDP improves LDP scalability and convergence. You can configure either a prefix list or host routes as a filter for local label allocation.

Note The **host-routes** keyword for the **allocate** command makes it convenient for you to specify a commonly used set of prefixes.

Note A maximum of one local label allocation filter is supported for the global table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp label**
4. **allocate global prefix-list** {*list-name* | *list-number*}
5. **allocate global host-routes**
6. **no allocate global** {**prefix-list** {*list-name* | *list-number*} | **host-routes**}
7. **no mpls ldp label**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ldp label** **Example:** `Device(config)# mpls ldp label`	Enters MPLS LDP label configuration mode to specify how LDP handles local label allocation.
Step 4	**allocate global prefix-list** {*list-name* | *list-number*} **Example:** `Device(config-ldp-lbl)# allocate global prefix-list list1`	Configures local label allocation filters for learned routes for LDP. • The **global** keyword specifies the global routing. • The **prefix-list** keyword specifies a prefix list to be used as a filter for MPLS LDP local label allocation. • The *list-name* argument indicates a name that identifies the prefix list. • The *list-number* argument indicates a number that identifies the prefix list.
Step 5	**allocate global host-routes** **Example:** `Device(config-ldp-lbl)# allocate global host-routes`	Configures local label allocation filters for learned routes for LDP. • The **global** keyword specifies the global routing. • The **host-routes** keyword specifies that local label allocation be done for host routes only.

	Command or Action	Purpose
Step 6	**no allocate global** {**prefix-list** {*list-name* | *list-number*} | **host-routes**} **Example:** `Device(config-ldp-lbl)# no allocate global host-routes`	Removes the specific MPLS LDP local label allocation filter without resetting the LDP session. • The **global** keyword specifies the global routing. • The **prefix-list** keyword specifies a prefix list to be used as a filter for MPLS LDP local label allocation. • The *list-name* argument indicates a name that identifies the prefix list. • The *list-number* argument indicates a number that identifies the prefix list. • The **host-routes** keyword specifies that host routes be used as a filter for MPLS LDP local label allocation.
Step 7	**no mpls ldp label** **Example:** `Device(config-ldp-lbl)# no mpls ldp label`	Removes all local label allocation filters configured under the MPLS LDP label configuration mode and restores LDP default behavior for local label allocation without a session reset.
Step 8	**end** **Example:** `Device(config-ldp-lbl)# end`	Returns to privileged EXEC mode.

Verifying MPLS LDP Local Label Allocation Filtering Configuration

SUMMARY STEPS

1. **enable**
2. **show mpls ldp bindings detail**
3. **debug mpls ldp binding filter**
4. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show mpls ldp bindings detail**

Verifies that local label allocation filtering is configured as you expect.

Example:

```
Device# show mpls ldp bindings detail

Advertisement spec:
        Prefix acl = bar
Local label filtering spec: host routes.
  lib entry: 10.1.1.1/32, rev 9
  lib entry: 10.10.7.0/24, rev 10
  lib entry: 10.10.8.0/24, rev 11
  lib entry: 10.10.9.0/24, rev 12
  lib entry: 10.41.41.41/32, rev 17
  lib entry: 10.50.50.50/32, rev 15
  lib entry: 10.60.60.60/32, rev 18
  lib entry: 10.70.70.70/32, rev 16
  lib entry: 10.80.80.80/32, rev 14
```

The output of this command verifies that host routes are configured as the local label allocation filter for the device.

Step 3 **debug mpls ldp binding filter**

Verifies that local label allocation filtering was configured properly and to display how LDP accepts or withdraw labels.

Example:

```
Device# debug mpls ldp binding filter
LDP Local Label Allocation Filtering changes debugging is on
.
.
.
```

Step 4 **exit**

Returns to user EXEC mode.

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS LDP Local Label Allocation Filtering

Examples: Creating a Prefix List for MPLS LDP Local Label Allocation Filtering

The following examples show how to configure a prefix list for MPLS LDP local label allocation filtering.

In this example, prefix list List1 permits only 192.168.0.0/16 prefixes. The Label Distribution Protocol (LDP) accepts 192.168.0.0/16 prefixes, but does not assign a local label for the following prefixes: 192.168.0.0/24 and 192.168.2.0/24. For example:

```
configure terminal
!
ip prefix-list List1 permit 192.168.0.0/16
end
```

In the following example, prefix list List2 permits a range of prefixes from 192.168.0.0/16 to /20 prefixes. LDP accepts 192.168.0.0/16 prefixes, but does not assign local labels for the following prefixes: 192.168.0.0/24 and 192.168.2.0/24.

```
configure terminal
!
ip prefix-list List2 permit 192.168.0.0/16 le 20
end
```

In the following example, prefix list List3 permits a range of prefixes greater than /18. LDP accepts 192.168.17.0/20 and 192.168.2.0/24 prefixes, but does not assign a local label for 192.168.0.0/16.

```
configure terminal
!
ip prefix-list List3 permit 192.168.0.0/16 ge 18
end
```

Examples: Configuring MPLS LDP Local Label Allocation Filtering

This examples shows how to allocate a prefix list to be used as a local label allocation filter:

```
configure terminal
!
ip prefix-list List3 permit 192.168.0.0/16 ge 18
!
mpls ldp label
 allocate global prefix-list List3
 exit
exit
```

Prefix list List3, which permits a range of prefixes greater than /18, is configured as the local label allocation filter for the device. The Label Distribution Protocol (LDP) allows 192.168.17.0/20 and 192.168.2.0/24 prefixes, but withdraws labels for prefixes not in the allowed range.

In the following example, host routes are configured as the local label allocation filter:

```
configure terminal
!
mpls ldp label
 allocate global host-routes
 exit
exit
```

LDP allocates local labels for host routes that are in the global routing table.

In the following example, a specific local label allocation filter is removed:

```
configure terminal
!
mpls ldp label
 no allocate global host-routes
 exit
exit
```

In the following example, all local label allocation filters configured in MPLS LDP label configuration mode are removed and the default LDP local label allocation is restored without a session reset:

```
configure terminal
!
no mpls ldp label
 exit
exit
```

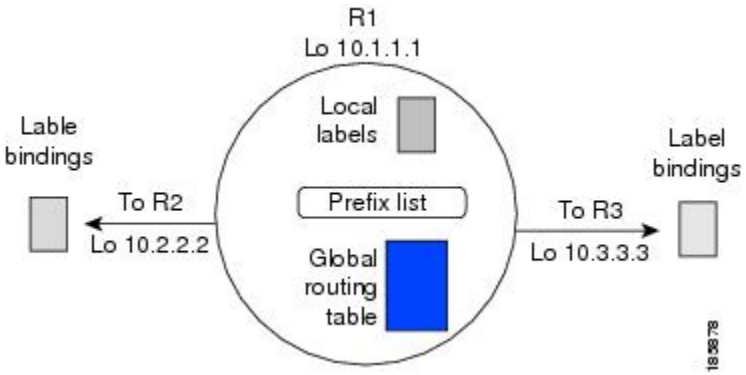
Examples: Sample MPLS LDP Local Label Allocation Filtering Configuration

The figure below is a sample configuration that is used in this section to show how Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) local label allocation filtering works:

- Devices R1, R2, and R3 have loopback addresses 10.1.1.1, 10.2.2.2, and 10.3.3.3 defined and advertised by the Interior Gateway Protocol (IGP), respectively.
- 10.1.1.1 is the router ID of Device R1, 10.2.2.2 is the router ID of Device R2, and 10.3.3.3 is the router ID of Device R3.
- A prefix list is defined on Device R1 to specify the local labels for which LDP allocates a local label.

Device RI learns a number of routes from its IGP neighbors on Devices R2 and R3.

Figure 48: LDP Local Label Allocation Filtering Example



You can use LDP CLI commands to verify the following:

- Device R1 has allocated a local label for the correct subset of the prefixes.
- Devices R2 and R3 did not receive any remote bindings for the prefixes for which Device R1 did not assign a local label.

Routing Table on Device R1

You can enter the **show ip route** command to display the current state of the routing table. The following example shows the routing table on Device R1 based on the figure above:

```
Device# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
```

```
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     10.0.0.0/32 is subnetted, 1 subnets
C       10.1.1.1 is directly connected, Loopback0
     10.2.0.0/32 is subnetted, 1 subnets
O       10.2.2.2 [110/11] via 10.10.7.1, 00:00:36, FastEthernet1/0/0
     10.3.0.0/32 is subnetted, 1 subnets
O       10.3.3.3 [110/11] via 10.10.9.1, 00:00:36, FastEthernet3/0/0
     10.0.0.0/24 is subnetted, 3 subnets
C       10.10.7.0 is directly connected, FastEthernet1/0/0
O       10.10.8.0 [110/20] via 10.10.9.1, 00:00:36, FastEthernet3/0/0
                  [110/20] via 10.10.7.1, 00:00:36, FastEthernet1/0/0
C       10.10.9.0 is directly connected, FastEthernet3/0/0
```

Local Label Bindings on Devices R1, R2, and R3

You can enter the **show mpls ldp bindings** command on Devices R1, R2, and R3 to display the contents of the Label Information Base (LIB) on each device. In the following examples, the default Label Distribution Protocol (LDP) allocation behavior is in operation; that is, LDP allocates a local label for every route and advertises a label binding for every route learned from the Interior Gateway Protocol (IGP).

LIB on Device R1

This example shows the contents of the LIB on Device R1 based on the configuration in the figure above:

```
Device# show mpls ldp bindings

  lib entry: 10.1.1.1/32, rev 7
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: 16
        remote binding: lsr: 10.2.2.2:0, label: 17
  lib entry: 10.2.2.2/32, rev 13
        local binding:  label: 1000
        remote binding: lsr: 10.3.3.3:0, label: 18
        remote binding: lsr: 10.2.2.2:0, label: imp-null
  lib entry: 10.3.3.3/32, rev 15
        local binding:  label: 1002
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: 18
  lib entry: 10.10.7.0/24, rev 8
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: 17
        remote binding: lsr: 10.2.2.2:0, label: imp-null
  lib entry: 10.10.8.0/24, rev 11
        local binding:  label: 1001
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: imp-null
  lib entry: 10.10.9.0/24, rev 9
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: 16
```

The local labels assigned to 10.2.2.2 and 10.3.3.3 on Device R1 are advertised to Devices R2 and R3.

LIB on Device R2

This example shows the contents of the LIB on Device R2 based on the configuration in the figure above:

```
Device# show mpls ldp bindings

  lib entry: 10.1.1.1/32, rev 11
        local binding:  label: 17
        remote binding: lsr: 10.3.3.3:0, label: 16
        remote binding: lsr: 10.1.1.1:0, label: imp-null
  lib entry: 10.2.2.2/32, rev 7
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: 18
        remote binding: lsr: 10.1.1.1:0, label: 1000
  lib entry: 10.3.3.3/32, rev 15
        local binding:  label: 18
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.1.1.1:0, label: 1002
  lib entry: 10.10.7.0/24, rev 8
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: 17
        remote binding: lsr: 10.1.1.1:0, label: imp-null
  lib entry: 10.10.8.0/24, rev 9
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.1.1.1:0, label: 1001
  lib entry: 10.10.9.0/24, rev 13
        local binding:  label: 16
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.1.1.1:0, label: imp-null
```

LIB on Device R3

This example shows the contents of the LIB on Device R3 based on the configuration in the figure above:

```
Device # show mpls ldp bindings

  lib entry: 10.1.1.1/32, rev 13
        local binding:  label: 16
        remote binding: lsr: 10.2.2.2:0, label: 17
        remote binding: lsr: 10.1.1.1:0, label: imp-null
  lib entry: 10.2.2.2/32, rev 15
        local binding:  label: 18
        remote binding: lsr: 10.2.2.2:0, label: imp-null
        remote binding: lsr: 10.1.1.1:0, label: 1000
  lib entry: 10.3.3.3/32, rev 7
        local binding:  label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: 18
        remote binding: lsr: 10.1.1.1:0, label: 1002
  lib entry: 10.10.7.0/24, rev 11
        local binding:  label: 17
        remote binding: lsr: 10.2.2.2:0, label: imp-null
        remote binding: lsr: 10.1.1.1:0, label: imp-null
  lib entry: 10.10.8.0/24, rev 8
        local binding:  label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: imp-null
        remote binding: lsr: 10.1.1.1:0, label: 1001
  lib entry: 10.10.9.0/24, rev 9
        local binding:  label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: 16
        remote binding: lsr: 10.1.1.1:0, label: imp-null
```

Local Label Allocation Filtering Configuration on Device R1

You enter the **mpls ldp label** command to configure a local label allocation filter. The following examples show how to configure a local label allocation filter by host routes only and by a prefix list.

Local Label Allocation Filter—Host Routes Only Configuration

This example shows the selection of host routes as the only filter.

The following local label allocation filtering is defined on Device R1 under MPLS LDP label configuration mode:

```
configure terminal
!
mpls ldp label
 allocate global host-routes
 exit
exit
```

Local Label Allocation Filter—Prefix List Configuration

The following example shows how to configure a local label allocation filter that allows or denies prefixes based on a prefix list:

```
configure terminal
!
mpls ldp label
 allocate global prefix-list ListA
 exit
end
```

ListA is a prefix list defined as:

```
configure terminal
!
ip prefix-list ListA permit 0.0.0.0/32 ge 32
```

Local Label Allocation Filtering Changes Label Bindings on Devices R1, R2, and R3

After configuring a local label allocation filter on Device R1, you can enter the **show mpls ldp bindings** command again to see the changes in the local label bindings in the Label Information Base (LIB) on each device. Changes to the output in the LIB entries are highlighted in bold text.

This sample prefix list is used for the examples in the this section:

```
ip prefix-list ListA permit 0.0.0.0/32 ge 32
```

LIB on Device R1 After Local Label Allocation Filtering

This example shows how the configuration of a local label allocation prefix-list filter changes the contents of the LIB on Device R1:

```
Device# show mpls ldp bindings

  lib entry: 10.1.1.1/32, rev 7
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: 16
```

```
        remote binding: lsr: 10.2.2.2:0, label: 17
  lib entry: 10.2.2.2/32, rev 13
        local binding:  label: 1000
        remote binding: lsr: 10.3.3.3:0, label: 18
        remote binding: lsr: 10.2.2.2:0, label: imp-null
  lib entry: 10.3.3.3/32, rev 15
        local binding:  label: 1002
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: 18
  lib entry: 10.10.7.0/24, rev 8
        no local binding
        remote binding: lsr: 10.3.3.3:0, label: 17
        remote binding: lsr: 10.2.2.2:0, label: imp-null
  lib entry: 10.10.8.0/24, rev 11
        no local binding
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: imp-null
  lib entry: 10.10.9.0/24, rev 9
        no local binding
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.2.2.2:0, label: 16
```

Local label bindings for all but 10.2.2.2 and 10.3.3.3 on Device R1 are advertised as withdrawn.

LIB on Device R2 After Local Label Allocation Filtering

This example shows how the configuration of a local label allocation prefix-list filter on Device R1 changes the contents of the LIB on Device R2:

```
Device# show mpls ldp bindings
  lib entry: 10.1.1.1/32, rev 11
        local binding:  label: 17
        remote binding: lsr: 10.3.3.3:0, label: 16
  lib entry: 10.2.2.2/32, rev 7
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: 18
        remote binding: lsr: 10.1.1.1:0, label: 1000
  lib entry: 10.3.3.3/32, rev 15
        local binding:  label: 18
        remote binding: lsr: 10.3.3.3:0, label: imp-null
        remote binding: lsr: 10.1.1.1:0, label: 1002
lib entry: 10.10.7.0/24, rev 8
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: 17
  lib entry: 10.10.8.0/24, rev 9
        local binding:  label: imp-null
        remote binding: lsr: 10.3.3.3:0, label: imp-null
  lib entry: 10.10.9.0/24, rev 13
        local binding:  label: 16
        remote binding: lsr: 10.3.3.3:0, label: imp-null
```

The 10.10.7.0/24, 10.10.8.0/24, and 10.10.9.0/24 prefixes are no longer assigned local labels. Therefore, Device R1 sends no label advertisement for these prefixes.

LIB on Device R3 After Local Label Allocation Filtering

This example shows how the configuration of a local label allocation prefix-list filter on Device R1 changes the contents of the LIB on Device R3:

```
Device# show mpls ldp bindings
  lib entry: 10.1.1.1/32, rev 13
```

```
      local binding:  label: 16
      remote binding: lsr: 10.2.2.2:0, label: 17
      remote binding: lsr: 10.1.1.1:0, label: imp-null
lib entry: 10.2.2.2/32, rev 15
      local binding:  label: 18
      remote binding: lsr: 10.2.2.2:0, label: imp-null
      remote binding: lsr: 10.1.1.1:0, label: 1000
lib entry: 10.3.3.3/32, rev 7
      local binding:  label: imp-null
      remote binding: lsr: 10.2.2.2:0, label: 18
      remote binding: lsr: 10.1.1.1:0, label: 1002
lib entry: 10.10.7.0/24, rev 11
      local binding:  label: 17
      remote binding: lsr: 10.2.2.2:0, label: imp-null
lib entry: 10.10.8.0/24, rev 8
      local binding:  label: imp-null
      remote binding: lsr: 10.2.2.2:0, label: imp-null
lib entry: 10.10.9.0/24, rev 9
      local binding:  label: imp-null
      remote binding: lsr: 10.2.2.2:0, label: 16
```

The 10.10.7.0/24, 10.10.8.0/24, and 10.10.9.0/24 prefixes are no longer assigned local labels. Again, Device R1 sends no label advertisement for these prefixes.

Command to Display the Local Label Allocation Filter

You can enter the **show mpls ldp detail** command to display the filter used for local label allocation. For example:

```
Device# show mpls ldp bindings detail

Advertisement spec:
        Prefix acl = List1
Local label filtering spec: host routes. ! <--- Local local label filtering spec

  lib entry: 10.1.1.1/32, rev 9
  lib entry: 10.10.7.0/24, rev 10
  lib entry: 10.10.8.0/24, rev 11
  lib entry: 10.10.9.0/24, rev 12
  lib entry: 10.41.41.41/32, rev 17
  lib entry: 10.50.50.50/32, rev 15
  lib entry: 10.60.60.60/32, rev 18
  lib entry: 10.70.70.70/32, rev 16
  lib entry: 10.80.80.80/32, rev 14
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Configuration tasks for MPLS LDP	"MPLS Label Distribution Protocol" module in the *MPLS Label Distribution Protocol Configuration Guide*

Related Topic	Document Title
Configuration tasks for inbound label binding filtering for MPLS LDP	"MPLS LDP Inbound Label Binding Filtering" module in the *MPLS Label Distribution Protocol Configuration Guide*

RFCs

RFC	Title
RFC 3037	LDP Applicability
RFC 3815	Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)
RFC 5036	LDP Specification

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP Local Label Allocation Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

BGP—Border Gateway Protocol. An interdomain routing protocol that replaces Exterior Gateway Protocol (EGP). A BGP system exchanges reachability information with other BGP systems. It is defined by RFC 1163.

CE device—customer edge device. A device that is part of a customer network and that interfaces to a provider edge (PE) device. CE devices do not have routes to associated Virtual Private Networks (VPNs) in their routing tables.

FEC—Forwarding Equivalence Class. A set of packets that can be handled equivalently for the purpose of forwarding and thus is suitable for binding to a single label. The set of packets destined for an address prefix is one example of an FEC.

IGP—Interior Gateway Protocol. Internet protocol used to exchange routing information within a single autonomous system. Examples of common Internet IGP protocols include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), and Routing Information protocol (RIP).

label—A short fixed-length label that tells switching nodes how to forward data (packets or cells).

LDP—Label Distribution Protocol. A standard protocol between Multiprotocol Label Switching (MPLS)-enabled devices that is used for the negotiation of the labels (addresses) used to forward packets.

LIB—Label Information Base. A database used by a label switch router (LSR) to store labels learned from other LSRs, and labels assigned by the local LSR.

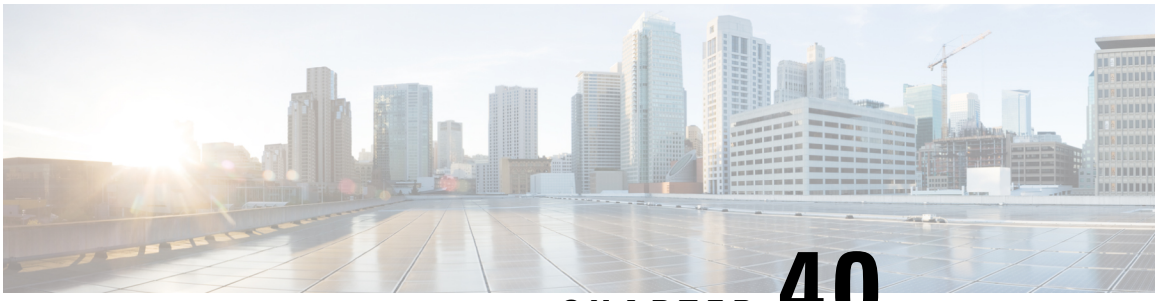
LSP—label switched path. A sequence of hops in which a packet travels from one device to another device by means of label switching mechanisms. A label switched path can be established dynamically, based on normal routing mechanisms, or through configuration.

LSR—label switch router. A device that forwards Multiprotocol Label Switching (MPLS) packets based on the value of a fixed-length label encapsulated in each packet.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the devices and the switches in the network where to forward the packets. The forwarding of MPLS packets is based on preestablished IP routing information

PE device—provider edge device. A device that is part of a service provider's network connected to a customer edge (CE) device. All Virtual Private Network (VPN) processing occurs in the PE device.

VPN—Virtual Private Network. A secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.



CHAPTER 40

MPLS LDP MD5 Global Configuration

The MPLS LDP MD5 Global Configuration feature provides enhancements to the Label Distribution Protocol (LDP) implementation of the Message Digest 5 (MD5) password. This feature allows you to enable LDP MD5 globally instead of on a per-peer basis. Using this feature you can set up password requirements for a set of LDP neighbors to help prevent unauthorized peers from establishing LDP sessions and to block spoofed TCP messages.

This document provides information about and configuration information for the global configuration of LDP MD5 protection.

Prerequisites for MPLS LDP MD5 Global Configuration

- Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on the label switch router (LSR).
- Routing (static or dynamic) must be configured for the LSR.
- Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) must be configured on the LSR. However, you can configure LDP Message Digest 5 (MD5) protection before you configure MPLS LDP. You can then use LDP MD5 protection after you configure MPLS LDP.
- A Virtual Private Network (VPN) routing and forwarding (VRF) instance must be configured if you want to configure MPLS LDP MD5 global configuration for a VRF. If you delete a VRF, the LDP MD5 global configuration for that VRF is automatically removed.

Restrictions for MPLS LDP MD5 Global Configuration

Message Digest 5 (MD5) protection described in this document applies only to Label Distribution Protocol (LDP) sessions. All enhancements described in this document do not affect Tag Distribution Protocol (TDP) sessions.

Information About MPLS LDP MD5 Global Configuration

Enhancements to LDP MD5 Protection for LDP Messages Between Peers

The MPLS LDP MD5 Global Configuration feature provides the following enhancements to the Label Distribution Protocol (LDP) support of Message Digest 5 (MD5) passwords:

- You can specify peers for which MD5 protection is required. This can prevent the establishment of LDP sessions with unexpected peers.
- You can configure passwords for groups of peers. This increases the scalability of LDP password configuration management.
- The established LDP session with a peer is not automatically torn down when the password for that peer is changed. The new password is used the next time an LDP session is established with the peer.
- You can control when the new password is used. You can configure the new password on the peer before forcing the use of the new password.
- If the neighboring nodes support graceful restart, then LDP sessions are gracefully restarted. The LDP MD5 password configuration is checkpointed to the standby Route Processors (RPs). The LDP MD5 password is used by the device when the new active RP attempts to establish LDP sessions with neighbors after the switchover.

LDP session, advertisement, and notification messages are exchanged between two LDP peers over a TCP connection. You can configure the TCP MD5 option to protect LDP messages that are exchanged over a TCP connection. You can configure this protection for each potential LDP peer. As a result, an LDP ignores any LDP hello messages sent from a label switch router (LSR) for which you have not configured a password. (LDP tries to establish an LDP session with each neighbor from which a hello message is received.)

Before the introduction of the MPLS LDP MD5 Global Configuration feature, you needed to configure a separate password for each LDP peer for which you wanted MD5 protection. This was the case even when the same password was used for multiple LDP peers. Before this feature, LDP would tear down LDP sessions with a peer immediately if a password for that peer had changed.

LDP MD5 Password Configuration Information

Before the introduction of the MPLS LDP MD5 Global Configuration feature, the command used for configuring a password for a Label Distribution Protocol (LDP) neighbor was **mpls ldp neighbor** [**vrf** *vrf-name*] *ip-address* **password** [**0** | **7**] *password*. This command configures a password for one neighbor whose router ID is the IP address in the specified virtual routing and forwarding (VRF). A label switch router (LSR) can have zero or one such configuration for each LDP neighbor.

You can use the commands provided by the MPLS LDP MD5 Global Configuration feature to configure passwords for LDP neighbors.

You must understand how LDP determines the password for an LDP session between peers before you configure Message Digest 5 (MD5) password protection for your network. LDP determines the passwords for its sessions based on the commands that you enter.

You can enter an **mpls ldp password vrf** *vrf-name* **required** [**for** *acl*] command, either with an optional *acl* argument that permits the LDP router ID of the neighbor or without an *acl* argument. Make sure that you enter a command that configures a password. Otherwise, LDP might not establish a session with the neighbor in question.

For the commands in the following password-determining process, *A.B.C.D:N* represents the LDP neighbor in VRF vpn1 and the neighbor LDP ID:

- *A.B.C.D* is the neighbor router ID.
- *N* is the neighbor label space ID.

To determine the password for an LDP session for the neighbor label space *A.B.C.D:N*, LDP looks at the password commands in the order indicated by the following statements:

- If you configured the **mpls ldp neighbor vrf vpn1** *A.B.C.D* **password** *pwd-nbr* command: The LDP session password is *pwd-nbr*. LDP looks no further and uses the password you specify. Otherwise, LDP looks to see if you configured one or more **mpls ldp vrf vpn1 password option** commands. LDP considers the commands in order of the ascending *number* arguments (*number-1st to number-n*). For example:
  - **mpls ldp vrf vpn1 password option** *number-1st* **for** *acl-1st pwd-1st*

    LDP compares the peer router ID of the neighbor (*A.B.C.D)* with this command. If *A.B.C.D* is permitted by the command access list *acl-1st*, the session password is the command password, that is, *pwd-1st*.

    If A.B.C.D is not permitted by *acl-1st*, LDP looks at the command with the next ascending *number* argument (*number-2nd*).
  - **mpls ldp vrf vpn1 password option** *number-2nd* **for** *acl-2nd pwd-2nd*

    If *A.B.C.D* is permitted by the command access list *acl-2nd*, the session password is *pwd-2nd*.

    If *A.B.C.D* is not permitted by the access list *acl-2nd*, LDP continues checking *A.B.C.D* against access lists until LDP:
    - Finds *A.B.C.D* permitted by an access list. Then the command password is the session password.
    - Has processed the *number-nth* argument of this command (*n* being the highest *number* argument you configured for this command).
  - If the **mpls ldp vrf vpn1 password option** *number-nth* **for** *acl-nth pwd-nth* command produces no match and, therefore no password, LDP looks to see if you configured the **mpls ldp password vrf vpn1 fallback** *pwd-fback* command.

    If you configured this command, the session password is *pwd-fback*.

    Otherwise, if LDP has not found a password, you did not configure a password for the session. LDP does not use MD5 protection for the session TCP connection.

LDP MD5 Password Configuration for Routing Tables

The MPLS LDP MD5 Global Configuration feature introduces commands that can establish password protection for Label Distribution Protocol (LDP) sessions between LDP neighbors or peers. These commands can apply to routes in the global routing table or in a virtual routing and forwarding (VRF) instance.

By default, if the **vrf** keyword is not specified in the command, the command applies to the global routing table. The following sample commands apply to routes in the global routing table:

```
Device# mpls ldp password required
Device# mpls ldp password option 15 for 99 pwd-acl
Device# mpls ldp password fallback pwd-fbck
```

You can configure LDP Message Digest 5 (MD5) password protection for routes in a VRF only when the VRF is configured on the label switch router (LSR). If you specify a VRF name and a VRF with that name is not configured on the LSR, LDP prints out a warning and discards the command. If you remove a VRF, LDP deletes the password configuration for that VRF. The following sample commands apply to routes in a VRF, for example, VRF vpn1:

```
Device# mpls ldp vrf vpn1 password required
Device# mpls ldp vrf vpn1 password option 15 for 99 pwd-acl
Device# mpls ldp vrf vpn1 password fallback pwd-flbk
```

How LDP Tears Down Sessions

You might require password protection for a certain set of neighbors for security reasons (for example, to prevent Label Distribution Protocol (LDP) sessions being established with unauthorized peers, or to block spoofed TCP messages). To enforce this security, you can configure a password requirement for LDP sessions with those neighbors that must have Message Digest 5 (MD5) protection (TCP session uses a password).

If you configure a password requirement for a neighbor and you did not configure a password for the neighbor, LDP tears down the LDP sessions with the neighbor. LDP also tears down the LDP sessions with the neighbor if you configured a password requirement and a password, and the password is not used in the LDP sessions.

If a password is required for a neighbor and the LDP sessions with the neighbor are established to use a password, any configuration that removes the password for the neighbor causes the LDP sessions to be torn down.

To avoid unnecessary LDP session flapping, you should perform the task as described in the next section and use caution when you change LDP passwords.

How to Configure MPLS LDP MD5 Global Configuration

Identifying LDP Neighbors for LDP MD5 Password Protection

Perform the following task to identify LDP neighbors for LDP MD5 password protection.

Before you begin

Before you start to configure passwords for Label Distribution Protocol (LDP) sessions, you must identify neighbors or groups of peers for which you want to provide Message Digest 5 (MD5) protection. For example:

- You might have several customers that all use the same core devices. To ensure security you might want to provide each customer with a different password.
- You could have defined several departmental virtual routing and forwarding (VRF) instances in your network. You could provide password protection for each VRF.
- Certain groups of peers might require password protection for security reasons. Password protection prevents unwanted LDP sessions.

SUMMARY STEPS

1. Identify LDP neighbors or groups of peers for LDP MD5 password protection.
2. Decide what LDP MD5 protection is required for each neighbor or group of peers.

DETAILED STEPS

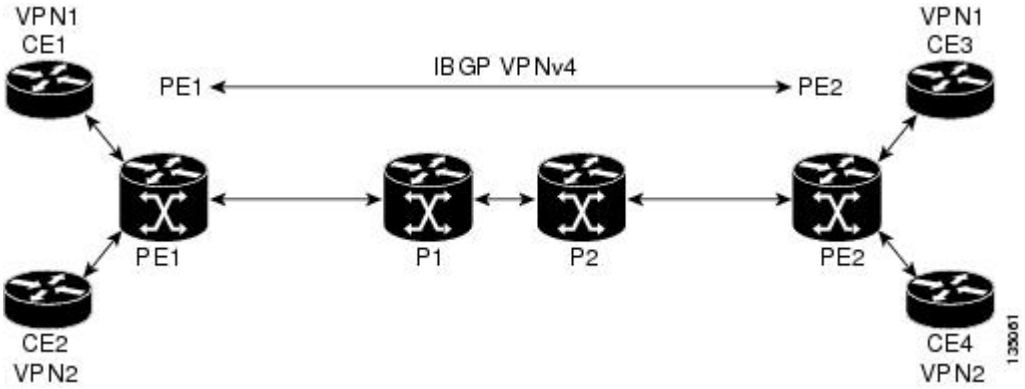
Step 1 Identify LDP neighbors or groups of peers for LDP MD5 password protection.

This task uses the network in the figure below to show how you might identify LDP neighbors for LDP MD5 protection.

The figure below shows a sample network that has the following topology:

- Carrier Supporting Carrier (CSC) is configured between provider edge (PE) device PE1 and customer edge (CE) device CE1 and between PE1 and CE2.
- Internal Border Gateway Protocol (IBGP) Virtual Private Network (VPN) IPv4 (VPNv4) to support Layer 3 VPNs is configured between PE1 and PE2.
- CE1 and CE3 are in VRF VPN1. CE2 and CE4 are in a different VRF, VPN2.

Figure 49: Sample Network: Identifying LDP Neighbors for LDP MD5 Protection



For the sample network in the figure above, you could configure separate passwords on PE1 for the following:

- VRF VPN1
- VRF VPN2

You could also configure a password requirement on PE1 for P1, P2, CE1 and CE2.

Step 2 Decide what LDP MD5 protection is required for each neighbor or group of peers.

Configuring an LDP MD5 Password for LDP Sessions

This section contains information about and instructions for configuring a Label Distribution Protocol (LDP) Message Digest 5 (MD5) password for LDP sessions. You configure an LDP MD5 password to protect your devices from unwanted LDP sessions and provide LDP session security. You can provide LDP session security for a specific neighbor, or for LDP peers from a specific virtual routing and forwarding (VRF) instance or from the global routing table, or for a specific set of LDP neighbors.

After you have identified the LDP neighbor, LDP neighbors, or LDP peers in your network for which you want LDP MD5 password protection, perform the following procedures, as you require, to configure an LDP MD5 password for LDP sessions:

Configuring an LDP MD5 Password for a Specified Neighbor

LDP looks first for a password between the device and neighbor that is configured with the **mpls ldp neighbor** [**vrf** *vrf-name*] *ip-address* **password** *pwd-string* command. If a password is configured with this command, LDP uses that password before checking passwords configured by other commands.

You must add a configuration command for each neighbor or peer for which you want password protection.

Before you begin

Identify the Label Distribution Protocol (LDP) neighbor or peer for which you want Message Digest 5 (MD5) password protection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp neighbor** [**vrf** *vrf-name*] *ip-address* **password** [**0** | **7**] *password-string*
4. **end**
5. **show mpls ldp neighbor** [**vrf** *vrf-name* | **all**] [*ip-address* | [*interface*] [**detail**] [**graceful-restart**]
6. **show mpls ldp neighbor** [**vrf** *vrf-name*] [*ip-address* | *interface*] **password** [**pending** | **current**]
7. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 3	**mpls ldp neighbor** [**vrf** *vrf-name*] *ip-address* **password** [**0** | **7**] *password-string*
	Example:
	`Device(config)# mpls ldp neighbor vrf vpn1 10.1.1.1 password nbrce1pwd`
		Configures a password key for computing MD5 checksums for the session TCP connection with the specified neighbor.
		• The **vrf** *vrf-name* keyword and argument specifies the virtual private network (VPN) routing and forwarding instance for the specified neighbor.
		• The *ip-address* argument specifies the router ID (IP address) that identifies a neighbor.
		• The [**0** | **7**] keywords specify whether the password that follows is encrypted:
		  • **0** specifies a clear-text (nonencrypted) password.
		  • **7** specifies a Cisco proprietary encrypted password.
		• The *password-string* argument defines the password key to be used for computing MD5 checksums for the session TCP connection with the specified neighbor.

Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 5	**show mpls ldp neighbor** [**vrf** *vrf-name* | **all**] [*ip-address* | [*interface*] [**detail**] [**graceful-restart**]
	Example:
	`Device# show mpls ldp neighbor vrf vpn1 detail`
		Displays the status of LDP sessions.
		• The **vrf** *vrf-name* keyword and argument displays the LDP neighbors for the specified VRF instance (*vrf-name*).
		• The **all** keyword displays LDP neighbor information for all VPNs, including those in the default routing domain.
		• The *ip-address* argument identifies the neighbor with the IP address for which you configured password protection.
		• The *interface* argument defines the LDP neighbors accessible over this interface.
		• The **detail** keyword displays information in long form, including password information for this neighbor. Here are the items displayed:
		  • An indication as to whether a password is mandatory for this neighbor (required or not required)
		  • The password source (neighbor, fallback or number [option number])

	Command or Action	Purpose
		• An indication as to whether the latest configured password for this neighbor is used by the TCP session (in use) or the TCP session uses an old password (stale) • The **graceful-restart** keyword displays per-neighbor graceful restart information.
Step 6	**show mpls ldp neighbor** [**vrf** *vrf-name*] [*ip-address* | *interface*] **password** [**pending** | **current**] **Example:** `Device# show mpls ldp neighbor vrf vpn1 password`	Displays password information used in established LDP sessions. • The **vrf** *vrf-name* keyword and argument displays the LDP neighbors for the specified VRF instance (*vrf-name*). • The *ip-address* argument identifies the neighbor with the IP address for which you configured password protection. • The *interface* argument defines the LDP neighbors accessible over this interface. • The **pending** keyword displays LDP sessions whose passwords are different from that in the current configuration. • The **current** keyword displays LDP sessions whose password is the same as that in current configuration. If you do not specify an optional keyword for this command, password information for all established LDP sessions is displayed.
Step 7	**show mpls ldp discovery** [**vrf** *vrf-name* | **all**] [**detail**] **Example:** `Device# show mpls ldp discovery vrf vpn1 detail`	Displays the status of the LDP discovery process. • The **vrf** *vrf-name* keyword and argument displays the neighbor discovery information for the specified VRF instance (*vrf-name*). • The **all** keyword displays LDP discovery information for all VPNs, including those in the default routing domain. • The **detail** keyword displays detailed information about all LDP discovery sources on a label switch router (LSR).

Configuring an LDP MD5 Password for LDP Sessions with Peers from a Specified VRF

This task provides you with Label Distribution Protocol (LDP) session protection with peers from a particular virtual routing and forwarding (VRF) instance or the global routing table. If you want a password requirement, you can use the **mpls ldp password required** command.

If only LDP sessions with a set of LDP neighbors need Message Digest 5 (MD5) protection, configure a standard IP access list that permits the desired set of LDP neighbors and denies the rest.

Before you begin

Identify LDP peers for which you want MD5 password protection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp** [**vrf** *vrf-name*] **password fallback** [**0** | **7**] *password*
4. **mpls ldp** [**vrf** *vrf-name*] **password required** [**for** *acl*]
5. **end**
6. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**mpls ldp** [**vrf** *vrf-name*] **password fallback** [**0** | **7**] *password*
	Example:
	`Device(config)# mpls ldp vrf vpn1 password fallback 0 vrfpwdvppn1`
		Configures an MD5 password for LDP sessions with peers.
		• The **vrf** *vrf-name* keyword and argument specifies a VRF configured on the label switch router (LSR).
		• The [**0** | **7**] keywords specify whether the password that follows is encrypted:
		  • **0** specifies a clear-text (nonencrypted) password.
		  • **7** specifies a Cisco proprietary encrypted password.
		• The *password* argument specifies the MD5 password to be used for the LDP sessions with peers whose connections are established through a named VRF or the global routing table.
		The example sets up an MD5 password for a VRF.

Step 4	**mpls ldp** [**vrf** *vrf-name*] **password required** [**for** *acl*] **Example:**	Specifies that LDP must use a password when establishing a session between LDP peers.

	Command or Action	Purpose
	`Device(config)# mpls ldp vrf vpn1 password required`	• The **vrf** *vrf-name* keyword and argument specifies a VRF configured on the LSR. • The **for** *acl* keyword and argument names an access list that specifies that a password is mandatory only for LDP sessions with neighbors whose LDP router IDs are permitted by the list. Only standard IP access lists can be used for the *acl* argument.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show mpls ldp discovery** [**vrf** *vrf-name* | **all**] [**detail**] **Example:** `Device# show mpls ldp discovery detail`	Displays the status of the LDP discovery process. • The **vrf** *vrf-name* keyword and argument displays the neighbor discovery information for the specified VPN routing and forwarding instance (*vrf-name*). • The **all** keyword displays LDP discovery information for all VPNs, including those in the default routing domain. • The **detail** keyword displays detailed information about all LDP discovery sources on an LSR. Use this command to verify that the password configuration is correct for all LDP neighbors.

Configuring an LDP MD5 Password for LDP Sessions with a Selected Group of Peers

If only Label Distribution Protocol (LDP) sessions with a selected group of peers need Message Digest 5 (MD5) protection, configure a standard IP access list that permits sessions with the desired group of peers (identified by LDP router IDs) and denies session with the rest. Configuring a password and password requirement for these neighbors or peers provides security by preventing LDP sessions from being established with unauthorized peers.

Before you begin

Identify the groups of peers for which you want MD5 password protection and define an access list that permits LDP sessions with the group of peers you require.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp** [**vrf** *vrf-name*] **password option** *number* **for** *acl* [**0** | **7**] *password*
4. **mpls ldp** [**vrf** *vrf-name*] **password required** [**for** *acl*]
5. **end**

6. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**mpls ldp** [**vrf** *vrf-name*] **password option** *number* **for** *acl* [**0** | **7**] *password*
	Example:
	`Device(config)# mpls ldp password option 25 for 10 aclpwdfor10`
		Configures an MD5 password for LDP sessions with neighbors whose LDP router IDs are permitted by a specified access list.
		• The **vrf** *vrf-name* keyword and argument specifies a virtual routing and forwarding (VRF) instance configured on the label switch router (LSR).
		• The *number* argument defines the order in which the access lists are evaluated in the determination of a neighbor password. The range is 1 through 32767.
		• The **for** *acl* keyword and argument specifies the name of the access list that includes the LDP router IDs of those neighbors for which the password applies. Only standard IP access list values (1 through 99) can be used for the *acl* argument.
		• The [**0** | **7**] keywords specify whether the password that follows is encrypted:
		  • **0** specifies a clear-text (nonencrypted) password.
		  • **7** specifies a Cisco proprietary encrypted password.
		• The *password* argument specifies the MD5 password to be used for the specified LDP sessions.

Step 4	**mpls ldp** [**vrf** *vrf-name*] **password required** [**for** *acl*] **Example:** `Device(config)# mpls ldp password required for 10`	Specifies that LDP must use a password when establishing a session between LDP peers. • The **vrf** *vrf-name* keyword and argument specifies a VRF configured on the LSR. • The **for** *acl* keyword and argument names an access list. The access list specifies a password is mandatory only for LDP sessions with neighbors whose LDP

	Command or Action	Purpose
		router IDs are permitted by the list. Only standard IP access lists can be used for the *acl* argument.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show mpls ldp discovery** [**vrf** *vrf-name* | **all**] [**detail**] **Example:** `Device# show mpls ldp discovery detail`	Displays the status of the LDP discovery process. • The **vrf** *vrf-name* keyword and argument displays the neighbor discovery information for the specified VPN routing and forwarding instance (*vrf-name*). • The **all** keyword displays LDP discovery information for all virtual private networks (VPNs), including those in the default routing domain. • The **detail** keyword displays detailed information about all LDP discovery sources on an LSR. Use this command to verify password configuration is correct for all LDP neighbors.

Verifying the LDP MD5 Configuration

Perform the following task to verify that the Label Distribution Protocol (LDP) Message Digest 5 (MD5) secure sessions are as you configured for all LDP neighbors.

SUMMARY STEPS

1. **enable**
2. **show mpls ldp discovery detail**
3. **show mpls ldp neighbor detail**
4. **show mpls ldp neighbor password** [**pending** | **current**]
5. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show mpls ldp discovery detail**

Verifies that the LDP MD5 password information is as you configured for each neighbor.

Example:

```
Device# show mpls ldp discovery detail

 Local LDP Identifier:
    10.1.1.1:0
    Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            Hello interval: 5000 ms; Transport IP addr: 10.1.1.1
            LDP Id: 10.4.4.4:0
              Src IP addr: 10.0.20.4; Transport IP addr: 10.4.4.4
              Hold time: 15 sec; Proposed local/peer: 15/15 sec
              Password: not required, none, stale
    Targeted Hellos:
        10.1.1.1 -> 10.3.3.3 (ldp): passive, xmit/recv
            Hello interval: 10000 ms; Transport IP addr: 10.1.1.1
            LDP Id: 10.3.3.3:0
              Src IP addr: 10.3.3.3; Transport IP addr: 10.3.3.3
              Hold time: 90 sec; Proposed local/peer: 90/90 sec
              Password: required, neighbor, in use
```

The Password field might display any of the following for the status of the password:

- Required or not required—Indicates whether password configuration is required.
- Neighbor, none, option #, or fallback—Indicates the password source when the password was configured.
- In use (current) or stale (previous)—Indicates the current LDP session password usage status.

Look at the output of the command to verify your configuration.

Step 3 **show mpls ldp neighbor detail**

Verifies that the password information for a neighbor is as you configured.

Example:

```
Device# show mpls ldp neighbor detail

    Peer LDP Ident: 10.3.3.3:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.3.3.3.11018 - 10.1.1.1.646
        Password: required, neighbor, in use
        State: Oper; Msgs sent/rcvd: 167/167; Downstream; Last TIB rev sent 9
        Up time: 02:24:02; UID: 5; Peer Id 3;
        LDP discovery sources:
          Targeted Hello 10.1.1.1 -> 10.3.3.3, passive;
            holdtime: 90000 ms, hello interval: 10000 ms
        Addresses bound to peer LDP Ident:
          10.3.3.3        10.0.30.3
        Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
    Peer LDP Ident: 10.4.4.4:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.4.4.4.11017 - 10.1.1.1.646
        Password: not required, none, stale
        State: Oper; Msgs sent/rcvd: 9/9; Downstream; Last TIB rev sent 9
        Up time: 00:05:35; UID: 6; Peer Id 1;
        LDP discovery sources:
          Ethernet1/0; Src IP addr: 10.0.20.4
            holdtime: 15000 ms, hello interval: 5000 ms
        Addresses bound to peer LDP Ident:
```

```
  10.0.40.4       10.4.4.4         10.0.20.4
Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
```

Step 4 **show mpls ldp neighbor password** [**pending** | **current**]

Verifies that LDP sessions are using the password configuration that you expect, either the same as or different from that in the current configuration. The **pending** keyword displays information for LDP sessions whose password is different from that in the current configuration. The **current** keyword displays information for LDP sessions whose password is the same as that in the current configuration.

Example:

```
Device# show mpls ldp neighbor password

    Peer LDP Ident: 10.4.4.4:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.4.4.4.11017 - 10.1.1.1.646
        Password: not required, none, stale
        State: Oper; Msgs sent/rcvd: 57/57
    Peer LDP Ident: 10.3.3.3:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.3.3.3.11018 - 10.1.1.1.646
        Password: required, neighbor, in use
        State: Oper; Msgs sent/rcvd: 216/215
Device# show mpls ldp neighbor password pending

    Peer LDP Ident: 10.4.4.4:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.4.4.4.11017 - 10.1.1.1.646
        Password: not required, none, stale
        State: Oper; Msgs sent/rcvd: 57/57
Device# show mpls ldp neighbor password current

    Peer LDP Ident: 10.3.3.3:0; Local LDP Ident 10.1.1.1:0
        TCP connection: 10.3.3.3.11018 - 10.1.1.1.646
        Password: required, neighbor, in use
        State: Oper; Msgs sent/rcvd: 216/215
```

This command displays password information used in established LDP sessions. If you do not enter an optional **pending** or **current** keyword for the command, password information for all established LDP sessions is displayed.

Step 5 **exit**

Returns to user EXEC mode. For example:

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS LDP MD5 Global Configuration

Example: Configuring an LDP MD5 Password for LDP Sessions for a Specified Neighbor

The following example shows how to configure a Label Distribution Protocol (LDP) Message Digest 5 (MD5) password for LDP sessions for a specified neighbor:

```
enable
configure terminal
mpls ldp vrf vpn1 10.1.1.1 password nbrscrtpwd
end
```

This sets up nbrscrtpwd as the password to use for LDP sessions for the neighbor whose LDP router ID is 10.1.1.1. Communication with this neighbor is through VRF vpn1.

Examples: Configuring an LDP MD5 Password for LDP Sessions with Peers from a Specified VRF

The following example shows how to configure a Label Distribution Protocol (LDP) Message Digest 5 (MD5) password for LDP sessions with peers from a specified virtual routing and forwarding (VRF) instance. The password vrfpwdvpn1 is configured for use with LDP peers that communicate using VRF vpn1. A password is required; otherwise, LDP tears down the session.

```
enable
configure terminal
mpls ldp vrf vpn1 password fallback vrfpwdvpn1
mpls ldp vrf vpn1 password required
end
```

The following example shows how to configure a password that is used for sessions for peers that communicate using the global routing table:

```
enable
configure terminal
mpls ldp password fallback vrfpwdvppn1
end
```

Example: Configuring an LDP MD5 Password for LDP Sessions with a Selected Group of Peers

The following example shows how to configure a Label Distribution Protocol (LDP) Message Digest 5 (MD5) password for LDP sessions with a selected group of peers. The required password aclpwdfor10 is configured for access list 10. Only those LDP router IDs permitted in access list 10 are required to use the password.

```
enable
configure terminal
mpls ldp password option 25 for 10 aclpwdfor10
mpls ldp password required for 10
end
```

Access list 10 might look something like this:

```
enable
configure terminal
access-list 10 permit 10.1.1.1
access-list 10 permit 10.3.3.3
access-list 10 permit 10.4.4.4
access-list 10 permit 10.1.1.1
access-list 10 permit 10.2.2.2
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP MD5 Global Configuration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Information for MPLS LDP MD5 Global Configuration

Table 93: Feature Information for MPLS LDP MD5 Global Configuration

Feature Name	Releases	Feature Information
MPLS LDP MD5 Global Configuration		The MPLS LDP MD5 Global Configuration feature provides enhancements to the Label Distribution Protocol (LDP) implementation of the Message Digest 5 (MD5) password. This feature allows you to enable LDP MD5 globally instead of on a per-peer basis. With this feature, you can set up password requirements for a set of LDP neighbors to help prevent unauthorized peers from establishing LDP sessions and to block spoofed TCP messages.

Feature Name	Releases	Feature Information
		The following commands were modified by this feature: **mpls ldp password fallback**, **mpls ldp password option**, **mpls ldp password required**, **show mpls ldp discovery**, **show mpls ldp neighbor**, **show mpls ldp neighbor password**.

Glossary

BGP—Border Gateway Protocol. An interdomain routing protocol that replaces External Gateway Protocol (EGP). BGP systems exchange reachability information with other BGP systems. BGP is defined by RFC 1163.

EGP—Exterior Gateway Protocol. An internet protocol for exchanging routing information between autonomous systems. EGP is documented in RFC 904. EGP is not to be confused with the general term exterior gateway protocol. EGP is an obsolete protocol that was replaced by Border Gateway Protocol (BGP).

CE device—customer edge device. A device that is part of a customer network and that interfaces to a provider edge (PE) device.

CSC—Carrier Supporting Carrier. A situation where one service provider allows another service provider to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

LDP—Label Distribution Protocol. A standard protocol between Multiprotocol Label Switching (MPLS)-enabled devices that is uses in the negotiation of the labels used to forward packets. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LDP peer—A label switch router (LSR) that is the receiver of label space information from another LSR. If an LSR has a label space to advertise to another LSR, or to multiple LSRs, one Label Distribution Protocol (LDP) session exists for each LSR (LDP peer) receiving the label space information.

MD5—Message Digest 5. A one-way hashing algorithm that produces a 128-bit hash. Both MD5 and Secure Hash Algorithm (SHA) are variations on MD4 and are designed to strengthen the security of the MD4 hashing algorithm. Cisco uses hashes for authentication within the IPSec framework. SNMP v2 uses MD5 for message authentication, to verify the integrity of the communication, to authenticate the message origin, and to check its timeliness.

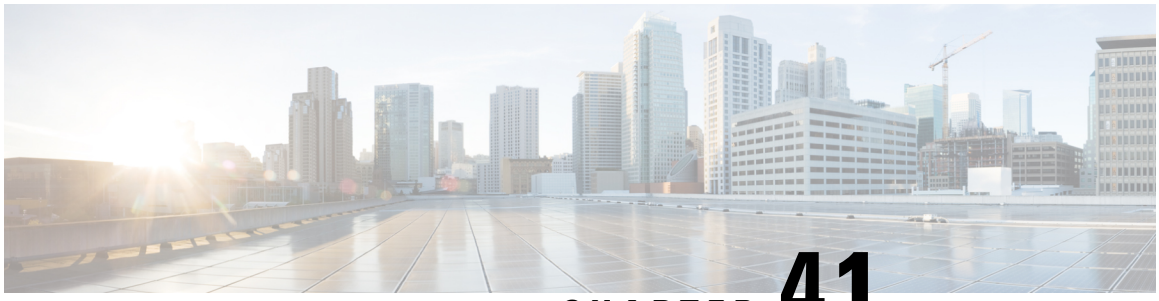
MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic through use of labels. Each label instructs the devices and the switches in the network where to forward a packet based on preestablished IP routing information.

PE device—provider edge device. A device that is part of a service provider's network connected to a customer edge (CE) device. All Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) processing occurs in the PE device.

VPN—Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic forwarded from one network to another. A VPN uses tunneling to encrypt all information at the IP level.

VRF—A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine

what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE device.



CHAPTER 41

MPLS LDP Lossless MD5 Session Authentication

The MPLS LDP Lossless MD5 Session Authentication feature enables a Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) session to be password-protected without tearing down and reestablishing the LDP session.

Prerequisites for MPLS LDP Lossless MD5 Session Authentication

The MPLS LDP Lossless MD5 Session Authentication feature is an enhancement to the MPLS LDP MD5 Global Configuration feature. Before configuring the MPLS LDP Lossless MD5 Session Authentication feature, see the “MPLS LDP MD5 Global Configuration” feature module for more information on how the message digest algorithm 5 (MD5) works with Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) to ensure that LDP segments remain properly protected.

Note The MPLS LDP Lossless MD5 Session Authentication feature must be configured before MPLS LDP is configured.

Configure the following features on the label switch router (LSR) before configuring the MPLS LDP Lossless MD5 Session Authentication feature:

- Distributed Cisco Express Forwarding
- Static or dynamic routing
- MPLS Virtual Private Network (VPN) routing and forwarding (VRFs) instances for MPLS VPNs
- MPLS LDP Lossless MD5 Session Authentication for the MPLS VPN VRFs

Note If a VRF is deleted, then the lossless MD5 session authentication for that VRF is automatically removed.

Restrictions for MPLS LDP Lossless MD5 Session Authentication

Message Digest 5 (MD5) protection applies to Label Distribution Protocol {LDP} sessions between peers. Tag Distribution Protocol (TDP) sessions between peers are not protected.

Information About MPLS LDP Lossless MD5 Session Authentication

How MPLS LDP Messages in MPLS LDP Lossless MD5 Session Authentication are Exchanged

Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) messages (discovery, session, advertisement, and notification messages) are exchanged between LDP peers through two channels:

- LDP discovery messages are transmitted as User Datagram Protocol (UDP) packets to the well-known LDP port.
- Session, advertisement, and notification messages are exchanged through a TCP connection established between two LDP peers.

The MPLS LDP Lossless MD5 Session Authentication feature allows an LDP session to be password-protected without tearing down and reestablishing the LDP session. The Message Digest 5 (MD5) password can be implemented and changed without interrupting the LDP session.

The Evolution of MPLS LDP MD5 Password Features

The initial version of Label Distribution Protocol (LDP) Message Digest 5 (MD5) protection allowed authentication to be enabled between two LDP peers and each segment sent on the TCP connection was verified between the peers. Authentication was configured on both LDP peers using the same password; otherwise, the peer session was not established. The **mpls ldp neighbor** command was issued with the **password** keyword. When MD5 protection was enabled, the device tore down the existing LDP sessions and established new sessions with the neighbor device.

An improved MD5 protection feature, called MPLS LDP MD5 Global Configuration, was later introduced that allowed LDP MD5 to be enabled globally instead of on a per-peer basis. Using this feature, password requirements for a set of LDP neighbors could be configured. The MPLS LDP MD5 Global Configuration feature also improved the ability to maintain the LDP session. The LDP session with a peer was not automatically torn down when the password for that peer was changed. The new password was implemented the next time an LDP session was established with the peer.

The MPLS LDP Lossless MD5 Session Authentication feature is based on the MPLS LDP MD5 Global Configuration feature. However, the MPLS LDP Lossless MD5 Session Authentication feature provides the following enhancements:

- Activate or change LDP MD5 session authentication without interrupting the LDP session.
- Configure multiple passwords, so one password can be used now and other passwords later.
- Configure asymmetric passwords, which allows one password to be used for incoming TCP segments and a different password to be used for outgoing TCP segments.
- Configure passwords so that they overlap for a period of time. This functionality is beneficial when the clocks on two label switch routers (LRS) are not synchronized.

These enhancements are available by using the **key-chain** command, which allows different key strings to be used at different times according to the keychain configuration.

Keychains Use with MPLS LDP Lossless MD5 Session Authentication

The MPLS LDP Lossless MD5 Session Authentication feature allows keychains to be used to specify different Message Digest 5 (MD5) keys to authenticate Label Distribution Protocol (LDP) traffic exchanged in each direction.

In the following example, three passwords are configured:

```
key chain ldp-pwd
 key 1
  key-string lab
  send-lifetime 10:00:00 Nov 2 2008 10:00:00 Dec 2 2008
  accept-lifetime 00:00:00 Jan 1 1970 duration 1
 key 2
  key-string lab2
  send-lifetime 00:00:00 Jan 1 1970 duration 1
  accept-lifetime 10:00:00 Nov 2 2008 10:00:00 Nov 17 2008
 key 3
  key-string lab3
  send-lifetime 00:00:00 Jan 1 1970 duration 1
  accept-lifetime 10:00:00 Nov 17 2008 10:00:00 Dec 2 2008
!
mpls ldp password option 1 for nbr-acl key-chain ldp-pwd
```

- Key 1 specifies the lab password. The **send-lifetime** command enables the lab password to authenticate the outgoing TCP segments from November 2, 2008, at 10:00:00 a.m. until December 2, 2008, at 10:00:00 a.m. The **accept-lifetime** command is configured so that the lab password is never used to authenticate incoming TCP segments. The **accept-lifetime** command enables the lab password for 1 second on January 1, 1970. By setting the date to the past and by enabling a duration of 1 second, the password for incoming TCP segments immediately expires. If the **accept-lifetime** command is omitted from the keychain configuration, then the password is always valid for incoming TCP segments.
- Key 2 and key 3 specify the lab2 and lab3 passwords, respectively. The **send-lifetime** commands enable the passwords for 1 second on January 1, 1970. By setting the date to the past and by enabling a duration of 1 second, the passwords for outgoing TCP segments immediately expire. If the **send-lifetime** commands are omitted from the keychain configuration, the passwords are always valid for outgoing TCP segments. The **accept-lifetime** commands for key 2 and key 3 enable the passwords to authenticate the incoming TCP segments from November 2, 2008, at 10:00:00 a.m. until November 17, 2008, at 10:00:00 a.m. and from November 17, 2008, at 10:00:00 a.m. until December 2, 2008, at 10:00:00 a.m., respectively.

Application of Rules to Overlapping Passwords

Overlapping passwords can be useful when two label switch routers (LSRs) have clocks that are not synchronized. The overlapping passwords provide a window to ensure that TCP packets are not dropped. The following rules apply to overlapping passwords:

- If the send-lifetime value for the next password begins before the send-lifetime value of the current password expires, the password with the shorter key ID is used during the overlap period. The send-lifetime value of the current password can be shortened by configuring a shorter send-lifetime value. Similarly, the send-lifetime value of the current password can be lengthened by configuring a longer send-lifetime value.
- If the accept-lifetime value for the next password begins before the accept-lifetime value of the current password expires, both the next password and the current password are used concurrently. The next password information is passed to TCP. If TCP fails to authenticate the incoming segments with the current password, it tries authenticating with the next password. If TCP authenticates a segment using the new password, it discards the current password and uses the new password from that point on.
- If a password for incoming or outgoing segments expires and no additional valid password is configured, one of the following actions take place:
  - If a password is required for the neighbor, the Label Distribution Protocol (LDP) drops the existing session.
  - If a password is not required for the neighbor, LDP attempts to roll over to a session that does not require authentication. This attempt also fails unless the password expires on both LSRs at the same time.

Password Rollover Period Guidelines

Both old and new passwords are valid during a rollover period. This ensures a smooth rollover when clocks are not synchronized between two Label Distribution Protocol (LDP) neighbors. When passwords are configured using a keychain, the rollover period is equal to the accept-lifetime overlap between two successive receive passwords.

The minimum rollover period (the duration between two consecutive Message Digest 5 (MD5) key updates) must be longer than the value of the LDP keepalive interval time to ensure an update of new MD5 authentication keys. If LDP session hold time is configured to its default value of 3 minutes, the LDP keepalive interval is 1 minute. The minimum rollover period should be 5 minutes. However, we recommend that the minimum rollover period is set to between 15 and 30 minutes.

To ensure a seamless rollover, follow these guidelines:

- Ensure that the local time on the peer label switch routers (LSRs) is the same before configuring the keychain.
- Check for error messages (TCP-6-BADAUTH) that indicate keychain misconfiguration.
- Validate the correct keychain configuration by checking for the following password messages:

```
%LDP-5-PWDCFG: Password configuration changed for 10.1.1.1:0
%LDP-5-PWDRO: Password rolled over for 10.1.1.1:0
```

Resolving LDP Password Problems

The Label Distribution Protocol (LDP) displays error messages when an unexpected neighbor attempts to open an LDP session, or the LDP password configuration is invalid. Some existing LDP debugs also display password information.

When a password is required for a potential LDP neighbor, but no password is configured for it, the label switch router (LSR) ignores LDP hello messages from that neighbor. When the LSR processes the hello message and tries to establish a TCP connection with the neighbor, it displays the error message and stops establishing the LDP session with the neighbor. The error is rate-limited and has the following format:

```
00:00:57: %LDP-5-PWD: MD5 protection is required for peer 10.2.2.2:0(glbl), no password
configured
```

When passwords do not match between LDP peers, TCP displays the following error message on the LSR that has the lower router ID; that is, the device that has the passive role in establishing TCP connections:

```
00:01:07: %TCP-6-BADAUTH: Invalid MD5 digest from 10.2.2.2(11051) to 10.1.1.1(646)
```

If one peer has a password configured and the other one does not, TCP displays the following error messages on the LSR that has a password configured:

```
00:02:07: %TCP-6-BADAUTH: No MD5 digest from 10.1.1.1(646) to 10.2.2.2(11099)
```

How to Configure MPLS LDP Lossless MD5 Session Authentication

Configuring MPLS LDP Lossless MD5 Session Authentication Using a Keychain

Perform the following task to configure the MPLS LDP Lossless MD5 Session Authentication feature using a keychain. Keychains allow a different key string to be used at different times according to the keychain configuration. Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) queries the appropriate keychain to obtain the current live key and key ID for the specified keychain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**permit** | **deny**} {*type-code wildcard-mask* | *ip-address mask*}
4. **key chain** *name-of-chain*
5. **key** *key-id*
6. **key-string** *string*
7. **accept-lifetime** {*start-time* | **local** *start-time*} {**duration** | *seconds end-time* | **infinite**}
8. **send-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds end-time* | **infinite**}
9. **exit**
10. **exit**
11. **mpls ldp** [**vrf** *vrf-name*] **password option** *number* **for** *acl* {**key-chain** *keychain-name* | [**0** | **7**] *password*}

12. **exit**
13. **show mpls ldp neighbor** [**vrf** *vrf-name* | **all**] [*ip-address* | *interface*] [**detail**] [**graceful-restart**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter the password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**permit** | **deny**} {*type-code wildcard-mask* | *ip-address mask*} **Example:** `Device(config)# access-list 10 permit 10.2.2.2`	Creates an access list.
Step 4	**key chain** *name-of-chain* **Example:** `Device(config)# key chain ldp-pwd`	Enables authentication for routing protocols and identifies a group of authentication keys. • Enters keychain configuration mode.
Step 5	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Identifies an authentication key on a keychain. • The *key-id* value must be a numeral. • Enters keychain key configuration mode.
Step 6	**key-string** *string* **Example:** `Device(config-keychain-key)# key-string pwd1`	Specifies the authentication string for a key. • The *string* value can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a numeral.
Step 7	**accept-lifetime** {*start-time* | **local** *start-time*} {**duration** | *seconds end-time* | **infinite**} **Example:** `Device(config-keychain-key)# accept-lifetime 10:00:00 Jan 13 2007 10:00:00 Jan 13 2009`	Specifies the time period during which the authentication key on a keychain can be used for verifying incoming TCP segments. The *start-time* argument identifies the time to start and the local *start-time* argument identifies the time to start in the local time zone. Both arguments have the same parameters:

	Command or Action	Purpose
		Note The time reference depends on the clock time zone configuration on the device. If no time zone configured, then the default time zone uses the Coordinated Universal Time (UTC) time. If it is configured, either the Eastern Standard Time (EST) or Pacific Standard Time (PST) time zone is used. • *hh:mm:ss* is the time format. • Enter the number of days from 1 to 31. • Enter the name of the month. • Enter the year from the present to 2035. Once the start time is entered, select from the following: • The **duration** keyword sets the key lifetime duration in seconds. • The *end-time* argument sets the time to stop. These parameters are the same as those used for the *start-time* argument. • The **infinite** keyword allows the accept-lifetime period to never expire. If the **no accept-lifetime** value is defined, the associated receive password is valid for authenticating incoming TCP segments.
Step 8	**send-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds end-time* | **infinite**} **Example:** `Device(config-keychain-key)# send-lifetime 10:00:00 Jan 13 2007 10:00:00 Jan 13 2009`	Specifies the time period during which the authentication key on a keychain can be used for verifying outgoing TCP segments. The *start-time* argument identifies the time to start and the local *start-time* argument identifies the time to start in the local time zone. Both arguments have the same parameters: **Note** The time reference depends on the clock time zone configuration on the device. If no time zone configured, then the default time zone uses the UTC time. If it is configured, either the EST or PST time zone is used. • *hh:mm:ss* is the time format. • Enter the number of days from 1 to 31. • Enter the name of the month. • Enter the year from 1993 to 2035. Once the start time is entered, select from the following:

	Command or Action	Purpose
		• The **duration** keyword sets the send lifetime duration in seconds. • The *end-time* argument sets the time to stop. These parameters are the same as those used for the *start-time* argument. • The **infinite** keyword allows the send lifetime period to never expire. If the **no send-lifetime** value is defined, the associated send password is valid for authenticating outgoing TCP segments.
Step 9	**exit** **Example:** `Device(config-keychain-key)# exit`	Returns to keychain configuration mode.
Step 10	**exit** **Example:** `Device(config-keychain)# exit`	Returns to global configuration mode.

Step 11	**mpls ldp** [**vrf** *vrf-name*] **password option** *number* **for** *acl* {**key-chain** *keychain-name* | [**0** | **7**] *password*}
	Example:
	`Device(config)# mpls ldp password option 1 for 10 keychain ldp-pwd`
		Configures a Message Digest 5 (MD5) password for LDP sessions with neighbors whose LDP router IDs are permitted by a specified access list.
		• The **vrf** *vrf-name* keyword-argument pair specifies a virtual routing and forwarding (VRF) configured on the label switch router (LSR).
		• The *number* argument defines the order in which the access lists are evaluated in the determination of a neighbor password. The range is 1 to 32767.
		• The **for** *acl* keyword and argument specifies the name of the access list that includes the LDP router IDs of those neighbors for which the password applies. Only standard IP access list values (1 to 99) can be used for the *acl* argument.
		• The **key-chain** *keychain-name* keyword and argument specifies the name of the keychain to use.
		• The **0** and **7** keywords specify whether the password that follows is hidden (encrypted);
		  • **0** specifies an unencrypted password.
		  • **7** specifies an encrypted password.
		• The *password* argument specifies the MD5 password to be used for the specified LDP sessions.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Step 13	**show mpls ldp neighbor** [**vrf** *vrf-name* | **all**] [*ip-address* | *interface*] [**detail**] [**graceful-restart**]
	Example:
	`Device# show mpls ldp neighbor detail`
		Displays the status of LDP sessions.
		• The **vrf** *vrf-name* keyword and argument displays the LDP neighbors for the specified VRF instance.
		• The *ip-address* argument identifies the neighbor with the IP address for which password protection is configured.
		• The *interface* argument identifies the LDP neighbors accessible over this interface.
		• The **detail** keyword displays information in long form, including password information for this neighbor. Here are the items displayed:
		  • An indication as to whether a password is mandatory for this neighbor (required/not required)
		  • The password source (neighbor/fallback/number [option number])
		  • An indication as to whether the latest configured password for this neighbor is used by the TCP session (in use) or the TCP session uses an old password (stale)
		• The **graceful-restart** keyword displays per-neighbor graceful restart information.

Enabling the Display of MPLS LDP Password Rollover Changes and Events

When a password is required for a neighbor, but no password is configured for the neighbor, the following debug message is displayed:

```
00:05:04: MDSym5 protection is required for peer 10.2.2.2:0(glbl), but no password configured.
```

To enable the display of events related to configuration changes and password rollover events, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp logging password configuration** [**rate-limit** *number*]

4. **mpls ldp logging password rollover** [**rate-limit** *number*]
5. **exit**
6. **debug mpls ldp transport events**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ldp logging password configuration** [**rate-limit** *number*] **Example:** `Device(config)# mpls ldp logging password configuration rate-limit 30`	Enables the display of events related to configuration changes. • The output displays events when a new password is configured or an existing password has been changed or deleted. A rate limit of 1 to 60 messages a minute can be specified.
Step 4	**mpls ldp logging password rollover** [**rate-limit** *number*] **Example:** `Device(config)# mpls ldp logging password rollover rate-limit 25`	Enables the display of events related to password rollover events. • Events are displayed when a new password is used for authentication or when authentication is disabled. A rate limit of 1 to 60 messages a minute can be specified.
Step 5	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 6	**debug mpls ldp transport events** **Example:** `Device# debug mpls ldp transport events`	Displays notifications when a session TCP Message Digest 5 (MD5) option is changed. • You can also use the **debug mpls ldp transport connections** command to display notifications when the MD5 option is changed.

Changing MPLS LDP Lossless MD5 Session Authentication Passwords

The MPLS LDP Lossless MD5 Session Authentication feature allows Message Digest 5 (MD5) passwords to be changed for Label Distribution Protocol (LDP) session authentication without having to close and reestablish an existing LDP session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ldp** [**vrf** *vrf-name*] **password rollover duration** *minutes*
4. **mpls ldp** [**vrf** *vrf-name*] **password fallback** {**key-chain** *keychain-name* | [**0** | **7**] *password*}
5. **no mpls ldp neighbor** [**vrf** *vrf-name*] *ip-address* **password** *password*
6. **exit**
7. **show mpls ldp neighbor** [**vrf** *vrf-name*] [*ip-address* | *interface*] [**detail**] [**graceful-restart**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter the password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls ldp** [**vrf** *vrf-name*] **password rollover duration** *minutes* **Example:** `Device(config)# mpls ldp password rollover duration 7`	Configures the duration before the new password takes effect. • The **vrf** *vrf-name* keyword and argument specifies a virtual routing and forwarding (VRF) configured on the label switch router (LSR). • The *minutes* argument specifies the number of minutes from 5 to 65535 before the password rollover occurs on this device.
Step 4	**mpls ldp** [**vrf** *vrf-name*] **password fallback** {**key-chain** *keychain-name*	[**0**	**7**] *password*} **Example:** `Device(config)# mpls ldp password fallback key-chain fallback`	Configures an MD5 password for LDP sessions with peers. • The **vrf** *vrf-name* keyword and argument specifies a VRF configured on the LSR. • The **key-chain** *keychain-name* keyword and argument specifies the name of the keychain used to specify the MD5 key that authenticates the exchange of bidirectional LDP traffic.

	Command or Action	Purpose

		• The **0** and **7** keywords specify whether the password that follows is hidden (encrypted);
		  • **0** specifies an unencrypted password.
		  • **7** specifies an encrypted password.
		• The *password* argument specifies the MD5 password to be used for the specified LDP sessions.

Step 5	**no mpls ldp neighbor** [**vrf** *vrf-name*] *ip-address* **password** *password* **Example:** `Device(config)# no mpls ldp neighbor 10.11.11.11 password lab1`	Disables the configuration of a password for computing MD5 checksums for the session TCP connection with the specified neighbor. • The **vrf** *vrf-name* keyword and argument optionally specifies the VRF instance for the specified neighbor. • The *ip-address* argument identifies the neighbor router ID. • The **password** *password* keyword and argument is necessary so that the device computes MD5 checksums for the session TCP connection with the specified neighbor.
Step 6	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Step 7	**show mpls ldp neighbor** [**vrf** *vrf-name*] [*ip-address* | *interface*] [**detail**] [**graceful-restart**]
	Example:
	`Device# show mpls ldp neighbor detail`
		Displays the status of LDP sessions.
		• The **vrf** *vrf-name* keyword argument displays the LDP neighbors for the specified VRF instance.
		• The *ip-address* argument identifies the neighbor with the IP address for which password protection is configured.
		• The *interface* argument lists the LDP neighbors accessible over this interface.
		• The **detail** keyword displays information in long form, including password information for this neighbor. Here are the items displayed:
		  • An indication as to whether a password is mandatory for this neighbor (required/not required)
		  • The password source (neighbor/fallback/number [option number])
		  • An indication as to whether the latest configured password for this neighbor is used by the TCP

	Command or Action	Purpose
		session (in use) or the TCP session uses an old password (stale) • The **graceful-restart** keyword displays per-neighbor graceful restart information.

Configuration Examples for MPLS LDP Lossless MD5 Session Authentication

Example: Configuring MPLS LDP Lossless MD5 Session Authentication Using a Keychain (Symmetrical)

The following example shows a configuration of two peer label switch routers (LSRs) that use symmetrical Message Digest 5 (MD5) keys:

LSR1

```
access-list 10 permit 10.2.2.2
mpls ldp password required for 10
mpls ldp password option 1 for 10 ldp-pwd
!
key chain ldp-pwd
 key 1
  key-string pwd1
  send-lifetime 10:00:00 Jan 1 2009 10:00:00 Feb 1 2009
  accept-lifetime 09:00:00 Jan 1 2009 11:00:00 Feb 1 2009
!
interface loopback0
  ip address 10.1.1.1 255.255.255.255
!
interface FastEthernet0/0/0
  ip address 10.0.1.1 255.255.255.254
  mpls label protocol ldp
  mpls ip
```

LSR2

```
access-list 10 permit 10.1.1.1
mpls ldp password required for 10
mpls ldp password option 1 for 10 ldp-pwd
!
key chain ldp-pwd
 key 1
  key-string pwd1
  send-lifetime 10:00:00 Jan 1 2009 10:00:00 Feb 1 2009
  accept-lifetime 09:00:00 Jan 1 2009 11:00:00 Feb 1 2009
!
interface loopback0
  ip address 10.2.2.2 255.255.255.255
!
```

```
interface FastEthernet0/0/0
  ip address 10.0.1.2 255.255.255.254
  mpls label protocol ldp
  mpls ip
```

Example: Configuring MPLS LDP Lossless MD5 Session Authentication Using a Keychain (Asymmetrical)

The following example shows a configuration of two peer label switch routers (LSRs) that use asymmetrical MD5 keys:

LSR1

```
access-list 10 permit 10.2.2.2
mpls ldp password required for 10
mpls ldp password option 1 for 10 ldp-pwd
!
key chain ldp-pwd
 key 1
  key-string pwd1
  accept-lifetime 00:00:00 Jan 1 2005 duration 1
  send-lifetime 10:00:00 Jan 1 2009 10:00:00 Feb 1 2009
 key 2
  key-string pwd2
  accept-lifetime 09:00:00 Jan 1 2009 11:00:00 Feb 1 2009
  send-lifetime 00:00:00 Jan 1 2005 duration 1
!
interface loopback0
  ip address 10.1.1.1 255.255.255.255
!
interface FastEthernet0/0/0
  ip address 10.0.1.1 255.255.255.254
  mpls label protocol ldp
  mpls ip
```

LSR2

```
access-list 10 permit 10.1.1.1
mpls ldp password required for 10
mpls ldp password option 1 for 10 ldp-pwd
!
key chain ldp-pwd
 key 1
  key-string pwd2
  accept-lifetime 00:00:00 Jan 1 2005 duration 1
  send-lifetime 10:00:00 Jan 1 2009 10:00:00 Feb 1 2009
 key 2
  key-string pwd1
  accept-lifetime 09:00:00 Jan 1 2009 11:00:00 Feb 1 2009
  send-lifetime 00:00:00 Jan 1 2005 duration 1
!
interface loopback0
  ip address 10.2.2.2 255.255.255.255
!
interface FastEthernet0/0/0
  ip address 10.0.1.2 255.255.255.254
  mpls label protocol ldp
  mpls ip
```

Examples: Changing MPLS LDP Lossless MD5 Session Authentication Password

The following example shows the existing password configuration for LSR A, LSR B, and LSR C:

LSR A Existing Configuration

```
mpls ldp router-id loopback0 force
mpls ldp neighbor 10.11.11.11 password lab1
mpls ldp neighbor 10.12.12.12 password lab1
mpls label protocol ldp
!
interface loopback0
ip address 10.10.10.10 255.255.255.255
!
interface FastEthernet1/0/0
ip address 10.2.0.1 255.255.0.0
mpls ip
!
interface FastEthernet2/0/0
ip address 10.0.0.1 255.255.0.0
mpls ip
```

LSR B Existing Configuration

```
mpls ldp router-id loopback0 force
mpls ldp neighbor 10.10.10.10 password lab1
mpls label protocol ldp
!
interface loopback0
ip address 10.11.11.11 255.255.255.255
!
interface FastEthernet1/0/0
ip address 10.2.0.2 255.255.0.0
mpls ip
```

LSR C Existing Configuration

```
mpls ldp router-id loopback0 force
mpls ldp neighbor 10.10.10.10 password lab1
mpls label protocol ldp
!
interface loopback0
ip address 10.12.12.12 255.255.255.255
!
interface FastEthernet2/0/0
ip address 10.0.0.2 255.255.0.0
mpls ip
!
```

The following example shows how the lossless password change is configured using the **mpls ldp password rollover duration** command for LSR A, LSR B, and LSR C so there is enough time to change all the passwords on all of the devices:

LSR A New Configuration

```
mpls ldp password rollover duration 10
mpls ldp password fallback lab2
no mpls ldp neighbor 10.11.11.11 password lab1
no mpls ldp neighbor 10.12.12.12 password lab1
```

LSR B New Configuration

```
mpls ldp password rollover duration 10
mpls ldp password fallback lab2
no mpls ldp neighbor 10.10.10.10 password lab1
```

LSR C New Configuration

```
mpls ldp password rollover duration 10
mpls ldp password fallback lab2
no mpls ldp neighbor 10.10.10.10 password lab1
```

After 10 minutes has elapsed, the password changes. The following system logging message for LSR A confirms that the password rollover was successful:

```
%LDP-5-PWDRO: Password rolled over for 10.11.11.11:0
%LDP-5-PWDRO: Password rolled over for 10.12.12.12:0
```

Examples: Changing MPLS LDP Lossless MD5 Session Authentication Password Using a Rollover Without Keychain

The MPLS LDP Lossless MD5 Session Authentication password can be changed in a lossless way (without tearing down an existing Label Distribution Protocol [LDP] session) by using a password rollover without a keychain.

The following example shows the existing password configuration for LSR A and LSR B:

LSR A Existing Configuration

```
mpls ldp router-id loopback0 force
mpls ldp neighbor 10.11.11.11 password lab1
mpls label protocol ldp
!
interface loopback0
ip address 10.10.10.10 255.255.255.255
!
interface FastEthernet1/0/0 ip address 10.2.0.1 255.255.0.0
mpls ip
```

LSR B Existing Configuration

```
mpls ldp router-id loopback0 force
mpls ldp neighbor 10.10.10.10 password lab1
mpls label protocol ldp
!
interface loopback0
ip address 10.11.11.11 255.255.255.255
```

```
!
interface FastEthernet1/0/0
ip address 10.2.0.2 255.255.0.0
mpls ip
```

The following example shows the new password configuration for LSR A and LSR B:

Note The rollover duration should be large enough so that the passwords can be changed on all impacted devices.

LSR A New Configuration

```
mpls ldp password rollover duration 10
mpls ldp neighbor 10.11.11.11 password lab2
```

LSR B New Configuration

```
mpls ldp password rollover duration 10
mpls ldp neighbor 10.10.10.10 password lab2
```

After 10 minutes (rollover duration), the password changes and the following system logging message confirms the password rollover at LSR A:

```
%LDP-5-PWDRO: Password rolled over for 10.11.11.11:0
```

Example: Changing MPLS LDP Lossless MD5 Session Authentication Password Using a Rollover with a Keychain

The MPLS LDP Lossless MD5 Session Authentication password can be changed in a lossless way by using a password rollover with a keychain. The following configuration example shows the new password keychain configuration for LSR A, LSR B, and LSR C, in which the new password is ldp-pwd.

In the example, the desired keychain is configured first. The first pair of keys authenticate incoming TCP segments (recv key) and compute Message Digest 5 (MD5) digests for outgoing TCP segments (xmit key). These keys should be the same keys as those currently in use; that is, in lab 1. The second recv key in the keychain should be valid after a few minutes. The second xmit key becomes valid at a future time.

Note The rollover duration should be large enough so that the passwords can be changed on all impacted devices.

LSR A New Configuration

```
mpls ldp password rollover duration 10
access-list 10 permit 10.11.11.11
access-list 10 permit 10.12.12.12
!
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
```

```
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
key 11
key-string lab2
send-lifetime 10:30:00 Jan 1 2009 10:30:00 Feb 1 2009
accept-lifetime 10:15:00 Jan 1 2009 10:45:00 Feb 1 2009
key 12
key-string lab3
send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009
accept-lifetime 10:15:00 Feb 1 2009 10:45:00 Mar 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
no mpls ldp neighbor 10.11.11.11 password lab1
no mpls ldp neighbor 10.12.12.12 password lab1
```

LSR B New Configuration

```
mpls ldp password rollover duration 10
access-list 10 permit 10.10.10.10
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
key 11
key-string lab2
send-lifetime 10:30:00 Jan 1 2009 10:30:00 Feb 1 2009
accept-lifetime 10:15:00 Jan 1 2009 10:45:00 Feb 1 2009
key 12
key-string lab3
send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009
accept-lifetime 10:15:00 Feb 1 2009 10:45:00 Mar 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
no mpls ldp neighbor 10.10.10.10 password lab1
```

LSR C New Configuration

```
mpls ldp password rollover duration 10
access-list 10 permit 10.10.10.10
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
key 11
key-string lab2
send-lifetime 10:30:00 Jan 1 2009 10:30:00 Feb 1 2009
accept-lifetime 10:15:00 Jan 1 2009 10:45:00 Feb 1 2009
key 12
key-string lab3
send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009
accept-lifetime 10:15:00 Feb 1 2009 10:45:00 Mar 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
no mpls ldp neighbor 10.10.10.10 password lab1
```

After 10 minutes, the password changes and the following system logging message confirms the password rollover at LSR A.

```
%LDP-5-PWDRO: Password rolled over for 10.11.11.11:0
%LDP-5-PWDRO: Password rolled over for 10.12.12.12:0
```

Examples: Changing MPLS LDP Lossless MD5 Session Authentication Password Using a Fallback Password with a Keychain

The MPLS LDP Lossless MD5 Session Authentication password can be changed in a lossless way by using a fallback password when doing a rollover with a keychain.

Note The fallback password is used only when there is no other keychain configured. If there is a keychain configured, then the fallback password is not used.

The following example shows the existing password configuration for LSR A, LSR B, and LSR C:

LSR A Existing Configuration

```
mpls ldp router-id loopback0 force
mpls label protocol ldp
!
interface loopback0
ip address 10.10.10.10 255.255.255.255
!
interface FastEthernet1/0/0
ip address 10.2.0.1 255.255.0.0
mpls ip
!
interface FastEthernet2/0/0
ip address 10.0.0.1 255.255.0.0
mpls ip
!
access-list 10 permit 10.11.11.11
access-list 10 permit 10.12.12.12
!
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
```

LSR B Existing Configuration

```
mpls ldp router-id loopback0 force
mpls label protocol ldp
!
interface loopback0
ip address 10.11.11.11 255.255.255.255
!
interface FastEthernet1/0/0
ip address 10.2.0.2 255.255.0.0
mpls ip
!
access-list 10 permit 10.10.10.10
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
```

```
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
```

LSR C Existing Configuration

```
mpls ldp router-id loopback0 force
mpls label protocol ldp
!
interface loopback0
ip address 10.12.12.12 255.255.255.255
!
interface FastEthernet2/0/0
ip address 10.0.0.2 255.255.0.0
mpls ip
!
access-list 10 permit 10.10.10.10
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
```

Note The fallback keychain is not used unless the keychain *ldp-pwd* is removed using the **no mpls ldp password option 5 for 10 key-chain ldp-pwd** command.

The following example shows the new configuration for LSR A, LSR B, and LSR C, where one keychain is configured with the name *ldp-pwd* and another keychain is configured with the name *fallback* for the fallback password.

Note The rollover duration should be large enough so that the passwords can be changed on all impacted devices.

LSR A New Configuration

```
mpls ldp password rollover duration 10
!
key chain fallback
key 10
key-string fbk1
!
mpls ldp password fallback key-chain fallback
!
no mpls ldp password option 5 for 10 key-chain ldp-pwd
```

LSR B New Configuration

```
mpls ldp password rollover duration 10
!
key chain fallback
key 10
```

```
key-string fbk1
!
mpls ldp password fallback key-chain fallback
!
no mpls ldp password option 5 for 10 key-chain ldp-pwd
```

LSR C New Configuration

```
mpls ldp password rollover duration 10
key chain fallback
key 10
key-string fbk1
!
mpls ldp password fallback key-chain fallback
!
no mpls ldp password option 5 for 10 key-chain ldp-pwd
```

After 10 minutes, the password changes and the following system logging message confirms the password rollover at LSR A:

```
%LDP-5-PWDRO: Password rolled over for 10.11.11.11:0
%LDP-5-PWDRO: Password rolled over for 10.12.12.12:0
```

Examples: Changing MPLS LDP Lossless MD5 Session Authentication Common Misconfiguration

The following sections describe common misconfiguration examples that can occur when the MPLS LDP Lossless MD5 Session Authentication password is migrated in a lossless way. Misconfigurations can lead to undesired behavior in a Label Distribution Protocol (LDP) session.

Examples: Incorrect Keychain LDP Password Configuration

Possible misconfigurations can occur when keychain-based commands are used with the **mpls ldp password option for key-chain** command. If the **accept-lifetime** or **send-lifetime** command is not specified in this configuration, then a misconfiguration can occur when more than two keys are in a keychain.

The following example shows an incorrect keychain configuration with three passwords for LSR A and LSR B in the keychain:

LSR A Incorrect Keychain LDP Password Configuration

```
access-list 10 permit 10.11.11.11
!
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
key 11
key-string lab2
send-lifetime 10:30:00 Jan 1 2009 10:30:00 Feb 1 2009
key 12
key-string lab3
send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
```

LSR B Incorrect Keychain LDP Password Configuration

```
access-list 10 permit 10.10.10.10
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
key 11
key-string lab2
send-lifetime 10:30:00 Jan 1 2009 10:30:00 Feb 1 2009
key 12
key-string lab3
send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
```

In the example, for both LSR A and LSR B, during the period of the third **send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009** command, all three configured keys are valid as receive keys, and only the last configured key is valid as a transmit key. The keychain resolution rules dictate that keys 10 and 11 are used as receive keys, and only the last key 12 can be used as the transmit key. Because the transmit and receive keys are mismatched, the Label Distribution Protocol (LDP) session will not stay active.

Note When more than two passwords are configured in a keychain, the configuration needs to have both **accept-lifetime** and **send-lifetime** commands configured correctly for effective rollovers.

The following example shows the correct keychain configuration with multiple passwords in the keychain:

LSR A Correct Keychain LDP Password Configuration

```
access-list 10 permit 10.11.11.11
!
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
key 11
key-string lab2
send-lifetime 10:30:00 Jan 1 2009 10:30:00 Feb 1 2009
accept-lifetime 10:15:00 Jan 1 2009 10:45:00 Feb 1 2009
key 12
key-string lab3
send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009
accept-lifetime 10:15:00 Feb 1 2009 10:45:00 Mar 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
```

LSR B Correct Keychain LDP Password Configuration

```
access-list 10 permit 10.10.10.10
key chain ldp-pwd
key 10
key-string lab1
send-lifetime 10:00:00 Jan 1 2009 10:30:00 Jan 1 2009
accept-lifetime 10:00:00 Jan 1 2009 10:45:00 Jan 1 2009
key 11
```

```
key-string lab2
send-lifetime 10:30:00 Jan 1 2009 10:30:00 Feb 1 2009
accept-lifetime 10:15:00 Jan 1 2009 10:45:00 Feb 1 2009
key 12
key-string lab3
send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009
accept-lifetime 10:15:00 Feb 1 2009 10:45:00 Mar 1 2009
!
mpls ldp password option 5 for 10 key-chain ldp-pwd
```

In the example above, for both LSR A and LSR B, during the period of the third **send-lifetime 10:30:00 Feb 1 2009 10:30:00 Mar 1 2009** command, only the last key 12 is valid as transmit and receive keys. Therefore, the LDP session remains active.

Avoiding Access List Configuration Problems

Use caution when modifying or deleting an access list. Any empty access list implies "permit any" by default. So when either the **mpls ldp password option for key-chain** command or the **mpls ldp password option** command is used for MPLS LDP MD5 session authentication, if the access list specified in the command becomes empty as a result of a modification or deletion, then all Label Distribution Protocol (LDP) sessions on the device expect a password. This configuration may cause undesired behavior in LDP sessions. To avoid this scenario, ensure that the proper access list is specified for each label switch router (LSR).

Examples: Changing MPLS LDP Lossless MD5 Session Authentication Using a Second Key to Avoid LDP Session Failure

The MPLS LDP Lossless MD5 Session Authentication feature works when a specified rollover period is configured. Typically, one rollover period overlaps the two accept lifetime values that are configured for two consecutive receive keys. The Label Distribution Protocol (LDP) process requests an update from the keychain manager for the latest valid transmit and receive keys once every minute. LDP compares the latest key set with the keys from the previous update in its database to determine if a key was removed, changed, or rolled over. When the rollover occurs, the LDP process detects the rollover and programs TCP with the next receive key.

The LDP session can fail if LDP is configured to use two keys for the MPLS LDP Lossless MD5 Session Authentication feature where the first key uses a send and accept lifetime value and the second key is not configured. The configuration creates a special case where there are two rollovers but there is only one rollover period.

The following sections provide an example of this problem and a solution:

Example: TCP Authentication and LDP Sessions Can Fail When a Second Rollover Period Is Missing

In the following configuration, the first rollover is from "secondpass" to "firstpass." The second rollover is from "firstpass" back to "secondpass." The only rollover period in this configuration is the overlapping between the "firstpass" and "secondpass." Because one rollover period is missing, LDP performs only the first rollover and not the second rollover, causing TCP authentication to fail and the Label Distribution Protocol (LDP) session to fail.

```
key chain ldp-pwd
 key 1
   key-string firstpass
   accept-lifetime 01:03:00 Sep 10 2009 01:10:00 Sep 10 2009
   send-lifetime 01:05:00 Sep 10 2009 01:08:00 Sep 10 2009
```

```
 key 2
   key-string secondpass
```

TCP authentication and LDP sessions can also fail if the second key has send and accept lifetime configured. In this case the accept lifetime of the first key is a subset of the accept lifetime of the second key. For example:

```
key chain ldp-pwd
 key 1
   key-string firstpass
   accept-lifetime 01:03:00 Sep 10 2009 01:10:00 Sep 10 2009
   send-lifetime 01:05:00 Sep 10 2009 01:08:00 Sep 10 2009
 key 2
   key-string secondpass
   accept-lifetime 01:03:00 Sep 9 2009 01:10:00 Sep 11 2009
   send-lifetime 01:05:00 Sep 9 2009 01:08:00 Sep 11 2009
```

Examples: Reconfigure a Keychain to Prevent TCP Authentication and LDP Session Failures

If the configuration needs to specify the last key in the keychain to always be valid, then configure the keychain to have at least two keys. Each key must be configured with both the send and accept lifetime period. For example:

```
key chain ldp-pwd
 key 1
   key-string firstpass
   accept-lifetime 01:03:00 Sep 10 2008 01:10:00 Sep 10 2008
   send-lifetime 01:05:00 Sep 10 2008 01:08:00 Sep 10 2008
 key 2
   key-string secondpass
   accept-lifetime 01:06:00 Sep 10 2008 01:17:00 Sep 10 2008
   send-lifetime 01:08:00 Sep 10 2008 01:15:00 Sep 10 2008
 key 3
   key-string thirdpass
```

If the configuration needs to specify the first keychain for the time interval, then switch to use the second key forever after that interval. This is done by configuring the start time for the second key to begin shortly before the end time of the first key, and by configuring the second key to be valid forever after that interval. For example:

```
key chain ldp-pwd
 key 1
   key-string firstpass
   accept-lifetime 00:03:00 Sep 10 2008 01:10:00 Sep 10 2008
   send-lifetime 00:05:00 Sep 10 2008 01:08:00 Sep 10 2008
 key 2
   key-string secondpass
   accept-lifetime 01:06:00 Sep 10 2008 infinite
   send-lifetime 01:08:00 Sep 10 2008 infinite
```

If the configuration needs to specify the two keys in the order of the second key, first key, and second key again, then specify three keys in that order with the proper rollover period. For example:

```
key chain ldp-pwd
 key 1
   key-string firstpass
   accept-lifetime 00:03:00 Sep 10 2008 01:10:00 Sep 10 2008
   send-lifetime 00:05:00 Sep 10 2008 01:08:00 Sep 10 2008
 key 2
   key-string secondpass
   accept-lifetime 01:06:00 Sep 10 2008 01:17:00 Sep 10 2008
```

```
  send-lifetime 01:08:00 Sep 10 2008 01:15:00 Sep 10 2008
key 3
  key-string firstpass
  accept-lifetime 01:13:00 Sep 10 2008 infinite
  send-lifetime 01:15:00 Sep 10 2008 infinite
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS Label Distribution Protocol	"MPLS Label Distribution Protocol" module in the *MPLS Label Distribution Protocol Configuration Guide*
LDP implementation enhancements for the MD5 password	"MPLS LDP MD5 Global Configuration" module in the *MPLS Label Distribution Protocol Configuration Guide*

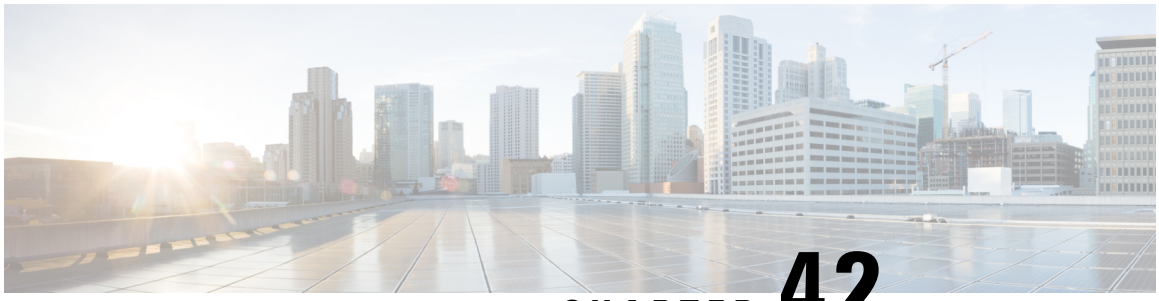
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP Lossless MD5 Session Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 42

MPLS LDP VRF-Aware Static Labels

This document explains how to configure the MPLS LDP VRF-Aware Static Labels feature and Multiprotocol Label Switching (MPLS) static labels. Virtual Private Network routing and forwarding (VRF)-aware static labels can be used at the edge of an MPLS Virtual Private Network (VPN), whereas MPLS static labels can be used only in the MPLS VPN provider core.

Information About MPLS LDP VRF-Aware Static Labels

Overview of MPLS Static Labels and MPLS LDP VRF-Aware Static Labels

Label switch routers (LSRs) dynamically learn the labels they should use to label-switch packets by means of the following label distribution protocols:

- Label Distribution Protocol (LDP), the Internet Engineering Task Force (IETF) standard used to bind labels to network addresses
- Resource Reservation Protocol (RSVP) used to distribute labels for traffic engineering (TE)
- Border Gateway Protocol (BGP) used to distribute labels for Multiprotocol Label Switching (MPLS) virtual private networks (VPNs)

The LSR installs the dynamically learned label into its Label Forwarding Information Base (LFIB).

You can configure static labels for the following purposes:

- To bind labels to IPv4 prefixes to support MPLS hop-by-hop forwarding through neighbor devices that do not implement LDP label distribution. MPLS static labels allow you to configure entries in the MPLS forwarding table and assign label values to forwarding equivalence classes (FECs) learned by LDP. You can manually configure an LSP without running an LDP between the endpoints.
- To create static cross connects to support MPLS label switched path (LSP) midpoints when neighbor devices do not implement the LDP or RSVP label distribution, but do implement an MPLS forwarding path.

- To statically bind a virtual routing and forwarding (VRF)-aware label on a provider edge (PE) device to a customer network prefix (VPN IPv4 prefix). VRF-aware static labels can be used with nonglobal VRF tables, so the labels can be used at the VPN edge. For example, with the Carrier Supporting Carrier (CSC) feature, the backbone carrier can assign specific labels to FECs it advertises to the edge devices of customer carriers. Then, backbone carrier can monitor backbone traffic coming from particular customer carriers for billing or other purposes. Depending on how you configure VRF-aware static labels, they are advertised one of the following ways:
  - By LDP between PE and customer edge (CE) devices within a VRF instance
  - In VPNv4 BGP in the service provider's backbone

Labels Reserved for Static Assignment

Before you can manually assign labels, you must reserve a range of labels to be used for the manual assignment. Reserving the labels ensures that the labels are not dynamically assigned.

How to Configure MPLS LDP VRF-Aware Static Labels

Reserving Labels to Use for MPLS Static Labels and MPLS LDP VRF-Aware Static Labels

To reserve the labels that are to be statically assigned so that the labels are not dynamically assigned, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label range** *minimum-value maximum-value* [**static** *minimum-static-value maximum-static-value*]
4. **exit**
5. **show mpls label range**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**mpls label range** *minimum-value maximum-value* [**static** *minimum-static-value maximum-static-value*] **Example:** `Device(config)# mpls label range 200 100000 static 16 199`	Reserves a range of labels for static labels assignment. The default is that no labels are reserved for static assignment. **Note** You might need to reload the device for the range of labels you reserve to take effect.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show mpls label range** **Example:** `Device# show mpls label range`	Displays information about the range of values for local labels, including those available for static assignment.

Configuring MPLS Static Labels in the MPLS VPN Provider Core

To configure MPLS static labels in the MPLS virtual private network (VPN) provider core, perform the following task.

MPLS static labels allow you to configure entries in the MPLS forwarding table and assign label values to forwarding equivalence classes (FECs) learned by the Label Distribution Protocol (LDP). You can manually configure a label switched path (LSP) without running a label distribution protocol between the endpoints. In MPLS VPN networks, static labels can be used only in the MPLS VPN provider core.

Note When static MPLS labels are used, LDP must be enabled on the interfaces, even though there is no need to establish an LDP session between devices.

Before you begin

- Globally enable Multiprotocol Label Switching (MPLS) on each label switch router (LSR).
- Enable Cisco Express Forwarding on each LSR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls static binding ipv4** *prefix mask* {*label* | **input** *label* | **output** *nexthop* {**explicit-null** | **implicit-null** | *label*}}
4. **exit**
5. **show mpls static binding ipv4**
6. **show mpls forwarding-table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls static binding ipv4** *prefix mask* {*label* | **input** *label* | **output** *nexthop* {**explicit-null** | **implicit-null** | *label*}} **Example:** `Device(config)# mpls static binding ipv4 10.2.2.0 255.255.255.255 input 17`	Specifies static binding of labels to IPv4 prefixes. • Specified bindings are installed automatically in the MPLS forwarding table as routing demands.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show mpls static binding ipv4** **Example:** `Device# show mpls static binding ipv4`	Displays the configured static labels.
Step 6	**show mpls forwarding-table** **Example:** `Device# show mpls forwarding-table`	Displays the static labels used for MPLS forwarding.

Configuring MPLS Static Cross Connects

You can configure MPLS static cross connects to support MPLS LSP midpoints when neighbor devices do not implement either the Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP) label distribution, but do implement an MPLS forwarding path.

Before you begin

- Globally enable Multiprotocol Label Switching (MPLS) on each label switch router (LSR).
- Enable Cisco Express Forwarding on each LSR.

Note

- MPLS static cross-connect labels remain in the Label Forwarding Information Base (LFIB) even if the device to which the entry points goes down.
- MPLS static cross-connect mappings remain in effect even with topology changes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls static crossconnect** *inlabel out-interface nexthop* {*outlabel* | **explicit-null** | **implicit-null**}
4. **end**
5. **show mpls static crossconnect**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls static crossconnect** *inlabel out-interface nexthop* {*outlabel*	**explicit-null**	**implicit-null**} **Example:** `Device(config)# mpls static crossconnect 45 pos5/0 45 explicit-null`	Specifies static cross connects. **Note** The *nexthop* argument is required for multiaccess interfaces.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show mpls static crossconnect** **Example:** `Device# show mpls static crossconnect`	Displays the configured static cross connects.

Configuring MPLS LDP VRF-Aware Static Labels at the Edge of the VPN

You can statically bind a virtual routing and forwarding (VRF)-aware label on a provider edge (PE) device to a customer network prefix (VPN IPv4 prefix). VRF-aware static labels can be used with nonglobal VRF tables, so the labels can be used at the VPN edge.

Before you begin

- Globally enable Multiprotocol Label Switching (MPLS) on each label switch router (LSR).
- Enable Cisco Express Forwarding on each LSR.
- Ensure the MPLS virtual private network (VPN) is configured.
- Ensure that the provider network has the MPLS Label Distribution Protocol (LDP) installed and running.

Note The MPLS LDP VRF-Aware Static Labels feature is supported only with MPLS VPN Carrier Supporting Carrier networks that use MPLS LDP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls static binding ipv4 vrf** *vpn-name prefix mask* {**input** *label* | *label*}
4. **exit**
5. **show mpls static binding ipv4 vrf** *vpn-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls static binding ipv4 vrf** *vpn-name prefix mask* {**input** *label* | *label*} **Example:** `Device(config)# mpls static binding ipv4 vrf vpn100 10.2.0.0 255.255.0.0 input 17`	Binds a prefix to a local label. • Specified bindings are installed automatically in the MPLS forwarding table as routing demands. **Note** You must configure the MPLS VPN and VRFs before creating VRF-aware static labels.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show mpls static binding ipv4 vrf** *vpn-name* **Example:** `Device(config)# show mpls static binding ipv4 vrf vpn100`	Displays the configured MPLS static bindings.

Troubleshooting Tips

To display information related to static binding events, use the **debug mpls static binding vrf** command.

Configuration Examples for MPLS LDP VRF-Aware Static Labels

Example: Reserving Labels to Use for MPLS Static Labels and MPLS LDP VRF-Aware Static Labels

In the following example, the **mpls label range** command reserves a generic range of labels from 200 to 100000 and configures a static label range of 16 to 199:

```
Device(config)# mpls label range 200 100000 static 16 199
% Label range changes take effect at the next reload.
```

In this example, the output from the **show mpls label range** command indicates that the new label ranges do not take effect until a reload occurs:

```
Device# show mpls label range

Downstream label pool: Min/Max label: 16/100000
   [Configured range for next reload: Min/Max label: 200/100000]
Range for static labels: Min/Max/Number: 16/199
```

In the following output, the **show mpls label range** command, executed after a reload, indicates that the new label ranges are in effect:

```
Device# show mpls label range

Downstream label pool: Min/Max label: 200/100000
Range for static labels: Min/Max/Number: 16/199
```

Example: Configuring MPLS Static Labels in the MPLS VPN Provider Core

The following example configures input and output labels for several prefixes:

```
Device(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55
Device(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 output 10.0.0.66 167
Device(config)# mpls static binding ipv4 10.66.0.0 255.255.0.0 input 17
Device(config)# mpls static binding ipv4 10.66.0.0 255.255.0.0 output 10.13.0.8 explicit-null
```

The **show mpls static binding ipv4** command displays the configured static labels:

```
Device# show mpls static binding ipv4

10.0.0.0/8: Incoming label: 55
  Outgoing labels:
      10.0.0.66   167
10.66.0.0/24: Incoming label: 17
  Outgoing labels:
     10.13.0.8  explicit-null
```

Example: Configuring MPLS LDP VRF-Aware Static Labels at the VPN Edge

In the following example, the **mpls static binding ipv4 vrf** command configures static label bindings. They also configure input (local) labels for various prefixes.

```
Device(config)# mpls static binding ipv4 vrf vpn100 10.0.0.0 10.0.0.0 55
Device(config)# mpls static binding ipv4 vrf vpn100 10.66.0.0 255.255.0.0 input 17
```

In the following output, the **show mpls static binding ipv4 vrf** command displays the configured VRF-aware static bindings:

```
Device# show mpls static binding ipv4 vrf vpn100
 10.0.0.0/8:   (vrf: vpn100)  Incoming label: 55
  Outgoing labels:  None
10.66.0.0/16:   (vrf: vpn100)  Incoming label: 17
  Outgoing labels:  None
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS VPN CSC with LDP and IGP	"MPLS VPN Carrier Supporting Carrier Using LDP and IGP" module in the *MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide*

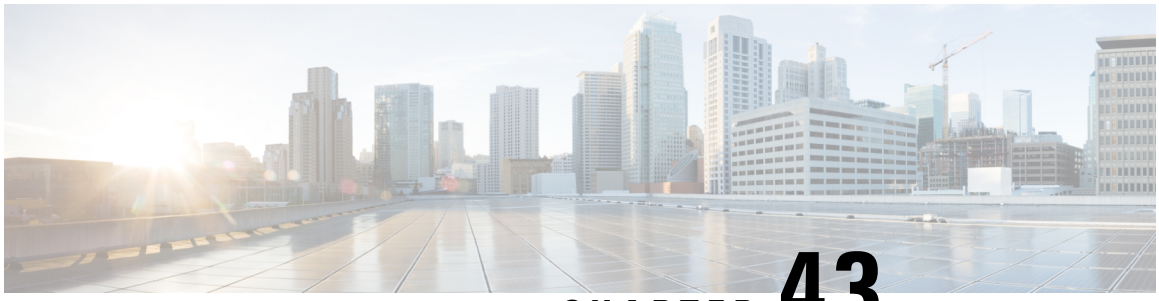
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP VRF-Aware Static Labels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 43

MPLS LDP Entropy Label Support

Load balancing is an important tool for engineering traffic across an MPLS network. The MPLS LDP Entropy Label Support feature describes methods of improving load balancing across MPLS networks using entropy labels.

Information About MPLS LDP Entropy Label Support

Overview of MPLS LDP Entropy Label

Prior to the MPLS LDP Entropy Label Support feature, at each MPLS hop, a deep packet inspection (DPI) is performed to determine the load balancing. With the introduction of MPLS LDP Entropy Label Support feature, at the ingress interface where MPLS is encapsulated, a DPI is performed to generate an entropy label. At the transit node, there is no need for DPI. The load balancing is done on the entropy label.

The entropy label provides ways of improving load balancing by eliminating the need for DPI at transit Label Switching Routers (LSRs). To eliminate DPI, the ingress LSR of an MPLS label switched path extracts the appropriate keys from a given packet, inputs them to the load balancing function, places the result in an additional label termed entropy label, and as a part of the MPLS label stack, the LSR pushes onto that packet. The transit LSRs use the label stack of the MPLS packet to perform load balancing.

If the transit LSR does not support LDP Entropy Label Capability (ELC), the transit LSR will propagate the label mapping message with the ELC flag so that the ingress LSR will add the entropy label. Any transit LSRs that do not support entropy label will ignore the entropy label and will load balance based on traditional DPI techniques.

Benefits of MPLS LDP Entropy Label Support

Each transit LSR along the path of a given LSP has to try to infer the underlying protocol within an MPLS packet to extract appropriate keys for load balancing. If the transit LSR is unable to infer the MPLS packet's protocol, it will use the topmost (or all) MPLS labels in the label stack as keys for the load-balancing function. The result may be an extremely inequitable distribution of traffic across equal cost paths exiting a LSR. This

is because MPLS labels are fairly coarse-grained forwarding labels that describe a next hop, or provide some demultiplexing or forwarding function, and do not describe the packet's underlying protocol.

For example, an ingress LSR, a Provider Edge (PE) device has detailed knowledge of a packet's contents, typically through a priority configuration of the encapsulations that are expected at a given Provider Edge-Customer Edge interface (IPv4, IPv6, VPLS, etc.), thus ensuring a more even distribution of load balancing for a given flow instead of overloading any particular path. The entropy label is protocol independent, it provides a unified way of load balancing without looking into the protocol header.

LDP Entropy Label Capability Signaling

Entropy Label Capability (ELC) is signaled from egress provider-edge (PE) device to ingress PE device. The egress LSR for a FEC initiates ELC by adding ELC Type Length Value (TLV), if configured, in the label mapping message that it sends to the peers. The presence of the ELC TLV in a label mapping message indicates to ingress LSRs that the egress LSR can process entropy labels for the associated LDP tunnel.

If the entire downstream peers have signaled entropy capability, ELC is propagated, else, it is not added in the label mapping message, and regular load balancing is used.

By configuring entropy labels, ingress LSR enables pushing of Entropy Label Indicator (ELI)/Entropy Label (EL). The ELI/EL value is pushed onto the label stack, if EL is enabled and downstream LSP is EL capable. The LDP processes the ELC TLV and sets EL capability flag accordingly for each LSP.

At the transit and egress LSR, LDP signals ELC, which indicates the ability to process entropy labels to upstream peers if ELC is received from downstream peers for all the next hops in a route.

How to Configure MPLS LDP Entropy Label Support

Enabling MPLS LDP Entropy Label Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls ldp entropy-label** [*label-value*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**mpls ip** **Example:** `Device(config)# mpls ip`	Configures MPLS hop-by-hop forwarding globally. • The **mpls ip** command is enabled by default; you do not have to specify this command. • Globally enabling MPLS forwarding does not enable it on the device interfaces. You must enable MPLS forwarding on the interfaces as well as for the device.
Step 4	**mpls ldp entropy-label** [*label-value*] **Example:** `Device(config)# mpls ldp entropy-label 7`	Enables entropy label on the egress PE device.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Verifying MPLS LDP Entropy Label Support

SUMMARY STEPS

1. **enable**
2. **show mpls ldp bindings** [*ip-address* | *mask*] [**detail**]
3. **show mpls infrastructure lfd lte** [*label-value*]
4. **show mpls forwarding-table** [*label-value*]
5. **ping mpls ipv4** [*ip-address* | *mask*] **entropy-label** [*label-value*]
6. **traceroute mpls** [*ip-address* | *mask*] **entropy-label** [*label-value*] [**verbose**]
7. **traceroute mpls multipath** [*ip-address* | *mask*] **entropy-label** [*label-value*] [**verbose**]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show mpls ldp bindings** [*ip-address* | *mask*] [**detail**]

Displays the total counts of advertised-to and remote-binding peers in IP address sort order, with remote bindings in tabular format.

Example:

```
Device# show mpls ldp bindings 172.16.0.0/32 detail

lib entry: 172.16.0.0/32, rev 59, chkpt: none, elc
  local binding: label: 20001 (owner LDP)
    Advertised to:
    1.1.1.1:0                    3.3.3.3:0
remote binding: lsr: 1.1.1.1:0, label: 10003 checkpointed, elc
remote binding: lsr: 3.3.3.3:0, label: 30004 checkpointed, elc
```

Step 3 **show mpls infrastructure lfd lte** [*label-value*]

Displays information about the Label Forwarding Database (LFD).

Example:

```
Device# show mpls infrastructure lfd lte 7

7 [table 0]
    ldm: No LDM, LTE Broker
    flags: nsf (0x1)
    plist: NONIP (0x7F45AAD23120) entropy
    output chain:
       label implicit-null()
       entropy label indicator
```

Step 4 **show mpls forwarding-table** [*label-value*]

Displays the MPLS packets that are forwarded with an entropy label indicator label (value of 7).

Example:

```
Device# show mpls forwarding-table 7

    Local      Outgoing     Prefix           Bytes Label        Outgoing     Next Hop
    Label      Label        or Tunnel Id     Switched           Interface
    7          Pop          Label entropy    7468               Exception
```

Step 5 **ping mpls ipv4** [*ip-address* | *mask*] **entropy-label** [*label-value*]

Displays the MPLS label switched path (LSP) connectivity.

Example:

```
Device# ping mpls ipv4 172.16.0.0/32 entropy-label

Sending 5, 72-byte MPLS Echos to 172.16.0.0/32,
           timeout is 2 seconds, send interval is 0 msec,
           over default entropy label:
Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
 Total Time Elapsed 4 ms
```

Example:

```
Device# ping mpls ipv4 172.16.0.0/32 entropy-label 20

Sending 5, 72-byte MPLS Echos to 172.16.0.0/32,
           timeout is 2 seconds, send interval is 0 msec,
           over entropy label 20:
Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
 Total Time Elapsed 4 ms
```

Step 6 **traceroute mpls** [*ip-address* | *mask*] **entropy-label** [*label-value*] [**verbose**]

Displays MPLS label switched path (LSP) routes that the packets actually take when traveling to their destinations.

Example:

```
Device# traceroute mpls ipv4 172.16.0.0/32 entropy-label verbose

Tracing MPLS Label Switched Path to 172.16.0.0/32, timeout is 2 seconds
       over entropy, default start label
Type escape sequence to abort.

0 192.168.2.20 192.168.2.30 MRU 1500 [Labels: 20001/7/524288 Exp: 0/0/0]
L 1 192.168.2.20 192.168.22.30 MRU 1500 [Labels: 30004/7/524288 Exp: 0/0/0] 1 ms, ret
code 8
L 2 192.168.2.20 192.168.33.30 MRU 1500 [Labels: implicit-null/7/524288 Exp: 0/0/0] 2
ms, ret code 8
! 3 192.168.43.2 1 ms, ret code 3
```

Example:

```
Device# traceroute mpls ipv4 172.16.0.0/32 entropy-label 20 verbose

Tracing MPLS Label Switched Path to 172.16.0.0/32, timeout is 2 seconds
       over entropy, start label 20
Type escape sequence to abort.
0 192.168.12.1 192.168.12.2 MRU 1500 [Labels: 20001/7/20 Exp: 0/0/0]
L 1 192.168.12.2 192.168.22.2 MRU 1500 [Labels: 30004/7/20 Exp: 0/0/0] 1 ms, ret code
8
L 2 192.168.23.2 192.168.42.2 MRU 1500 [Labels: implicit-null/7/20 Exp: 0/0/0] 1 ms,
ret code 8
! 3 192.168.43.2 1 ms, ret code 3
```

Step 7 **traceroute mpls multipath** [*ip-address* | *mask*] **entropy-label** [*label-value*] [**verbose**]

Displays multiple MPLS label switched path (LSP) routes from an egress device to an ingress device.

Example:

```
Device#traceroute mpls multipath ipv4 172.16.0.0/32 entropy-label verbose

Starting LSP Multipath Traceroute for 172.16.0.0/32
```

```
     over entropy, default start label

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
LL!
Path 0 found,
output interface Et0/0 nexthop 10.10.1.2
source 10.10.1.1 destination 127.0.0.0
  0 10.10.1.1 10.10.1.2 MRU 1500 [Labels: 19/7/524288 Exp: 0/0/0] multipaths 0
L 1 10.10.1.2 10.10.2.2 MRU 1500 [Labels: 23/7/524289 Exp: 0/0/0] ret code 8 multipaths 2
L 2 10.10.2.2 10.10.5.2 MRU 1500 [Labels: implicit-null/7/524289 Exp: 0/0/0] ret code 8 multipaths
1
! 3 10.10.5.2, ret code 3 multipaths 0
L!
Path 1 found,
output interface Et0/0 nexthop 10.10.1.2
source 10.10.1.1 destination 127.0.0.0
  0 10.10.1.1 10.10.1.2 MRU 1500 [Labels: 19/7/524288 Exp: 0/0/0] multipaths 0
L 1 10.10.1.2 10.10.3.2 MRU 1500 [Labels: 16/7/524288 Exp: 0/0/0] ret code 8 multipaths 2
L 2 10.10.3.2 10.10.8.2 MRU 1500 [Labels: implicit-null/7/524288 Exp: 0/0/0] ret code 8 multipaths
1
! 3 10.10.8.2, ret code 3 multipaths 0

Paths (found/broken/unexplored) (2/0/0)
Echo Request (sent/fail) (5/0)
Echo Reply (received/timeout) (5/0)
Echo Reply (received/timeout) (5/0)
Total Time Elapsed 18 ms
```

Example:

```
Device#traceroute mpls multipath ipv4 172.16.0.0/0 entropy-label 700 verbose

Starting LSP Multipath Traceroute for 172.16.0.0/32
     over entropy, start label 700
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LL!
Path 0 found,
output interface Et0/0 nexthop 10.10.1.2
source 10.10.1.1 destination 127.0.0.0
  0 10.10.1.1 10.10.1.2 MRU 1500 [Labels: 19/7/700 Exp: 0/0/0] multipaths 0
L 1 10.10.1.2 10.10.2.2 MRU 1500 [Labels: 23/7/701 Exp: 0/0/0] ret code 8 multipaths 2
L 2 10.10.2.2 10.10.5.2 MRU 1500 [Labels: implicit-null/7/701 Exp: 0/0/0] ret code 8 multipaths 1
! 3 10.10.5.2, ret code 3 multipaths 0
L!
Path 1 found,
```

```
output interface Et0/0 nexthop 10.10.1.2
source 10.10.1.1 destination 127.0.0.0
  0 10.10.1.1 10.10.1.2 MRU 1500 [Labels: 19/7/700 Exp: 0/0/0] multipaths 0
L 1 10.10.1.2 10.10.3.2 MRU 1500 [Labels: 16/7/700 Exp: 0/0/0] ret code 8 multipaths 2
L 2 10.10.3.2 10.10.8.2 MRU 1500 [Labels: implicit-null/7/700 Exp: 0/0/0] ret code 8 multipaths 1
! 3 10.10.8.2, ret code 3 multipaths 0
Paths (found/broken/unexplored) (2/0/0)
Echo Request (sent/fail) (5/0)
Echo Reply (received/timeout) (5/0)
Echo Reply (received/timeout) (5/0)
Total Time Elapsed 47 ms
```

Additional References for MPLS LDP Entropy Label Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS Multiprotocol Label Switching commands	Cisco IOS Multiprotocol Label Switching Command Reference

RFCs

RFC	Title
RFC 6790	*The Use of Entropy Labels in MPLS Forwarding*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS LDP Entropy Label Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

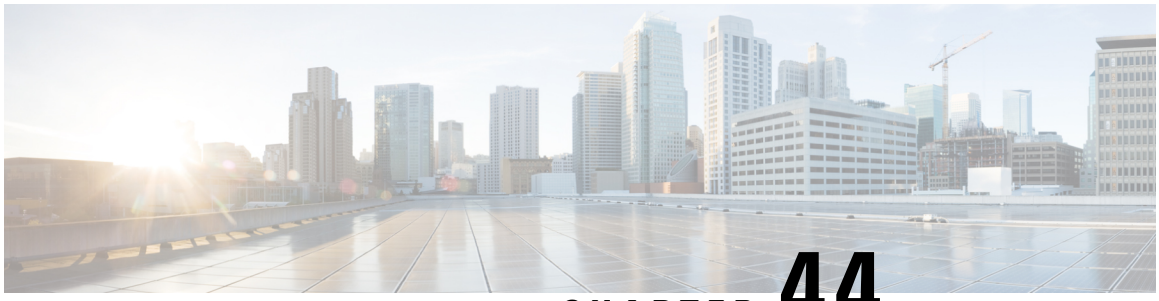
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 94: Feature Information for MPLS LDP Entropy Label Support

Feature Name	Releases	Feature Information
MPLS LDP Entropy Label Support		The MPLS LDP Entropy Label Support describes ways of improving load balancing across MPLS networks using the concept of entropy labels. The following commands were introduced or modified: **mpls ldp entropy-label, ping mpls, show mpls forwarding-table, show mpls infrastructure lfd lte, show mpls ldp bindings, traceroute mpls, traceroute mpls multipath**.

PART V

MPLS Layer 2 VPNs



CHAPTER 44

L2VPN Protocol-Based CLIs

The L2VPN Protocol-Based CLIs feature provides a set of processes and an improved infrastructure for developing and delivering Cisco IOS software on various Cisco platforms. This feature introduces new commands and modifies or replaces existing commands to achieve a consistent functionality across Cisco platforms and provide cross-Operating System (OS) support.

Information About L2VPN Protocol-Based CLIs

Overview of L2VPN Protocol-Based CLIs

The L2VPN Protocol-Based CLIs feature introduces new commands and modifies or replaces existing commands to achieve a consistent functionality across Cisco platforms and provide cross-Operating System (OS) support.

Note The new, updated, and replacement commands are available in Cisco IOS XE Release 3.7S and Cisco IOS Release 15.3(1)S. However, the legacy commands that are being replaced will be deprecated in later releases.

Benefits of L2VPN Protocol-Based CLIs

The L2VPN Protocol-Based CLIs feature provides the following benefits:

- Consistent user experience across different operating systems.
- Consistent configuration for all Layer 2 VPN (L2VPN) scenarios.
- Enhanced functionality that is achieved by configuring pseudowires as virtual interfaces and monitoring the pseudowires as physical ports.
- Feature configuration such as quality of service (QoS) service policies on individual pseudowires .

- Redundant pseudowire configuration that is independent of the primary pseudowire to provide enhanced high availability.

These benefits are achieved through the following enhancements:

- New service contexts can be created for point-to-point and multipoint Layer 2 services by using the new L2VPN cross connect and L2VPN virtual forwarding interface (VFI) contexts.
  - The L2VPN cross connect context is used for configuring point-to-point pseudowires, pseudowire stitching, and local switching (hair pinning). Ethernet interfaces , Ethernet Flow Points (EFP), ATM interfaces and WAN interfaces (PPP,HDLC,Serial), and pseudowire interfaces can be defined as members of an L2VPN cross connect context.
  - The L2VPN VFI context instantiates Virtual Private LAN Services (VPLS) VFI for multipoint scenarios. Pseudowires can be defined as members of an L2VPN VFI context.
  - Bridge domains are used for multipoint scenarios. EFPs, pseudowires, or VFIs can be configured as members of a bridge domain. Pseudowires can be configured as member of a VFI. The VFI can be configured as a member of a .
- New port contexts can be created (dynamically or manually) for pseudowires by using the pseudowire interface.
- Pseudowire customization can be achieved using interface templates and pseudowire interfaces that are applied to L2VPN context members. Pseudowire customizations include following features:
  - Encapsulation type
  - Control word
  - Maximum Transmission Unit (MTU)
  - Pseudowire signaling type
  - Tunnel selection
- Interworking and redundancy group service attributes can be configured under the L2VPN service context. The redundancy groups are configured independently from the primary pseudowire, which helps achieve zero traffic interruptions while adding, modifying, or deleting backup pseudowires.

L2VPN Protocol-Based CLI Changes

The following commands are introduced in Cisco IOS XE Release 3.7S, Cisco IOS Release 15.3(1)S, and Cisco IOS Release 15.4(1)S:

- **debug l2vpn pseudowire**
- **l2vpn**
- **l2vpn pseudowire static-oam class**
- **monitor event-trace l2vpn**
- **show interface pseudowire**
- **show l2vpn service**

- **shutdown (MPLS)**
- **vc**

The following commands are modified in Cisco IOS XE Release 3.7S and Cisco IOS Release 15.3(1)S:

- **auto-route-target**
- **bridge-domain parameterized vlan**
- **debug condition xconnect fib**
- **debug condition xconnect interface**
- **debug condition xconnect peer**
- **debug condition xconnect segment**
- **description**
- **encapsulation (MPLS)**
- **forward permit l2protocol all**
- **interworking**
- **l2vpn subscriber authorization group**
- **l2vpn xconnect context**
- **load-balance flow**
- **monitor event-trace ac**
- **monitor event-trace atom**
- **monitor event-trace l2tp**
- **monitor peer bfd**
- **mtu**
- **preferred-path**
- **remote circuit id**
- **rd (VPLS)**
- **route-target (VPLS)**
- **sequencing**
- **status**
- **status admin-down disconnect**
- **status control-plane route-watch**
- **status decoupled**
- **status peer topology dual-homed**
- **status protocol notification static**

- **status redundancy**
- **switching tlv**
- **tlv**
- **tlv template**
- **vccv**
- **vccv bfd status signaling**
- **vccv bfd template**
- **vpls-id**
- **vpn id (MPLS)**

The table below lists the legacy commands that will be replaced in future releases. From Cisco IOS XE Release 3.7S and Cisco IOS Release 15.3(1)S both new and legacy commands will coexist until the legacy commands are deprecated in future releases.

Table 95: Replacement Commands Introduced in Cisco IOS XE Release 3.7S and Cisco IOS Release 15.3(1)S

Legacy Command	Replacement Command Introduced in Cisco IOS XE Release 3.7S and Cisco IOS Release 15.3(1)S
backup delay	**redundancy delay (under l2vpn xconnect context)**
bridge-domain (service instance)	**member (bridge-domain)**
clear mpls l2transport fsm state transition	**clear l2vpn atom fsm state transition**
clear mpls l2transport fsm event	**clear l2vpn atom fsm event**
clear xconnect	**clear l2vpn service**
connect (L2VPN local switching)	**l2vpn xconnect context**
debug acircuit	**debug l2vpn acircuit**
debug mpls l2transport checkpoint	**debug l2vpn atom checkpoint**
debug mpls l2transport event-trace	**debug l2vpn atom event-trace**
debug mpls l2transport fast-failure-detect	**debug l2vpn atom fast-failure-detect**
debug mpls l2transport signaling	**debug l2vpn atom signaling**
debug mpls l2transport static-oam	**debug l2vpn atom static-oam**
debug mpls l2transport vc subscriber	**debug l2vpn atom vc**
debug mpls l2transport vc	**debug l2vpn atom vc**
debug mpls l2transport vc vccv bfd event	**debug l2vpn atom vc vccv**
debug vfi	**debug l2vpn vfi**

Legacy Command	Replacement Command Introduced in Cisco IOS XE Release 3.7S and Cisco IOS Release 15.3(1)S
debug vfi checkpoint	**debug l2vpn vfi checkpoint**
debug xconnect	**debug l2vpn xconnect**
debug xconnect rib	**debug l2vpn xconnect rib**
description (L2VFI)	**description (L2VPN)**
l2 pseudowire routing	**pseudowire routing**
l2 router-id	**router-id**
l2 vfi	**l2vpn vfi context**
l2 subscriber	**l2vpn subscriber**
l2 vfi autodiscovery	**autodiscovery**
l2 vfi point-to-point	**l2vpn xconnect context**
local interface	**pseudowire type**
monitor event-trace st-pw-oam	**monitor event-trace pwoam**
mpls label	**label (pseudowire)**
mpls control-word	**control-word (encapsulation mpls under l2vpn connect context)**
neighbor (l2 vfi)	**member (l2vpn vfi)**
protocol	**signaling protocol**
pseudowire-static-oam class	**l2vpn pseudowire static-oam class**
pseudowire tlv template	**l2vpn pseudowire tlv template**
pw-class keyword in the **xconnect** command	**source template type pseudowire**
remote link failure notification	**l2vpn remote link failure notification**
show mpls l2transport binding	**show l2vpn atom binding**
show mpls l2transport checkpoint	**show l2vpn atom checkpoint**
show mpls l2transport hw-capability	**show l2vpn atom hw-capability**
show mpls l2transport static-oam	**show l2vpn atom static-oam**
show mpls l2transport summary	**show l2vpn atom summary**
show mpls l2transport pwid	**show l2vpn atom pwid**

Legacy Command	Replacement Command Introduced in Cisco IOS XE Release 3.7S and Cisco IOS Release 15.3(1)S
show mpls l2transport vc	**show l2vpn atom vc**
show xconnect pwmib	**show l2vpn pwmib**
show xconnect rib	**show l2vpn rib**
show xconnect	**show l2vpn service**
show vfi	**show l2vpn vfi**
xconnect	**l2vpn xconnect context** and **member**
xconnect logging pseudowire status global	**logging pseudowire status**
xconnect logging redundancy global	**logging redundancy**
xconnect *peer-ip vc-id*	**neighbor peer-ip vc-id (xconnect context)**

MPLS L2VPN Protocol-Based CLI: Examples

The examples in this section provide the new configurations that are introduced by the MPLS L2VPN Protocol-Based CLIs feature that replace the existing (legacy) MPLS L2VPN CLIs.

MPLS L2VPN VPWS Configuration Using Replacement (or New) Commands

The following example shows the configuration for Virtual Private Wired Service (VPWS)—Ethernet over Multiprotocol Label Switching (EoMPLS). In this example, L2VPN members point to peer ID or virtual circuit (VC) ID. This configuration is used in most cases except when features like quality of service (QoS), need to be applied at the pseudowire level.

```
l2vpn xconnect context foo
  member GigabitEthernet2/1/1 service-instance 300
  member 10.0.0.1 888 encapsulation mpls
!
interface GigabitEthernet2/1/1
  service instance 300
   encapsulation dot1q 30
   rewrite ingress tag pop 1 symmetric
  !
  service instance 400
   encapsulation dot1q 40
   rewrite ingress tag pop 1 symmetric

l2vpn xconnect context faa
  member GigabitEthernet2/1/1 service-instance 400
  member 10.0.0.1 999 encapsulation mpls
!
```

MPLS L2VPN Pseudowire Configuration Using Replacement (or New) Commands

In the following example, L2VPN members point to a pseudowire interface. The pseudowire interface is manually configured and includes peer ID and VC ID. This configuration is used in most cases except when features like quality of service (QoS), need to be applied at the pseudowire level.

```
l2vpn xconnect context foo
  member GigabitEthernet2/1/1 service-instance 300
```

```
  member Pseudowire888
!
interface Pseudowire 888
 encapsulation mpls
 neighbor 10.0.0.1 888
!
interface Pseudowire 999
 encapsulation mpls
 neighbor 10.0.0.1 999
!
interface GigabitEthernet2/1/1
  service instance 300
   encapsulation dot1q 30
   rewrite ingress tag pop 1 symmetric
  !
  service instance 400
   encapsulation dot1q 40
   rewrite ingress tag pop 1 symmetric

l2vpn xconnect context faa
  member GigabitEthernet2/1/1 service-instance 400
  member Pseudowire 999
!
```

MPLS L2VPN Pseudowire Redundancy Configuration Using Replacement (or New) Commands

The following example shows the configuration for pseudowire redundancy. The new configuration shows concise pseudowire redundancy with no submodes or separate groups. This configuration allows the addition of redundant members to a service without service disruption. This configuration also allows modifying or deleting redundant service configurations without service disruption.

```
l2vpn xconnect context sample-pw-redundancy
  member  service-instance 200
  member 10.1.1.1 180 encap mpls group Denver
  member 2.2.2.2 180180 encap mpls group Denver priority 1
  member 3.3.3.3 180181 encap mpls group Denver priority 2
  redundancy delay 1 20 group Denver
!
interface GigabitEthernet2/1/1
  service instance 200
   encapsulation dot1q 100
   rewrite ingress tag pop 1 symmetric
```

MPLS L2VPN Static Pseudowire Configuration Using Replacement (or New) Commands

Note The following configuration is shown for the Provider Edge (PE) 1 router in a network scheme where Customer Edge (CE) 1 and PE 1 and PE 2 and CE 2 traverse through a Provider core (P) router (CE 1—PE 1—P—PE 2—CE 2).

```
interface g2/1/1
service instance 300 ethernet
encapsulation dot1q 300
no shutdown
!
interface pseudowire 100
neighbor 10.4.4.4 121
encapsulation mpls
label 200 300
signaling protocol none
no shutdown
```

```
!
l2vpn xconnect context foo
  member GigabitEthernet2/1/1 service-instance 300
  member pseudowire 100
```

MPLS L2VPN Static Pseudowire Template Configuration Using Replacement (or New) Commands

Note The following configuration is shown for the Provider Edge (PE) 1 router in a network scheme where Customer Edge (CE) 1 and PE 1 and PE 2 and CE 2 traverse through a Provider core (P) router (CE 1—PE 1—P—PE 2—CE 2).

```
template type pseudowire test
encapsulation mpls
signaling protocol none
!
interface g2/1/1
service instance 300 ethernet
encapsulation dot1q 300
no shutdown
!
interface pseudowire 100
neighbor 10.4.4.4 121
source template type pseudowire test
label 200 300
no shutdown
!
l2vpn xconnect context foo
  member GigabitEthernet2/1/1 service-instance 300
  member pseudowire 100
```

MPLS L2VPN Dynamic Pseudowire Template Configuration Using Replacement (or New) Commands

Note The following configuration is shown for the Provider Edge (PE) 1 router in a network scheme where Customer Edge (CE) 1 and PE 1 and PE 2 and CE 2 traverse through a Provider core (P) router (CE 1—PE 1—P—PE 2—CE 2).

```
template type pseudowire test
encapsulation mpls
signaling protocol ldp
!
!
interface g2/1/1
service instance 300 ethernet
encapsulation dot1q 300
no shutdown
!
interface pseudowire 100
neighbor 10.4.4.4 121
source template type pseudowire test
no shutdown
!
l2vpn xconnect context foo
  member GigabitEthernet2/1/1 service-instance 300
  member pseudowire 100
```

MPLS L2VPN Multi-segment Static-Dynamic Pseudowire Template Configuration Using Replacement (or New) Commands

The following PE router configuration is for a multi-segment static-dynamic pseudowire:

```
l2vpn pseudowire tlv template TLV
 tlv mtu 1 4 dec 1500
!
interface pseudowire401
  source template type pseudowire staticTempl
encapsulation mpls
neighbor 10.4.4.4 101
signaling protocol none
label 4401 4301
pseudowire type 4
tlv template TLV
tlv 1 4 dec 1500
tlv vccv-flags C 4 hexstr 0110
!
interface pseudowire501
  source template type pseudowire dynTempl
encapsulation mpls
neighbor 10.2.2.2 101
signaling protocol ldp
```

Displaying MPLS L2VPN Pseudowire Template Configuration Using Replacement (or New) Commands

The following example displays output from the **show interface pseudowire** command:

```
PE1#show interface pseudowire 100
pseudowire100 is up
    Description: Pseudowire Interface
    MTU 1500 bytes, BW 10000000 Kbit
    Encapsulation mpls
    Peer IP 10.4.4.4, VC ID 121
    RX
      21 packets 2623 bytes 0 drops
    TX
      20 packets 2746 bytes 0 drops
```

The following example displays output from the **show template** command:

```
PE1#show template

Template        class/type    Component(s)
ABC             owner        interface pseudowire
 BOUND: pw1
```

Sourcing a Template Under an Interface Pseudowire Using Replacement (or New) Commands

The following example configures the interface pseudowire to inherit all attributes defined from a template on the PE 2 router.

```
PE2(config-subif)#interface pseudowire 100
PE2(config-if)#source template type pseudowire test
PE2(config-if)#neighbor 10.4.4.4 121
PE2(config-if)#no shutdown
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
MPLS commands	Multiprotocol Label Switching Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

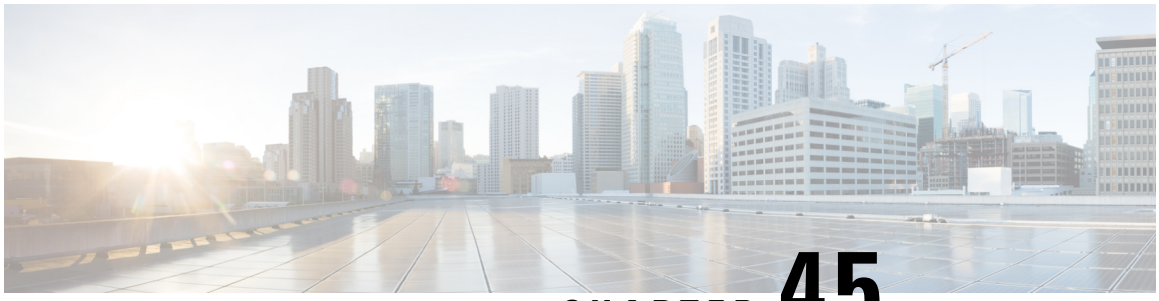
Feature Information for L2VPN Protocol-Based CLIs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 96: Feature Information for L2VPN Protocol-Based CLIs

Feature Name	Releases	Feature Information
L2VPN Protocol-Based CLIs	Cisco IOS XE Release 3.7S	The L2VPN Protocol-Based CLIs feature provides a set of processes and an improved infrastructure for developing and delivering Cisco IOS software on various Cisco platforms. This feature introduces new commands and modifies or replaces existing commands to achieve a consistent functionality across Cisco platforms and provide cross-Operating System (OS) support. In Cisco IOS XE Release 3.7S, this feature was introduced on the Cisco ASR 903 Router.



CHAPTER 45

Any Transport over MPLS

This module describes how to configure Any Transport over MPLS (AToM) transports data link layer (Layer 2) packets over a Multiprotocol Label Switching (MPLS) backbone. AToM enables service providers to connect customer sites with existing Layer 2 networks by using a single, integrated, packet-based network infrastructure--a Cisco MPLS network. Instead of using separate networks with network management environments, service providers can deliver Layer 2 connections over an MPLS backbone. AToM provides a common framework to encapsulate and transport supported Layer 2 traffic types over an MPLS network core.

AToM supports the following like-to-like transport types:

- ATM Adaptation Layer Type-5 (AAL5) over MPLS
- ATM Cell Relay over MPLS
- Ethernet over MPLS (port modes)

Prerequisites for Any Transport over MPLS

- IP routing must be configured in the core so that the provider edge (PE) routers can reach each other via IP.
- MPLS must be configured in the core so that a label-switched path (LSP) exists between the PE routers.
- A loopback interface must be configured for originating and terminating Layer 2 traffic. Ensure that the PE routers can access the other router's loopback interface. Note that the loopback interface is not needed in all cases. For example, tunnel selection does not need a loopback interface when AToM is directly mapped to a traffic engineering (TE) tunnel.

Restrictions for Any Transport over MPLS

General Restrictions

The following general restrictions pertain to all transport types under AToM:

- Address format: Configure the Label Distribution Protocol (LDP) router ID on all PE routers to be a loopback address with a /32 mask. Otherwise, some configurations might not function properly.

Ethernet over MPLS (EoMPLS) Restrictions

The following restrictions pertain to the Ethernet over MPLS feature:

- Ethernet over MPLS supports VLAN packets that conform to the IEEE 802.1Q standard. The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames. The Inter-Switch Link (ISL) protocol is not supported between the PE and CE routers.
- The AToM control word is supported. However, if the peer PE does not support a control word, the control word is disabled. This negotiation is done by LDP label binding.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.

General Restrictions

- In a member configuration, the **l2vpn xconnect context** command does not prompt any error or warning, if you specify without a service instance.
- The **show mpls l2transport vc** <*vcid*> **detail** command output displays few LDP-related information, even in case of static pseudowire.
- Address format--Configure the Label Distribution Protocol (LDP) router ID on all PE routers to be a loopback address with a /32 mask. Otherwise, some configurations might not function properly.
- For PTPoIP configuration with explicit Null MPLS encapsulation, when a Transparent Clock (TC) is placed between a PTP primary and a PTP subordinate, the TC does not update the correction field.
- If an AToM tunnel spans different service providers that exchange MPLS labels using IPv4 Border Gateway Protocol (BGP) (RFC 3107), you add a label to the stack. The maximum MPLS label stack is .
- Hot standby pseudowire (HSPW) convergence without pseudowire grouping increments linearly. For example, for a thousand virtual circuits, it requires about 54 seconds of convergence time. This is applicable only for the Cisco RSP3 Module.

  Clear interface is not the recommended way to measure the convergence numbers.
- With two ECMP paths, load sharing on L2VPN traffic occurs based on odd or even MPLS VC labels. If L2VPN circuits have either odd **or** even MPLS VC labels, load sharing is not performed. But if L2VPN circuits have a combination of both odd **and** even MPLS VC labels, then the odd MPLS VC labels circuits select one link whereas the even MPLS VC labels circuits select another link.

ATM AAL5 over MPLS Restrictions

- AAL5 over MPLS is supported only in SDU mode.

ATM Cell Relay over MPLS Restrictions

- If you have TE tunnels running between the PE routers, you must enable LDP on the tunnel interfaces.
- The F4 end-to-end OAM cells are transparently transported along with the ATM cells. When a permanent virtual path (PVP) or permanent virtual circuit (PVC) is down on one PE router, the label associated with that PVP or PVC is withdrawn. Subsequently, the peer PE router detects the label withdrawal and sends an F4 AIS/RDI signal to its corresponding CE router. The PVP or PVC on the peer PE router remains in the up state.
- VC class configuration mode is not supported in port mode.
- The AToM control word is supported. However, if a peer PE does not support the control word, it is disabled.

For configuring ATM cell relay over MPLS in VP mode, the following restrictions apply:

- If a VPI is configured for VP cell relay, you cannot configure a PVC using the same VPI.
- VP trunking (mapping multiple VPs to one emulated VC label) is not supported. Each VP is mapped to one emulated VC.
- VP mode and VC mode drop idle cells.

Ethernet over MPLS (EoMPLS) Restrictions

- The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet.
- The subinterface on the adjoining CE router must be on the same VLAN as the PE router.
- Ethernet over MPLS supports VLAN packets that conform to the IEEE 802.1Q standard. The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames. The Inter-Switch Link (ISL) protocol is not supported between the PE and CE routers.
- The AToM control word is supported. However, if the peer PE does not support a control word, the control word is disabled.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.

Per-Subinterface MTU for Ethernet over MPLS Restrictions

- The following features do not support MTU values in xconnect subinterface configuration mode:
  - Layer 2 Tunnel Protocol Version 3 (L2TPv3)
  - Virtual Private LAN services (VPLS)
  - L2VPN Pseudowire Switching

- The MTU value can be configured in xconnect subinterface configuration mode only on the following interfaces and subinterfaces:
  - Fast Ethernet
  - Gigabit Ethernet
- The router uses an MTU validation process for remote VCs established through LDP, which compares the MTU value configured in xconnect subinterface configuration mode to the MTU value of the remote customer interface. If an MTU value has not been configured in xconnect subinterface configuration mode, then the validation process compares the MTU value of the local customer interface to the MTU value of the remote xconnect, either explicitly configured or inherited from the underlying interface or subinterface.
- When you configure the MTU value in xconnect subinterface configuration mode, the specified MTU value is not enforced by the dataplane. The dataplane enforces the MTU values of the interface (port mode) or subinterface (VLAN mode).
- Ensure that the interface MTU is larger than the MTU value configured in xconnect subinterface configuration mode. If the MTU value of the customer-facing subinterface is larger than the MTU value of the core-facing interface, traffic may not be able to travel across the pseudowire.

Frame Relay over MPLS Restrictions

Frame Relay traffic shaping is not supported with AToM switched VCs.

HDLC over MPLS Restrictions

- Asynchronous interfaces are not supported.
- You must configure HDLC over MPLS on router interfaces only. You cannot configure HDLC over MPLS on subinterfaces.

PPP over MPLS Restrictions

- Zero hops on one router is not supported. However, you can have back-to-back PE routers.
- Asynchronous interfaces are not supported. The connections between the CE and PE routers on both ends of the backbone must have similar link layer characteristics. The connections between the CE and PE routers must both be synchronous.
- Multilink PPP (MLP) is not supported.
- You must configure PPP on router interfaces only. You cannot configure PPP on subinterfaces.

Tunnel Selection Restrictions

- The selected path should be an LSP destined to the peer PE router.
- The selected tunnel must be an MPLS TE tunnel.

- If you specify an IP address, that address must be the IP address of the loopback interface on the remote PE router. The address must have a /32 mask. There must be an LSP destined to that selected address. The LSP need not be a TE tunnel.

Experimental Bits with AToM Restrictions

- You must statically set the experimental (EXP) bits in both the VC label and the LSP tunnel label, because the LSP tunnel label might be removed at the penultimate router.
- For EXP bits and ATM AAL5 over MPLS and for EXP bits and Frame Relay over MPLS, if you do not assign values to the experimental bits, the priority bits in the header's "tag control information" field are set to zero.
- For EXP bits and ATM Cell Relay over MPLS in VC mode, if you do not assign values to the experimental bits, the priority bits in the header's "tag control information" field are set to zero.
- For EXP bits and HDLC over MPLS and PPP over MPLS, if you do not assign values to the experimental bits, zeros are written into the experimental bit fields.

Remote Ethernet Port Shutdown Restrictions

This feature is not symmetrical if the remote PE router is running an older version image or is on another platform that does not support the EoMPLS remote Ethernet port shutdown feature and the local PE is running an image which supports this feature.

Remote Ethernet Port Shutdown is supported only on EFP with encapsulation default.

Information About Any Transport over MPLS

To configure AToM, you must understand the following concepts:

How AToM Transports Layer 2 Packets

AToM encapsulates Layer 2 frames at the ingress PE and sends them to a corresponding PE at the other end of a pseudowire, which is a connection between the two PE routers. The egress PE removes the encapsulation and sends out the Layer 2 frame.

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a pseudowire, between the routers. You specify the following information on each PE router:

- The type of Layer 2 data that will be transported across the pseudowire, such as Ethernet, Frame Relay, or ATM
- The IP address of the loopback interface of the peer PE router, which enables the PE routers to communicate
- A unique combination of peer PE IP address and VC ID that identifies the pseudowire

The following example shows the basic configuration steps on a PE router that enable the transport of Layer 2 packets. Each transport type has slightly different steps.

Step 1 defines the interface or subinterface on the PE router:

```
Router# interface
 interface-type interface-number
```

Step specifies the encapsulation type for the interface, such as dot1q:

```
Router(config-if-srv)# encapsulation
encapsulation-type
```

Step 4 does the following:

- Makes a connection to the peer PE router by specifying the LDP router ID of the peer PE router.
- Specifies a 32-bit unique identifier, called the VC ID, which is shared between the two PE routers.

The combination of the peer router ID and the VC ID must be unique on the router. Two circuits cannot use the same combination of peer router ID and VC ID.

- Specifies the tunneling method used to encapsulate data in the pseudowire. AToM uses MPLS as the tunneling method.

```
Router(config-if-srv)# xconnect
peer-router-id vcid
encapsulation mpls
```

As an alternative, you can set up a pseudowire class to specify the tunneling method and other characteristics. For more information, see the Configuring the Pseudowire Class, on page 747.

How AToM Transports Layer 2 Packets Using Commands Associated with L2VPN Protocol-Based Feature

AToM encapsulates Layer 2 frames at the ingress PE and sends them to a corresponding PE at the other end of a pseudowire, which is a connection between the two PE routers. The egress PE removes the encapsulation and sends out the Layer 2 frame.

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a pseudowire, between the routers. You specify the following information on each PE router:

- The type of Layer 2 data that will be transported across the pseudowire, such as Ethernet, Frame Relay, or ATM
- The IP address of the loopback interface of the peer PE router, which enables the PE routers to communicate
- A unique combination of peer PE IP address and VC ID that identifies the pseudowire

The following example shows the basic configuration steps on a PE router that enable the transport of Layer 2 packets. Each transport type has slightly different steps.

Step 1 defines the interface or subinterface on the PE router:

```
Router# interface
interface-type interface-number
```

Step 3 specifies the encapsulation type for the interface, such as dot1q:

```
Router(config-if)# encapsulation
encapsulation-type
```

Step 3 does the following:

- Makes a connection to the peer PE router by specifying the LDP router ID of the peer PE router.
- Specifies a 32-bit unique identifier, called the VC ID, which is shared between the two PE routers.

The combination of the peer router ID and the VC ID must be unique on the router. Two circuits cannot use the same combination of peer router ID and VC ID.

- Specifies the tunneling method used to encapsulate data in the pseudowire. AToM uses MPLS as the tunneling method.

```
Router(config)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.0.0.1 123
Router(config-if)# exit
!
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100
```

```
Router(config-xconnect)# exit
```

As an alternative, you can set up a pseudowire class to specify the tunneling method and other characteristics. For more information, see the Configuring the Pseudowire Class, on page 747.

Benefits of AToM

The following list explains some of the benefits of enabling Layer 2 packets to be sent in the MPLS network:

- The AToM product set accommodates many types of Layer 2 packets, including Ethernet and Frame Relay, across multiple Cisco router platforms. This enables the service provider to transport all types of traffic over the backbone and accommodate all types of customers.
- AToM adheres to the standards developed for transporting Layer 2 packets over MPLS. This benefits the service provider that wants to incorporate industry-standard methodologies in the network. Other Layer 2 solutions are proprietary, which can limit the service provider's ability to expand the network and can force the service provider to use only one vendor's equipment.
- Upgrading to AToM is transparent to the customer. Because the service provider network is separate from the customer network, the service provider can upgrade to AToM without disruption of service to the customer. The customers assume that they are using a traditional Layer 2 backbone.

MPLS Traffic Engineering Fast Reroute

AToM can use MPLS traffic engineering (TE) tunnels with fast reroute (FRR) support. AToM VCs can be rerouted around a failed link or node at the same time as MPLS and IP prefixes.

Enabling fast reroute on AToM does not require any special commands; you can use standard fast reroute commands. At the ingress PE, an AToM tunnel is protected by fast reroute when it is routed to an FRR-protected TE tunnel. Both link and node protection are supported for AToM VCs at the ingress PE.

Maximum Transmission Unit Guidelines for Estimating Packet Size

The following calculation helps you determine the size of the packets traveling through the core network. You set the maximum transmission unit (MTU) on the core-facing interfaces of the P and PE routers to accommodate packets of this size. The MTU should be greater than or equal to the total bytes of the items in the following equation:

```
Core MTU >= (Edge MTU + Transport header + AToM header + (MPLS label stack * MPLS label
size))
```

The following sections describe the variables used in the equation.

Edge MTU

The edge MTU is the MTU for the customer-facing interfaces.

Transport Header

The Transport header depends on the transport type. The table below lists the specific sizes of the headers.

Table 97: Header Size of Packets

Transport Type	Packet Size
AAL5	0-32 bytes
Ethernet VLAN	18 bytes
Ethernet Port	14 bytes
Frame Relay DLCI	2 bytes for Cisco encapsulation, 8 bytes for Internet Engineering Task Force (IETF) encapsulation
HDLC	4 bytes
PPP	4 bytes

AToM Header

The AToM header is 4 bytes (control word). The control word is optional for Ethernet, PPP, HDLC, and cell relay transport types. The control word is required for Frame Relay and ATM AAL5 transport types.

MPLS Label Stack

The MPLS label stack size depends on the configuration of the core MPLS network:

- AToM uses one MPLS label to identify the AToM VCs (VC label). Therefore, the minimum MPLS label stack is one for directly connected AToM PEs, which are PE routers that do not have a P router between them.
- If LDP is used in the MPLS network, the label stack size is two (the LDP label and the VC label).
- If a TE tunnel instead of LDP is used between PE routers in the MPLS network, the label stack size is two (the TE label and the VC label).
- If a TE tunnel and LDP are used in the MPLS network (for example, a TE tunnel between P routers or between P and PE routers, with LDP on the tunnel), the label stack is three (TE label, LDP label, VC label).
- If you use MPLS fast reroute in the MPLS network, you add a label to the stack. The maximum MPLS label stack in this case is four (FRR label, TE label, LDP label, VC label).
- If AToM is used by the customer carrier in an MPLS VPN Carrier Supporting Carrier environment, you add a label to the stack. The maximum MPLS label stack in the provider carrier network is .
- If an AToM tunnel spans different service providers that exchange MPLS labels using IPv4 Border Gateway Protocol (BGP) (RFC 3107), you add a label to the stack. The maximum MPLS label stack is
- TE-FRR with BGP labels for layer 2 and layer 3 VPNs must terminate on the BGP gateway because of the four-label limitation.

Other circumstances can increase the MPLS label stack size. Therefore, analyze the complete data path between the AToM tunnel endpoints and determine the maximum MPLS label stack size for your network. Then multiply the label stack size by the size of the MPLS label.

Estimating Packet Size Example

The estimated packet size in the following example is 1526 bytes, based on the following assumptions:

- The edge MTU is 1500 bytes.
- The transport type is Ethernet VLAN, which designates 18 bytes for the transport header.
- The AToM header is 0, because the control word is not used.
- The MPLS label stack is 2, because LDP is used. The MPLS label is 4 bytes.

```
Edge MTU + Transport header + AToM header + (MPLS label stack * MPLS label) = Core MTU
1500     + 18                     + 0      + (2                * 4          ) = 1526
```

You must configure the P and PE routers in the core to accept packets of 1526 bytes.

Per-Subinterface MTU for Ethernet over MPLS

MTU values can be specified in xconnect subinterface configuration mode. When you use xconnect subinterface configuration mode to set the MTU value, you establish a pseudowire connection for situations where the interfaces have different MTU values that cannot be changed.

If you specify an MTU value in xconnect subinterface configuration mode that is outside the range of supported MTU values (64 bytes to the maximum number of bytes supported by the interface), the command might be rejected. If you specify an MTU value that is out of range in xconnect subinterface configuration mode, the router enters the command in subinterface configuration mode.

For example, if you specify an MTU of 1501 in xconnect subinterface configuration mode, and that value is out of range, the router enters the command in subinterface configuration mode, where it is accepted:

```
Router# configure terminal
Router(config)# interface gigabitethernet0/0/2.1
Router(config-subif)# xconnect 10.10.10.1 100 encapsulation mpls
Router(config-subif-xconn)# mtu ?
<64 - 1500> MTU size in bytes
Router(config-subif-xconn)# mtu 1501 <<================
Router(config-subif)# mtu ?
<64 - 17940> MTU size in bytes
```

If the MTU value is not accepted in either xconnect subinterface configuration mode or subinterface configuration mode, then the command is rejected.

Per-Subinterface MTU for Ethernet over MPLS using the commands associated with the L2VPN Protocol-Based CLIs feature

MTU values can be specified in xconnect configuration mode. When you use xconnect configuration mode to set the MTU value, you establish a pseudowire connection for situations where the interfaces have different MTU values that cannot be changed.

If you specify an MTU value in xconnect configuration mode that is outside the range of supported MTU values (64 bytes to the maximum number of bytes supported by the interface), the command might be rejected. If you specify an MTU value that is out of range in xconnect configuration mode, the router enters the command in subinterface configuration mode.

For example, if you specify an MTU of 1501 in xconnect configuration mode, and that value is out of range, the router enters the command in subinterface configuration mode, where it is accepted:

```
Router# configure terminal
Router(config)# interface gigabitethernet0/0/2.1
Router(config)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.10.10.1 100
Router(config-if)# mtu ?
<64 - 1500> MTU size in bytes
Router(config-if)# mtu 1501 <<================
Router(config-if)# mtu ?
<64 - 17940> MTU size in bytes
Router(config-if)# exit
!
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100 Router
Router(config-xconnect)# member gigabitethernet0/0/2.1
Router(config-xconnect)# exit
```

If the MTU value is not accepted in either xconnect configuration mode or subinterface configuration mode, then the command is rejected.

Frame Relay over MPLS and DTE DCE and NNI Connections

You can configure an interface as a DTE device or a DCE switch, or as a switch connected to a switch with network-to-network interface (NNI) connections. Use the following command in interface configuration mode:

frame-relay intf-type [**dce** | **dte** | **nni**]

The keywords are explained in the table below.

Table 98: frame-relay intf-type Command Keywords

Keyword	Description
dce	Enables the router or access server to function as a switch connected to a router.
dte	Enables the router or access server to function as a DTE device. DTE is the default.
nni	Enables the router or access server to function as a switch connected to a switch.

Local Management Interface and Frame Relay over MPLS

Local Management Interface (LMI) is a protocol that communicates status information about PVCs. When a PVC is added, deleted, or changed, the LMI notifies the endpoint of the status change. LMI also provides a polling mechanism that verifies that a link is up.

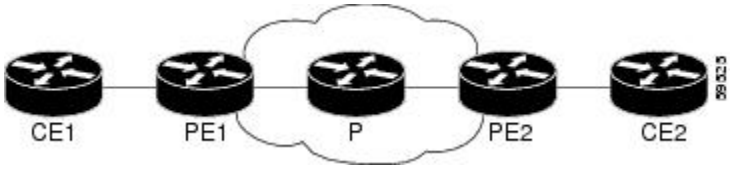
How LMI Works

To determine the PVC status, LMI checks that a PVC is available from the reporting device to the Frame Relay end-user device. If a PVC is available, LMI reports that the status is "Active," which means that all interfaces, line protocols, and core segments are operational between the reporting device and the Frame Relay end-user device. If any of those components is not available, the LMI reports a status of "Inactive."

Note Only the DCE and NNI interface types can report the LMI status.

The figure below is a sample topology that helps illustrate how LMI works.

Figure 50: Sample Topology



In the figure above, note the following:

- CE1 and PE1 and PE2 and CE2 are Frame Relay LMI peers.
- CE1 and CE2 can be Frame Relay switches or end-user devices.
- Each Frame Relay PVC comprises multiple segments.
- The DLCI value is local to each segment and is changed as traffic is switched from segment to segment. Two Frame Relay PVC segments exist in the figure; one is between PE1 and CE1 and the other is between PE2 and CE2.

The LMI protocol behavior depends on whether you have DLCI-to-DLCI or port-to-port connections.

DLCI-to-DLCI Connections

If you have DLCI-to-DLCI connections, LMI runs locally on the Frame Relay ports between the PE and CE devices:

- CE1 sends an active status to PE1 if the PVC for CE1 is available. If CE1 is a switch, LMI checks that the PVC is available from CE1 to the user device attached to CE1.
- PE1 sends an active status to CE1 if the following conditions are met:
  - A PVC for PE1 is available.
  - PE1 received an MPLS label from the remote PE router.
  - An MPLS tunnel label exists between PE1 and the remote PE.

For DTE or DCE configurations, the following LMI behavior exists: The Frame Relay device accessing the network (DTE) does not report the PVC status. Only the network device (DCE) or NNI can report the status. Therefore, if a problem exists on the DTE side, the DCE is not aware of the problem.

Port-to-Port Connections

If you have port-to-port connections, the PE routers do not participate in the LMI status-checking procedures. LMI operates only between the CE routers. The CE routers must be configured as DCE-DTE or NNI-NNI.

For information about LMI, including configuration instructions, see the “Configuring the LMI” section of the Configuring Frame Relay document.

QoS Features Supported with AToM

The tables below list the QoS features supported by AToM.

Table 99: QoS Features Supported with Ethernet over MPLS

QoS Feature	Ethernet over MPLS
Service policy	Can be applied to: • Interface (input and output)
Classification	Supports the following commands: • **match cos** (on interfaces) • **match mpls experimental** (on interfaces) • **match qos-group** (on interfaces) (output policy)
Marking	Supports the following commands: • **set cos** (output policy) • **set discard-class** (input policy) • **set mpls experimental** (input policy) (on interfaces) • **set qos-group** (input policy)

QoS Feature	Ethernet over MPLS
Policing	Supports the following: • Color-aware policing • Multiple-action policing • Single-rate policing • Two-rate policing
Queueing and shaping	Supports the following: • Byte-based WRED • Low Latency Queueing (LLQ) • Weighted Random Early Detection (WRED)

Table 100: QoS Features Supported with Frame Relay over MPLS

QoS Feature	Frame Relay over MPLS
Service policy	Can be applied to: • Interface (input and output) • PVC (input and output)
Classification	Supports the following commands: • **match fr-de** (on interfaces and VCs) • **match fr-dlci** (on interfaces) • **match qos-group**
Marking	Supports the following commands: • **frame-relay congestion management** (output) • **set discard-class** • **set fr-de** (output policy) • **set fr-fecn-becn** (output) • **set mpls experimental** • **set qos-group** • **threshold ecn** (output)

QoS Feature	Frame Relay over MPLS
Policing	Supports the following: • Color-aware policing • Multiple-action policing • Single-rate policing • Two-rate policing
Queueing and shaping	Supports the following: • Byte-based WRED • Class-based weighted fair queueing (CBWFQ) • LLQ • **random-detect discard-class-based** command • Traffic shaping • WRED

Table 101: QoS Features Supported with ATM Cell Relay and AAL5 over MPLS

QoS Feature	ATM Cell Relay and AAL5 over MPLS
Service policy	Can be applied to: • Interface (input and output) • PVC (input and output) • Subinterface (input and output)
Classification	Supports the following commands: • **match mpls experimental** (on VCs) • **match qos-group** (output)
Marking	Supports the following commands: • **random-detect discard-class-based** (input) • **set clp** (output) (on interfaces, subinterfaces, and VCs) • **set discard-class** (input) • **set mpls experimental** (input) (on interfaces, subinterfaces, and VCs) • **set qos-group** (input)

QoS Feature	ATM Cell Relay and AAL5 over MPLS
Policing	Supports the following: • Color-aware policing • Multiple-action policing • Single-rate policing • Two-rate policing
Queueing and shaping	Supports the following: • Byte-based WRED • CBWFQ • Class-based shaping support on ATM PVCs • LLQ • **random-detect discard-class-based** command • WRED

OAM Cell Emulation for ATM AAL5 over MPLS

If a PE router does not support the transport of Operation, Administration, and Maintenance (OAM) cells across a label switched path (LSP), you can use OAM cell emulation to locally terminate or loop back the OAM cells. You configure OAM cell emulation on both PE routers, which emulates a VC by forming two unidirectional LSPs. You use Cisco software commands on both PE routers to enable OAM cell emulation.

After you enable OAM cell emulation on a router, you can configure and manage the ATM VC in the same manner as you would a terminated VC. A VC that has been configured with OAM cell emulation can send loopback cells at configured intervals toward the local CE router. The endpoint can be either of the following:

- End-to-end loopback, which sends OAM cells to the local CE router.
- Segment loopback, which responds to OAM cells to a device along the path between the PE and CE routers.

The OAM cells include the following cells:

- Alarm indication signal (AIS)
- Remote defect indication (RDI)

These cells identify and report defects along a VC. When a physical link or interface failure occurs, intermediate nodes insert OAM AIS cells into all the downstream devices affected by the failure. When a router receives an AIS cell, it marks the ATM VC down and sends an RDI cell to let the remote end know about the failure.

OAM Cell Emulation for ATM AAL5 over MPLS in VC Class Configuration Mode

You can configure OAM cell emulation as part of a VC class and then apply the VC class to an interface, a subinterface, or a VC. When you configure OAM cell emulation in VC class configuration mode and then

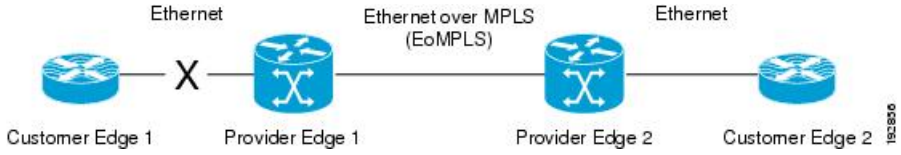
apply the VC class to an interface, the settings in the VC class apply to all the VCs on the interface, unless you specify a different OAM cell emulation value at a lower level, such as the subinterface or VC level. For example, you can create a VC class that specifies OAM cell emulation and sets the rate of AIS cells to every 30 seconds. You can apply the VC class to an interface. Then, for one PVC, you can enable OAM cell emulation and set the rate of AIS cells to every 15 seconds. All the PVCs on the interface use the cell rate of 30 seconds, except for the one PVC that was set to 15 seconds.

Any Transport over MPLS (AToM) Remote Ethernet Port Shutdown

This Cisco IOS XE feature allows a service provider edge (PE) router on the local end of an Ethernet over MPLS (EoMPLS) pseudowire to detect a remote link failure and cause the shutdown of the Ethernet port on the local customer edge (CE) router. Because the Ethernet port on the local CE router is shut down, the router does not lose data by continuously sending traffic to the failed remote link. This is beneficial if the link is configured as a static IP route.

The figure below illustrates a condition in an EoMPLS WAN, with a down Layer 2 tunnel link between a CE router (Customer Edge 1) and the PE router (Provider Edge 1). A CE router on the far side of the Layer 2 tunnel (Customer Edge 2), continues to forward traffic to Customer Edge 1 through the L2 tunnel.

Figure 51: Remote Link Outage in EoMPLS WAN



Previous to this feature, the Provider Edge 2 router could not detect a failed remote link. Traffic forwarded from Customer Edge 2 to Customer Edge 1 would be lost until routing or spanning tree protocols detected the down remote link. If the link was configured with static routing, the remote link outage would be even more difficult to detect.

With this feature, the Provider Edge 2 router detects the remote link failure and causes a shutdown of the local Customer Edge 2 Ethernet port. When the remote L2 tunnel link is restored, the local interface is automatically restored as well. The possibility of data loss is thus diminished.

With reference to the figure above, the Remote Ethernet Shutdown sequence is generally described as follows:

1. The remote link between Customer Edge 1 and Provider Edge 1 fails.
2. Provider Edge 2 detects the remote link failure and disables the transmit laser on the line card interface connected to Customer Edge 2.
3. An RX_LOS error alarm is received by Customer Edge 2 causing Customer Edge 2 to bring down the interface.
4. Provider Edge 2 maintains its interface with Customer Edge 2 in an up state.
5. When the remote link and EoMPLS connection is restored, the Provider Edge 2 router enables the transmit laser.
6. The Customer Edge 2 router brings up its downed interface.

This feature is enabled by default for Ethernet over MPLS (EoMPLS). You can also enable this feature by using the **remote link failure notification** command in xconnect configuration mode as shown in the following example:

```
pseudowire-class eompls
 encapsulation mpls
!
interface GigabitEthernet1/0/0
 xconnect 10.13.13.13 1 pw-class eompls
  remote link failure notification
!
```

This feature can be disabled using the **no remote link failure notification** command in xconnect configuration mode. Use the **show ip interface brief** privileged EXEC command to display the status of all remote L2 tunnel links. Use the **show interface** privileged EXEC command to show the status of the L2 tunnel on a specific interface.

Note The **no remote link failure notification** command will not give notification to clients for remote attachment circuit status down.

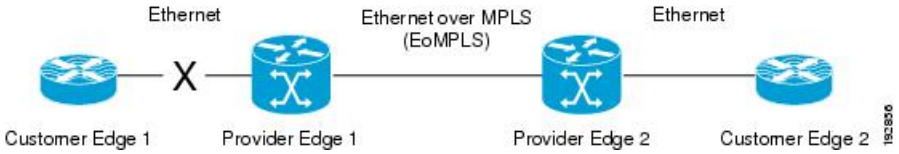
Note Remote Ethernet Port Shutdown is supported only on EFP with encapsulation default.

Any Transport over MPLS (AToM) Remote Ethernet Port Shutdown Using Commands Associated with L2VPN Protocol-Based Feature

This Cisco IOS XE feature allows a service provider edge (PE) router on the local end of an Ethernet over MPLS (EoMPLS) pseudowire to detect a remote link failure and cause the shutdown of the Ethernet port on the local customer edge (CE) router. Because the Ethernet port on the local CE router is shut down, the router does not lose data by continuously sending traffic to the failed remote link. This is beneficial if the link is configured as a static IP route.

The figure below illustrates a condition in an EoMPLS WAN, with a down Layer 2 tunnel link between a CE router (Customer Edge 1) and the PE router (Provider Edge 1). A CE router on the far side of the Layer 2 tunnel (Customer Edge 2), continues to forward traffic to Customer Edge 1 through the L2 tunnel.

Figure 52: Remote Link Outage in EoMPLS WAN



Previous to this feature, the Provider Edge 2 router could not detect a failed remote link. Traffic forwarded from Customer Edge 2 to Customer Edge 1 would be lost until routing or spanning tree protocols detected the down remote link. If the link was configured with static routing, the remote link outage would be even more difficult to detect.

With this feature, the Provider Edge 2 router detects the remote link failure and causes a shutdown of the local Customer Edge 2 Ethernet port. When the remote L2 tunnel link is restored, the local interface is automatically restored as well. The possibility of data loss is thus diminished.

With reference to the figure above, the Remote Ethernet Shutdown sequence is generally described as follows:

1. The remote link between Customer Edge 1 and Provider Edge 1 fails.
2. Provider Edge 2 detects the remote link failure and disables the transmit laser on the line card interface connected to Customer Edge 2.
3. An RX_LOS error alarm is received by Customer Edge 2 causing Customer Edge 2 to bring down the interface.
4. Provider Edge 2 maintains its interface with Customer Edge 2 in an up state.
5. When the remote link and EoMPLS connection is restored, the Provider Edge 2 router enables the transmit laser.
6. The Customer Edge 2 router brings up its downed interface.

This feature is enabled by default for Ethernet over MPLS (EoMPLS). You can also enable this feature by using the **remote link failure notification** command in xconnect configuration mode as shown in the following example:

This feature can be disabled using the **no remote link failure notification** command in xconnect configuration mode. Use the **show ip interface brief** privileged EXEC command to display the status of all remote L2 tunnel links. Use the **show interface** privileged EXEC command to show the status of the L2 tunnel on a specific interface.

Note The **no remote link failure notification** command will not give notification to clients for remote attachment circuit status down.

AToM Load Balancing with Single PW

The AToM Load Balancing with Single PW feature enables load balancing for packets within the same pseudowire by further classifying packets within the same pseudowire into different flows based on certain fields in the packet received on an attachment circuit. For example, for Ethernet this load balancing is based on the source MAC address in the incoming packets.

Flow-Aware Transport (FAT) Load Balancing

The Flow-Aware Transport of MPLS Pseudowires feature enables load balancing of packets within the same pseudowire by further classifying the packets into different flows by adding a flow label at the bottom of the MPLS label stack.

Information About EoMPLS over IPv6 GRE Tunnel

Ethernet over MPLS (EoMPLS) is a tunneling mechanism that allows you to tunnel Layer 2 traffic through a Layer 3 MPLS network. EoMPLS is also known as Layer 2 tunneling.

The EoMPLS over IPv6 GRE Tunnel feature supports tunneling of EoMPLS traffic via an IPv6 network by using GRE tunnels. Effective from Cisco IOS XE Release 3.15s, EoMPLS is supported over IPv6 GRE tunnel.

Additional Information on EoMPLS over IPv6 GRE Tunnel

For more information on EoMPLS over IPv6 GRE Tunnel feature, see GRE IPv6 Tunnels chapter of the *Interface and Hardware Component Configuration Guide, Cisco IOS XE Release 3S (ASR 1000)*.

How to Configure Any Transport over MPLS

This section explains how to perform a basic AToM configuration and includes the following procedures:

Configuring the Pseudowire Class

Note In simple configurations, this task is optional. You need not specify a pseudowire class if you specify the tunneling method as part of the **xconnect** command.

- You must specify the **encapsulation mpls** command as part of the pseudowire class or as part of the **xconnect** command for the AToM VCs to work properly. If you omit the **encapsulation mpls** command as part of the **xconnect** command, you receive the following error:

```
% Incomplete command.
```

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **pseudowire-class** *name*

Example:

```
Router(config)# pseudowire-class atom
```

Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.

Step 4 **encapsulation mpls**

Example:

```
Router(config-pw)# encapsulation mpls
```

Specifies the tunneling encapsulation.

Configuring the Pseudowire Class Using Commands Associated with L2VPN Protocol-Based Feature

Note In simple configurations, this task is optional. You need not specify a pseudowire class if you specify the tunneling method as part of the **l2vpn xconnect context** command.

- You must specify the **encapsulation mpls** command as part of the pseudowire class or as part of the **l2vpn xconnect context** command for the AToM VCs to work properly. If you omit the **encapsulation mpls** command as part of the **l2vpn xconnect context**command, you receive the following error:

```
% Incomplete command.
```

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface pseudowire** *name*

Example:

```
Router(config)# interface pseudowire atom
```

Establishes an interface pseudowire with a name that you specify and enters pseudowire class configuration mode.

Step 4 **encapsulation mpls**

Example:

```
Router(config-pw-class)# encapsulation mpls
```

Specifies the tunneling encapsulation.

Step 5 **neighbor** *peer-address vcid-value*

Example:

```
Router(config-pw-class)#  neighbor 33.33.33.33 1
```

Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.

Changing the Encapsulation Type and Removing a Pseudowire

Once you specify the **encapsulation mpls** command, you cannot remove it using the **no encapsulation mpls** command.

Those methods result in the following error message:

```
Encapsulation changes are not allowed on an existing pw-class.
```

To remove the **encapsulation mpls** command, you must delete the pseudowire with the **no pseudowire-class** command.

To change the type of encapsulation, remove the pseudowire using the **no pseudowire-class** command and reconfigure the pseudowire to specify the new encapsulation type.

Changing the Encapsulation Type and Removing a Pseudowire Using Commands Associated with the L2VPN Protocol-Based Feature

Once you specify the **encapsulation mpls** command, you cannot remove it using the **no encapsulation mpls** command.

Those methods result in the following error message:

To remove the **encapsulation mpls** command, you must delete the pseudowire with the **no interface pseudowire** command.

To change the type of encapsulation, remove the pseudowire using the **no template type pseudowire** command and reconfigure the pseudowire to specify the new encapsulation type.

Configuring ATM AAL5 over MPLS

Configuring ATM AAL5 over MPLS on PVCs

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type slot* / *subslot* / *port* [**.** *subinterface*]

Example:

```
Router(config)# interface atm1/0/0
```

Specifies the interface type and enters interface configuration mode.

Step 4 **pvc** [*name*] *vpi* / *vci* **l2transport**

Example:

```
Router(config-if)# pvc 1/200 l2transport
```

Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.

- The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.

Step 5 **encapsulation aal5**

Example:

```
Router(config-if-atm-l2trans-pvc)# encapsulation aal5
```

Specifies ATM AAL5 encapsulation for the PVC. Make sure you specify the same encapsulation type on the PE and customer edge (CE) routers.

Step 6 **xconnect** *peer-router-id vcid* **encapsulation mpls**

Example:

```
Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls
```

Binds the attachment circuit to a pseudowire VC.

Step 7 **end**

Example:

```
Router(config-if-atm-l2trans-pvc)# end
```

Exits to privileged EXEC mode.

Step 8 **show mpls l2transport vc**

Example:

```
Router# show mpls l2transport vc
```

Displays output that shows ATM AAL5 over MPLS is configured on a PVC.

Examples

The following is sample output from the **show mpls l2transport vc** command that shows that ATM AAL5 over MPLS is configured on a PVC:

```
Router# show mpls l2transport vc
Local intf   Local circuit          Dest address      VC ID      Status
---------    -------------          ------------      -----      ------
ATM1/0       ATM AAL5 1/100         10.4.4.4           100        UP
```

Configuring ATM AAL5 over MPLS on PVCs using the commands associated with the L2VPN Protocol-Based CLIs feature

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type slot* / *subslot* / *port*[**.** *subinterface*]

Example:

```
Device(config)# interface atm1/0/0
```

Specifies the interface type and enters interface configuration mode.

Step 4 **pvc** [*name*] *vpi* / *vci* **l2transport**

Example:

```
Device(config-if)# pvc 1/200 l2transport
```

Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.

- The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.

Step 5 **encapsulation aal5**

Example:

```
Device(config-if-atm-l2trans-pvc)# encapsulation aal5
```

Specifies ATM AAL5 encapsulation for the PVC. Make sure you specify the same encapsulation type on the PE and customer edge (CE) routers.

Step 6 **end**

Example:

```
Device(config-if-atm-l2trans-pvc)# end
```

Exits to privileged EXEC mode.

Step 7 **interface pseudowire** *number*

Example:

```
Device(config)# interface pseudowire 100
```

Specifies the pseudowire interface and enters interface configuration mode.

Step 8 **encapsulation mpls**

Example:

```
Device(config-if)# encapsulation mpls
```

Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.

Step 9 **neighbor** *peer-address vcid-value*

Example:

```
Device(config-if)# neighbor 10.13.13.13 100
```

Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

Step 10 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode.

Step 11 **l2vpn xconnect context** *context-name*

Example:

```
Device(config)# l2vpn xconnect context con1
```

Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

Step 12 **member pseudowire** *interface-number*

Example:

```
Device(config-xconnect)# member pseudowire 100
```

Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

Step 13 **member atm** *interface-number* **pvc** *vpi* / *vci*

Example:

```
Device(config-xconnect)# member atm 100 pvc 1/200
```

Specifies the location of the ATM member interface.

Step 14 **end**

Example:

```
Device(config-xconnect)# end
```

Exits to privileged EXEC mode.

Step 15 **show l2vpn atom vc**

Example:

```
Device# show l2vpn atom vc
```

Displays output that shows ATM AAL5 over MPLS is configured on a PVC.

Examples

The following is sample output from the **show l2vpn atom vc** command that shows that ATM AAL5 over MPLS is configured on a PVC:

```
Device# show l2vpn atom vc
Local intf   Local circuit          Dest address      VC ID      Status
---------    -------------          ------------      -----      ------
ATM1/0       ATM AAL5 1/100         10.4.4.4           100        UP
```

Configuring ATM AAL5 over MPLS in VC Class Configuration Mode

---

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **vc-class atm** *vc-class-name*

Example:

```
Router(config)# vc-class atm aal5class
```

Creates a VC class and enters VC class configuration mode.

Step 4 **encapsulation** *layer-type*

Example:

```
Router(config-vc-class)# encapsulation aal5
```

Configures the AAL and encapsulation type.

Step 5 **exit**

Example:

```
Router(config-vc-class)# exit
```

Exits VC class configuration mode.

Step 6 **interface** *type slot* **/** *subslot* **/** *port* [**.** *subinterface*]

Example:

```
Router(config)# interface atm1/0/0
```

Specifies the interface type enters interface configuration mode.

Step 7 **class-int** *vc-class-name*

Example:

```
Router(config-if)# class-int aal5class
```

Applies a VC class to the ATM main interface or subinterface.

Note You can also apply a VC class to a PVC.

Step 8 **pvc** [*name*] *vpi* **/** *vci* **l2transport**

Example:

```
Router(config-if)# pvc 1/200 l2transport
```

Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.

- The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.

Step 9 **xconnect** *peer-router-id vcid* **encapsulation mpls**

Example:

```
Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls
```

Binds the attachment circuit to a pseudowire VC.

Step 10 **end**

Example:

```
Router(config-if-atm-l2trans-pvc)# end
```

Exits to privileged EXEC mode.

Step 11 **show atm class-links**

Example:

```
Router# show atm class-links
```

Displays the type of encapsulation and that the VC class was applied to an interface.

Examples

In the following example, the command output from the **show atm class-links** command verifies that ATM AAL5 over MPLS is configured as part of a VC class. The command output shows the type of encapsulation and that the VC class was applied to an interface.

```
Router# show atm class-links 1/100
Displaying vc-class inheritance for ATM1/0/0.0, vc 1/100:
no broadcast - Not configured - using default
encapsulation aal5 - VC-class configured on main interface
```

Configuring ATM AAL5 over MPLS in VC Class Configuration Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

---

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **vc-class atm** *vc-class-name*

Example:

```
Router(config)# vc-class atm aal5class
```

Creates a VC class and enters VC class configuration mode.

Step 4 **encapsulation** *layer-type*

Example:

```
Router(config-vc-class)# encapsulation aal5
```

Configures the AAL and encapsulation type.

Step 5 **exit**

Example:

```
Router(config-vc-class)# exit
```

Exits VC class configuration mode.

Step 6 **interface** *type slot* **/** *subslot* **/** *port* [**.** *subinterface*]

Example:

```
Router(config)# interface atm1/0/0
```

Specifies the interface type enters interface configuration mode.

Step 7 **class-int** *vc-class-name*

Example:

```
Router(config-if)# class-int aal5class
```

Applies a VC class to the ATM main interface or subinterface.

Note You can also apply a VC class to a PVC.

Step 8 **pvc** [*name*] *vpi* **/** *vci* **l2transport**

Example:

```
Router(config-if)# pvc 1/200 l2transport
```

Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.

- The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.

Step 9 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode.

Step 10 **interface pseudowire** *number*

Example:

```
Router(config)# interface pseudowire 100
```

Specifies the pseudowire interface and enters interface configuration mode.

Step 11 **encapsulation mpls**

Example:

```
Router(config-if)# encapsulation mpls
```

Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.

Step 12 **neighbor** *peer-address vcid-value*

Example:

```
Router(config-if)# neighbor 10.0.0.1 123
```

Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

Step 13 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode.

Step 14 **l2vpn xconnect context** *context-name*

Example:

```
Router(config)# l2vpn xconnect context con1
```

Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

Step 15 **member pseudowire** *interface-number*

Example:

```
Router(config-xconnect)# member pseudowire 100
```

Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

Step 16 **member atm** *interface-number*

Example:

```
Device(config-xconnect)# member atm 100
```

Specifies the location of the ATM member interface.

Step 17 **end**

Example:

```
Router(config-if-atm-l2trans-pvc)# end
```

Exits to privileged EXEC mode.

Step 18 **show atm class-links**

Example:

```
Router# show atm class-links
```

Displays the type of encapsulation and that the VC class was applied to an interface.

Examples

In the following example, the command output from the **show atm class-links** command verifies that ATM AAL5 over MPLS is configured as part of a VC class. The command output shows the type of encapsulation and that the VC class was applied to an interface.

```
Router# show atm class-links 1/100
Displaying vc-class inheritance for ATM1/0/0.0, vc 1/100:
no broadcast - Not configured - using default
encapsulation aal5 - VC-class configured on main interface
```

Configuring OAM Cell Emulation for ATM AAL5 over MPLS

Configuring OAM Cell Emulation for ATM AAL5 over MPLS on PVCs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / subslot / port* [**.** *subinterface*]
4. **pvc** [*name*] *vpi / vci* **l2transport**
5. **encapsulation aal5**
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**
7. **oam-ac emulation-enable** [*ais-rate*]
8. *oam-pvc manage* [*frequency*]
9. **end**
10. **show atm pvc**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / subslot / port* [**.** *subinterface*] **Example:**	Specifies the interface type enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface atm1/0/0`	
Step 4	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal5** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC. • Specify the same encapsulation type on the PE and CE routers.
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.
Step 7	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-if-atm-l2trans-pvc)# oam-ac emulation-enable 30`	Enables OAM cell emulation for AAL5 over MPLS. The *ais-rate* argument lets you specify the rate at which AIS cells are sent. The default is one cell every second. The range is 0 to 60 seconds.
Step 8	*oam-pvc manage* [*frequency*] **Example:** `Router(config-if-atm-l2trans-pvc)# oam-pvc manage`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. The optional *frequency* argument is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds.
Step 9	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits to privileged EXEC mode.
Step 10	**show atm pvc** **Example:** `Router# show atm pvc`	Displays output that shows OAM cell emulation is enabled on the ATM PVC.

Examples

The following output from the **show atm pvc** command shows that OAM cell emulation is enabled on the ATM PVC:

```
Router# show atm pvc 5/500
ATM4/1/0.200: VCD: 6, VPI: 5, VCI: 500
UBR, PeakRate: 1
AAL5-LLC/SNAP, etype:0x0, Flags: 0x34000C20, VCmode: 0x0
OAM Cell Emulation: enabled, F5 End2end AIS Xmit frequency: 1 second(s)
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not ManagedVerified
ILMI VC state: Not Managed
InPkts: 564, OutPkts: 560, InBytes: 19792, OutBytes: 19680
InPRoc: 0, OutPRoc: 0
InFast: 4, OutFast: 0, InAS: 560, OutAS: 560
InPktDrops: 0, OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
Out CLP=1 Pkts: 0
OAM cells received: 26
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 26
OAM cells sent: 77
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutAIS: 77, F5 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Configuring OAM Cell Emulation for ATM AAL5 over MPLS on PVCs using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *subslot* / *port* [**.** *subinterface*]
4. **pvc** [*name*] *vpi* / *vci* **l2transport**
5. **encapsulation aal5**
6. **exit**
7. **interface pseudowire** *number*
8. **encapsulation mpls**
9. **neighbor** *peer-address vcid-value*
10. **exit**
11. **l2vpn xconnect context** *context-name*
12. **member pseudowire** *interface-number*
13. **member atm** *interface-number* **pvc** *vpi* / *vci*
14. **exit**
15. **pvc** [*name*] *vpi* / *vci* **l2transport**
16. **oam-ac emulation-enable** [*ais-rate*]
17. *oam-pvc manage* [*frequency*]
18. **end**
19. **show atm pvc**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / subslot / port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Specifies the interface type enters interface configuration mode.
Step 4	**pvc** [*name*] *vpi / vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 5	**encapsulation aal5** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Specifies ATM AAL5 encapsulation for the PVC. • Specify the same encapsulation type on the PE and CE routers.
Step 6	**exit** **Example:** `Router(config-if-atm-l2trans-pvc)# exit`	Exits L2transport PVC configuration mode.
Step 7	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 8	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 9	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 11	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 12	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 13	**member atm** *interface-number* **pvc** *vpi* / *vci* **Example:** `Device(config-xconnect)# member atm 100 pvc 1/200`	Specifies the location of the ATM member interface.
Step 14	**exit** **Example:** `Router(config-xconnect)# exit`	Exits xconnect configuration mode.
Step 15	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.
Step 16	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-if-atm-l2trans-pvc)# oam-ac emulation-enable 30`	Enables OAM cell emulation for AAL5 over MPLS. The *ais-rate* argument lets you specify the rate at which AIS cells are sent. The default is one cell every second. The range is 0 to 60 seconds.
Step 17	*oam-pvc manage* [*frequency*] **Example:** `Router(config-if-atm-l2trans-pvc)# oam-pvc manage`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit. The optional *frequency* argument is the interval between transmission of loopback cells and ranges from 0 to 600 seconds. The default value is 10 seconds.
Step 18	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 19	**show atm pvc** **Example:** `Router# show atm pvc`	Displays output that shows OAM cell emulation is enabled on the ATM PVC.

Examples

The following output from the **show atm pvc** command shows that OAM cell emulation is enabled on the ATM PVC:

```
Router# show atm pvc 5/500
ATM4/1/0.200: VCD: 6, VPI: 5, VCI: 500
UBR, PeakRate: 1
AAL5-LLC/SNAP, etype:0x0, Flags: 0x34000C20, VCmode: 0x0
OAM Cell Emulation: enabled, F5 End2end AIS Xmit frequency: 1 second(s)
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not ManagedVerified
ILMI VC state: Not Managed
InPkts: 564, OutPkts: 560, InBytes: 19792, OutBytes: 19680
InPRoc: 0, OutPRoc: 0
InFast: 4, OutFast: 0, InAS: 560, OutAS: 560
InPktDrops: 0, OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
Out CLP=1 Pkts: 0
OAM cells received: 26
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 26
OAM cells sent: 77
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutAIS: 77, F5 OutRDI: 0
OAM cell drops: 0
Status: UP
```

Configuring OAM Cell Emulation for ATM AAL5 over MPLS in VC Class Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** *layer-type*
5. **oam-ac emulation-enable** [*ais-rate*]
6. **oam-pvc manage** [*frequency*]
7. **exit**
8. **interface** *type slot* / *subslot* / *port* [**.** *subinterface*]
9. **class-int** *vc-class-name*
10. **pvc** [*name*] *vpi* / *vci* **l2transport**
11. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm oamclass`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal5`	Configures the AAL and encapsulation type.
Step 5	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-vc-class)# oam-ac emulation-enable 30`	Enables OAM cell emulation for AAL5 over MPLS and specifies the rate at which AIS cells are sent.
Step 6	**oam-pvc manage** [*frequency*] **Example:** `Router(config-vc-class)# oam-pvc manage`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit.
Step 7	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 8	**interface** *type slot / subslot / port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Specifies the interface type and enters interface configuration mode.
Step 9	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int oamclass`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.

	Command or Action	Purpose
Step 10	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.
Step 11	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.

Configuring OAM Cell Emulation for ATM AAL5 over MPLS in VC Class Configuration Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** *layer-type*
5. **oam-ac emulation-enable** [*ais-rate*]
6. **oam-pvc manage** [*frequency*]
7. **exit**
8. **interface** *type slot* / *subslot* / *port* [**.** *subinterface*]
9. **class-int** *vc-class-name*
10. **pvc** [*name*] *vpi* / *vci* **l2transport**
11. **end**
12. **interface pseudowire** *number*
13. **encapsulation mpls**
14. **neighbor** *peer-address vcid-value*
15. **exit**
16. **l2vpn xconnect context** *context-name*
17. **member pseudowire** *interface-number*
18. **member atm** *interface-number*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm oamclass`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal5`	Configures the AAL and encapsulation type.
Step 5	**oam-ac emulation-enable** [*ais-rate*] **Example:** `Router(config-vc-class)# oam-ac emulation-enable 30`	Enables OAM cell emulation for AAL5 over MPLS and specifies the rate at which AIS cells are sent.
Step 6	**oam-pvc manage** [*frequency*] **Example:** `Router(config-vc-class)# oam-pvc manage`	Enables the PVC to generate end-to-end OAM loopback cells that verify connectivity on the virtual circuit.
Step 7	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 8	**interface** *type slot / subslot / port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Specifies the interface type and enters interface configuration mode.
Step 9	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int oamclass`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 10	**pvc** [*name*] *vpi / vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode. • The **l2transport** keyword indicates that the PVC is a switched PVC instead of a terminated PVC.

	Command or Action	Purpose
Step 11	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits to privileged EXEC mode.
Step 12	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 13	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 14	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 16	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 17	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 18	**member atm** *interface-number* **Example:** `Device(config-xconnect)# member atm 100`	Specifies the location of the ATM member interface.
Step 19	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring ATM Cell Relay over MPLS

Configuring ATM Cell Relay over MPLS in VC Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **pvc** *vpi* / *vci* **l2transport**
5. **encapsulation aal0**
6. **xconnect** *peer-router-id vcid* **encapsulation mpls**
7. **end**
8. **show atm vc**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 0/100 l2transport`	Assigns a virtual path identifier (VPI) and virtual circuit identifier (VCI) and enters L2transport VC configuration mode.
Step 5	**encapsulation aal0** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal0`	For ATM cell relay, specifies raw cell encapsulation for the interface. • Make sure you specify the same encapsulation type on the PE and CE routers.
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:**	Binds the attachment circuit to a pseudowire VC.

	Command or Action	Purpose
	`Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	
Step 7	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits to privileged EXEC mode.
Step 8	**show atm vc** **Example:** `Router# show atm vc`	Verifies that OAM cell emulation is enabled on the ATM VC.

Example

The following sample output from the **show atm vc** command shows that the interface is configured for VC mode cell relay:

```
Router# show atm vc 7
ATM3/0: VCD: 7, VPI: 23, VCI: 100
UBR, PeakRate: 149760
AAL0-Cell Relay, etype:0x10, Flags: 0x10000C2D, VCmode: 0x0
OAM Cell Emulation: not configured
InBytes: 0, OutBytes: 0
Status: UP
```

Configuring ATM Cell Relay over MPLS in VC Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **pvc** *vpi* / *vci* **l2transport**
5. **encapsulation aal0**
6. **end**
7. **interface pseudowire** *number*
8. **encapsulation mpls**
9. **neighbor** *peer-address vcid-value*
10. **exit**
11. **l2vpn xconnect context** *context-name*
12. **member pseudowire** *interface-number*
13. **member atm** *interface-number*
14. **end**
15. **show atm vc**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Specifies an ATM interface and enters interface configuration mode.
Step 4	**pvc** *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 0/100 l2transport`	Assigns a virtual path identifier (VPI) and virtual circuit identifier (VCI) and enters L2transport VC configuration mode.
Step 5	**encapsulation aal0** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal0`	For ATM cell relay, specifies raw cell encapsulation for the interface. • Make sure you specify the same encapsulation type on the PE and CE routers.
Step 6	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits to privileged EXEC mode.
Step 7	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 8	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 9	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 11	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 12	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 13	**member atm** *interface-number* **Example:** `Device(config-xconnect)# member atm 100`	Specifies the location of the ATM member interface.
Step 14	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.
Step 15	**show atm vc** **Example:** `Router# show atm vc`	Verifies that OAM cell emulation is enabled on the ATM VC.

Example

The following sample output from the **show atm vc** command shows that the interface is configured for VC mode cell relay:

```
Router# show atm vc 7
ATM3/0: VCD: 7, VPI: 23, VCI: 100
UBR, PeakRate: 149760
AAL0-Cell Relay, etype:0x10, Flags: 0x10000C2D, VCmode: 0x0
OAM Cell Emulation: not configured
InBytes: 0, OutBytes: 0
Status: UP
```

Configuring ATM Cell Relay over MPLS in VC Mode Using VC Class Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** *layer-type*
5. **exit**
6. **interface** *type slot* / *subslot* / *port* [**.** *subinterface*]
7. **class-int** *vc-class-name*
8. **pvc** [*name*] *vpi* / *vci* **l2transport**
9. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm cellrelay`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal0`	Configures the AAL and encapsulation type.
Step 5	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 6	**interface** *type slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Specifies the interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int cellrelay`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 8	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.13.13.13 100 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC.

Configuring ATM Cell Relay over MPLS in VC Mode Using VC Class Configuration Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **encapsulation** *layer-type*
5. **exit**
6. **interface** *type slot* / *subslot* / *port* [**.** *subinterface*]
7. **class-int** *vc-class-name*
8. **pvc** [*name*] *vpi* / *vci* **l2transport**
9. **end**
10. **interface pseudowire** *number*
11. **encapsulation mpls**
12. **neighbor** *peer-address vcid-value*
13. **exit**
14. **l2vpn xconnect context** *context-name*
15. **member pseudowire** *interface-number*
16. **member atm** *interface-number*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm cellrelay`	Creates a VC class and enters VC class configuration mode.
Step 4	**encapsulation** *layer-type* **Example:** `Router(config-vc-class)# encapsulation aal0`	Configures the AAL and encapsulation type.
Step 5	**exit** **Example:** `Router(config-vc-class)# exit`	Exits VC class configuration mode.
Step 6	**interface** *type slot / subslot / port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Specifies the interface type and enters interface configuration mode.
Step 7	**class-int** *vc-class-name* **Example:** `Router(config-if)# class-int cellrelay`	Applies a VC class to the ATM main interface or subinterface. **Note** You can also apply a VC class to a PVC.
Step 8	**pvc** [*name*] *vpi / vci* **l2transport** **Example:** `Router(config-if)# pvc 1/200 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.
Step 9	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits to privileged EXEC mode.
Step 10	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.

	Command or Action	Purpose
Step 11	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 12	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 14	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 15	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 16	**member atm** *interface-number* **Example:** `Device(config-xconnect)# member atm 100`	Specifies the location of the ATM member interface.
Step 17	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring ATM Cell Relay over MPLS in PVP Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **atm pvp** *vpi* **l2transport**
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**
6. **end**
7. **show atm vp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Defines the interface and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells and enters L2transport PVP configuration mode. • The **l2transport** keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvp)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.
Step 6	**end** **Example:** `Router(config-if-atm-l2trans-pvp)# end`	Exits to privileged EXEC mode.
Step 7	**show atm vp** **Example:** `Router# show atm vp`	Displays output that shows OAM cell emulation is enabled on the ATM VP.

Examples

The following output from the **show atm vp** command shows that the interface is configured for VP mode cell relay:

```
Router# show atm vp 1
ATM5/0  VPI: 1, Cell Relay, PeakRate: 149760, CesRate: 0, DataVCs: 1, CesVCs: 0, Status:
ACTIVE
```

```
  VCD    VCI   Type   InPkts   OutPkts   AAL/Encap     Status
  6      3     PVC    0        0         F4 OAM        ACTIVE
  7      4     PVC    0        0         F4 OAM        ACTIVE
TotalInPkts: 0, TotalOutPkts: 0, TotalInFast: 0, TotalOutFast: 0,
TotalBroadcasts: 0 TotalInPktDrops: 0, TotalOutPktDrops: 0
```

Configuring ATM Cell Relay over MPLS in PVP Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **atm pvp** *vpi* **l2transport**
5. **end**
6. **interface pseudowire** *number*
7. **encapsulation mpls**
8. **neighbor** *peer-address vcid-value*
9. **exit**
10. **l2vpn xconnect context** *context-name*
11. **member pseudowire** *interface-number*
12. **member atm** *interface-number* **pvp** *vpi*
13. **end**
14. **show atm vp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Defines the interface and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:**	Specifies that the PVP is dedicated to transporting ATM cells and enters L2transport PVP configuration mode.

	Command or Action	Purpose
	`Router(config-if)# atm pvp 1 l2transport`	• The **l2transport** keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits to privileged EXEC mode.
Step 6	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 8	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 10	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 11	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 12	**member atm** *interface-number* **pvp** *vpi* **Example:** `Device(config-xconnect)# member atm 100 pvp 1`	Specifies the location of the ATM member interface.
Step 13	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-xconnect)# end`	
Step 14	**show atm vp** **Example:** `Router# show atm vp`	Displays output that shows OAM cell emulation is enabled on the ATM VP.

Examples

The following output from the **show atm vp** command shows that the interface is configured for VP mode cell relay:

```
Router# show atm vp 1
ATM5/0  VPI: 1, Cell Relay, PeakRate: 149760, CesRate: 0, DataVCs: 1, CesVCs: 0, Status:
ACTIVE
  VCD    VCI   Type   InPkts   OutPkts   AAL/Encap     Status
  6      3     PVC    0        0         F4 OAM        ACTIVE
  7      4     PVC    0        0         F4 OAM        ACTIVE
TotalInPkts: 0, TotalOutPkts: 0, TotalInFast: 0, TotalOutFast: 0,
TotalBroadcasts: 0 TotalInPktDrops: 0, TotalOutPktDrops: 0
```

Configuring Ethernet over MPLS

Configuring Ethernet over MPLS in VLAN Mode to Connect Two VLAN Networks That Are in Different Locations.

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*]

Example:

```
Router(config)# interface gigabitethernet4/0/0.1
```

Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode.

- Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.

Step 4 **encapsulation dot1q** *vlan-id*

Example:

```
Router(config-subif)# encapsulation dot1q 100
```

Enables the subinterface to accept 802.1Q VLAN packets.

Step 5 **xconnect** *peer-router-id vcid* **encapsulation mpls**

Example:

```
Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls
```

Binds the attachment circuit to a pseudowire VC.

Configuring Ethernet over MPLS in VLAN Mode to Connect Two VLAN Networks That Are in Different Locations using the commands associated with the L2VPN Protocol-Based CLIs feature

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*]

Example:

```
Router(config)# interface gigabitethernet4/0/0.1
```

Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode.

- Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.

Step 4 **encapsulation dot1q** *vlan-id*

Example:

```
Router(config-subif)# encapsulation dot1q 100
```

Enables the subinterface to accept 802.1Q VLAN packets.

Step 5 **end**

Example:

```
Router(config-subif)# end
```

Exits to privileged EXEC mode.

Step 6 **interface pseudowire** *number*

Example:

```
Router(config)# interface pseudowire 100
```

Specifies the pseudowire interface and enters interface configuration mode.

Step 7 **encapsulation mpls**

Example:

```
Router(config-if)# encapsulation mpls
```

Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.

Step 8 **neighbor** *peer-address vcid-value*

Example:

```
Router(config-if)# neighbor 10.0.0.1 123
```

Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

Step 9 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode.

Step 10 **l2vpn xconnect context** *context-name*

Example:

```
Router(config)# l2vpn xconnect context con1
```

Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

Step 11 **member pseudowire** *interface-number*

Example:

```
Router(config-xconnect)# member pseudowire 100
```

Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

Step 12 **member gigabitethernet** *interface-number*

Example:

```
Router(config-xconnect)# member GigabitEthernet0/0/0.1
```

Specifies the location of the Gigabit Ethernet member interface.

Step 13 **end**

Example:

```
Router(config-xconnect)# end
```

Exits to privileged EXEC mode.

Configuring Ethernet over MPLS in Port Mode

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface gigabitethernet** *slot* **/** *subslot* **/** *port*

Example:

Specifies the Gigabit Ethernet interface and enters interface configuration mode.

Step 4 **xconnect** *peer-router-id vcid* **encapsulation mpls**

Example:

```
Router(config-if)# xconnect 10.0.0.1 123 encapsulation mpls
```

Binds the attachment circuit to a pseudowire VC.

Step 5 **end**

Example:

```
Router(config-if)# end
```

Exits to privileged EXEC mode.

Step 6 **show mpls l2transport vc**

Example:

```
Router# show mpls l2transport vc
```

Displays information about Ethernet over MPLS port mode.

Configuring Ethernet over MPLS in Port Mode Using Commands Associated with the L2VPN Protocol-Based Feature

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface gigabitethernet** *slot* / *subslot* / *port*[**.** *subinterface*]

Example:

```
Device(config)# interface gigabitethernet4/0/0
```

Specifies the Gigabit Ethernet interface and enters interface configuration mode.

- Make sure the interface on the adjoining CE router is on the same VLAN as this PE router.

Step 4 **end**

Example:

```
Device(config-if)# end
```

Exits to privileged EXEC mode.

Step 5 **interface pseudowire** *number*

Example:

```
Device(config)# interface pseudowire 100
```

Specifies the pseudowire interface and enters interface configuration mode.

Step 6 **encapsulation mpls**

Example:

```
Device(config-if)# encapsulation mpls
```

Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.

Step 7 **neighbor** *peer-address vcid-value*

Example:

```
Device(config-if)# neighbor 10.0.0.1 123
```

Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

Step 8 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode.

Step 9 **l2vpn xconnect context** *context-name*

Example:

```
Device(config)# l2vpn xconnect context con1
```

Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

Step 10 **member pseudowire** *interface-number*

Example:

```
Device(config-xconnect)# member pseudowire 100
```

Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

Step 11 **member gigabitethernet** *interface-number*

Example:

```
Device(config-xconnect)# member GigabitEthernet0/0/0.1
```

Specifies the location of the Gigabit Ethernet member interface.

Step 12 **end**

Example:

```
Device(config-xconnect)# end
```

Exits to privileged EXEC mode.

Step 13 **end**

Example:

```
Device(config-if)# end
```

Exits to privileged EXEC mode.

Step 14 **show l2vpn atom vc**

Example:

```
Device# show l2vpn atom vc
```

Displays information about Ethernet over MPLS port mode.

Configuring Ethernet over MPLS with VLAN ID Rewrite

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface gigabitethernet** *slot* / *subslot* / *port*

Example:

Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode.

Step 4 **encapsulation dot1q** *vlan-id*

Example:

```
Router(config-subif)# encapsulation dot1q 100
```

Enables the subinterface to accept 802.1Q VLAN packets.

Step 5 **xconnect** *peer-router-id vcid* **encapsulation mpls**

Example:

```
Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls
```

Binds the attachment circuit to a pseudowire VC and enters xconnect configuration mode.

Step 6 **remote circuit id** *remote-vlan-id*

Example:

```
Router(config-subif-xconn)# remote circuit id 101
```

(Optional) Enables you to use VLAN interfaces with different VLAN IDs at both ends of the tunnel.

Step 7 **end**

Example:

```
Router(config-subif-xconn)# end
```

Exits to privileged EXEC mode.

Configuring Ethernet over MPLS with VLAN ID Rewrite Using Commands Associated with the L2VPN Protocol-Based Feature

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **encapsulation dot1q** *vlan-id*

Example:

```
Router(config-subif)# encapsulation dot1q 100
```

Enables the subinterface to accept 802.1Q VLAN packets.

Step 4 **end**

Example:

```
Router(config-subif)# end
```

Exits to privileged EXEC mode.

Step 5 **interface pseudowire** *number*

Example:

```
Router(config)# interface pseudowire 100
```

Specifies the pseudowire interface and enters interface configuration mode.

Step 6 **encapsulation mpls**

Example:

```
Router(config-if)# encapsulation mpls
```

Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.

Step 7 **neighbor** *peer-address vcid-value*

Example:

```
Router(config-if)# neighbor 10.0.0.1 123
```

Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.

Step 8 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode.

Step 9 **l2vpn xconnect context** *context-name*

Example:

```
Router(config)# l2vpn xconnect context con1
```

Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

Step 10 **member pseudowire** *interface-number*

Example:

```
Router(config-xconnect)# member pseudowire 100
```

Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

Step 11 **member gigabitethernet** *interface-number*

Example:

Specifies the location of the Gigabit Ethernet member interface.

Step 12 **remote circuit id** *remote-vlan-id*

Example:

```
Router(config-xconnect)# remote circuit id 101
```

(Optional) Enables you to use VLAN interfaces with different VLAN IDs at both ends of the tunnel.

Step 13 **end**

Example:

```
Router(config-xconnect)# end
```

Exits to privileged EXEC mode.

Step 14 **show controllers eompls forwarding-table**

Example:

```
Router# show controllers eompls forwarding-table
```

Displays information about VLAN ID rewrite.

Configuring per-Subinterface MTU for Ethernet over MPLS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **mtu** *mtu-value*
5. **interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*]
6. **encapsulation dot1q** *vlan-id*
7. **xconnect** *peer-router-id vcid* **encapsulation mpls**
8. **mtu** *mtu-value*
9. **end**
10. **show mpls l2transport binding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface gigabitethernet4/0/0`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**mtu** *mtu-value* **Example:** `Router(config-if)# mtu 2000`	Specifies the MTU value for the interface. The MTU value specified at the interface level can be inherited by a subinterface.

	Command or Action	Purpose
Step 5	**interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config-if)# interface gigabitethernet4/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 6	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers need not be.
Step 7	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports. Enters xconnect subinterface configuration mode.
Step 8	**mtu** *mtu-value* **Example:** `Router(config-if-xconn)# mtu 1400`	Specifies the MTU for the VC.
Step 9	**end** **Example:** `Router(config-if-xconn)# end`	Exits to privileged EXEC mode.
Step 10	**show mpls l2transport binding** **Example:** `Router# show mpls l2transport binding`	Displays the MTU values assigned to the local and remote interfaces.

Configuring per-Subinterface MTU for Ethernet over MPLS using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port*[**.** *subinterface*]
4. **mtu** *mtu-value*
5. **interface gigabitethernet** *slot* / *subslot* / *port*[**.** *subinterface*]
6. **encapsulation dot1q** *vlan-id*
7. **end**
8. **interface pseudowire** *number*

9. **encapsulation mpls**
10. **neighbor** *peer-address vcid-value*
11. **mtu** *mtu-value*
12. **exit**
13. **l2vpn xconnect context** *context-name*
14. **member pseudowire** *interface-number*
15. **member gigabitethernet** *interface-number*
16. **end**
17. **show l2vpn atom binding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port*[**.** *subinterface*] **Example:** `Device(config)# interface gigabitethernet4/0/0`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**mtu** *mtu-value* **Example:** `Device(config-if)# mtu 2000`	Specifies the MTU value for the interface. The MTU value specified at the interface level can be inherited by a subinterface.
Step 5	**interface gigabitethernet** *slot* / *subslot* / *port*[**.** *subinterface*] **Example:** `Device(config-if)# interface gigabitethernet4/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 6	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers need not be.
Step 7	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-subif)# end`	
Step 8	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 9	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 10	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 11	**mtu** *mtu-value* **Example:** `Device(config-if)# mtu 1400`	Specifies the MTU for the VC.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 13	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 14	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 15	**member gigabitethernet** *interface-number* **Example:** `Device(config-xconnect)# member GigabitEthernet0/0/0.1`	Specifies the location of the Gigabit Ethernet member interface.
Step 16	**end** **Example:** `Device(config-xconnect)# end`	Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 17	**show l2vpn atom binding** **Example:** `Device# show l2vpn atom binding`	Displays Layer 2 VPN (L2VPN) Any Transport over MPLS (AToM) label binding information.

Configuring Frame Relay over MPLS

Configuring Frame Relay over MPLS with DLCI-to-DLCI Connections

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **frame-relay switching**
4. **interface serial** *slot* / *subslot* / *port* [**.** *subinterface*]
5. **encapsulation frame-relay** [**cisco** | **ietf**]
6. **frame-relay intf-type dce**
7. **exit**
8. **connect** *connection-name interface dlci* **l2transport**
9. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**frame-relay switching** **Example:** `Router(config)# frame-relay switching`	Enables PVC switching on a Frame Relay device.
Step 4	**interface serial** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface serial3/1/0`	Specifies a serial interface and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Router(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. You can specify different types of encapsulations. You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 6	**frame-relay intf-type dce** **Example:** `Router(config-if)# frame-relay intf-type dce`	Specifies that the interface is a DCE switch. You can also specify the interface to support Network-to-Network Interface (NNI) and DTE connections.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits from interface configuration mode.
Step 8	**connect** *connection-name interface dlci* **l2transport** **Example:** `Router(config)# connect fr1 serial5/0 1000 l2transport`	Defines connections between Frame Relay PVCs and enters connect configuration mode. Using the **l2transport** keyword specifies that the PVC will not be a locally switched PVC, but will be tunneled over the backbone network. The *connection-name* argument is a text string that you provide. The *interface* argument is the interface on which a PVC connection will be defined. The *dlci* argument is the DLCI number of the PVC that will be connected.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets. In a DLCI-to DLCI connection type, Frame Relay over MPLS uses the **xconnect** command in connect configuration mode.

Configuring Frame Relay over MPLS with DLCI-to-DLCI Connections using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **frame-relay switching**
4. **interface serial** *slot* / *subslot* / *port* [**.** *subinterface*]
5. **encapsulation frame-relay** [**cisco** | **ietf**]
6. **frame-relay intf-type dce**
7. **exit**

8. **connect** *connection-name interface dlci* **l2transport**
9. **end**
10. **interface pseudowire** *number*
11. **encapsulation mpls**
12. **neighbor** *peer-address vcid-value*
13. **exit**
14. **l2vpn xconnect context** *context-name*
15. **member pseudowire** *interface-number*
16. **member** *ip-address vc-id* **encapsulation mpls**
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**frame-relay switching** **Example:** `Router(config)# frame-relay switching`	Enables PVC switching on a Frame Relay device.
Step 4	**interface serial** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface serial3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 5	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Router(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. You can specify different types of encapsulations. You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 6	**frame-relay intf-type dce** **Example:** `Router(config-if)# frame-relay intf-type dce`	Specifies that the interface is a DCE switch. You can also specify the interface to support Network-to-Network Interface (NNI) and DTE connections.
Step 7	**exit** **Example:**	Exits from interface configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	
Step 8	**connect** *connection-name interface dlci* **l2transport** **Example:** `Router(config)# connect fr1 serial5/0 1000 l2transport`	Defines connections between Frame Relay PVCs and enters connect configuration mode. Using the **l2transport**keyword specifies that the PVC will not be a locally switched PVC, but will be tunneled over the backbone network. The *connection-name*argument is a text string that you provide. The *interface*argument is the interface on which a PVC connection will be defined. The *dlci*argument is the DLCI number of the PVC that will be connected.
Step 9	**end** **Example:** `Router(config-xconnect-conn-config)# end`	Exits to privileged EXEC mode.
Step 10	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 11	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 12	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 14	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 15	**member pseudowire** *interface-number* **Example:**	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

	Command or Action	Purpose
	`Router(config-xconnect)# member pseudowire 100`	
Step 16	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 17	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring Frame Relay over MPLS with Port-to-Port Connections

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **encapsulation hdlc**
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface serial5/0/0`	Specifies a serial interface and enters interface configuration mode.
Step 4	**encapsulation hdlc** **Example:** `Router(config-if)# encapsulation hdlc`	Specifies that Frame Relay PDUs will be encapsulated in HDLC packets.

	Command or Action	Purpose
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Configuring Frame Relay over MPLS with Port-to-Port Connections using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **encapsulation hdlc**
5. **end**
6. **interface pseudowire** *number*
7. **encapsulation mpls**
8. **neighbor** *peer-address vcid-value*
9. **exit**
10. **l2vpn xconnect context** *context-name*
11. **member pseudowire** *interface-number*
12. **member** *ip-address vc-id* **encapsulation mpls**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface serial5/0/0`	Specifies a serial interface and enters interface configuration mode.
Step 4	**encapsulation hdlc** **Example:**	Specifies that Frame Relay PDUs will be encapsulated in HDLC packets.

	Command or Action	Purpose
	`Router(config-if)# encapsulation hdlc`	
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 6	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 8	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 10	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 11	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 12	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 13	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring HDLC or PPP over MPLS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot* / *subslot* / *port* [**.** *subinterface*]
4. Do one of the following:
   - **encapsulation ppp**
   - **encapsulation hdlc**
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface serial5/0/0`	Specifies a serial interface and enters interface configuration mode.

Step 4	Do one of the following:
	• **encapsulation ppp**
	• **encapsulation hdlc**
	Example:
	`Router(config-if)# encapsulation ppp`
	Example:
	`or`
	Example:
	Example:
		Specifies HDLC or PPP encapsulation and enters connect configuration mode.

	Command or Action	Purpose
	`Router(config-if)# encapsulation hdlc`	
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Configuring HDLC or PPP over MPLS using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot* / *subslot* / *port* [**.** *subinterface*]
4. Do one of the following:
   - **encapsulation ppp**
   - **encapsulation hdlc**
5. **end**
6. **interface pseudowire** *number*
7. **encapsulation mpls**
8. **neighbor** *peer-address vcid-value*
9. **exit**
10. **l2vpn xconnect context** *context-name*
11. **member pseudowire** *interface-number*
12. **member** *ip-address vc-id* **encapsulation mpls**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface serial** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface serial5/0/0`	Specifies a serial interface and enters interface configuration mode.

Step 4	Do one of the following:
	• **encapsulation ppp**
	• **encapsulation hdlc**
	Example:
	`Router(config-if)# encapsulation ppp`
	Example:
	`Router(config-if)# encapsulation hdlc`
		Specifies HDLC or PPP encapsulation and enters connect configuration mode.

Step 5	**end** **Example:** `Router(config-xconnect-conn-config)# end`	Exits to privileged EXEC mode.
Step 6	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 8	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 10	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

	Command or Action	Purpose
Step 11	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 12	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 13	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring Tunnel Selection

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **pseudowire-class** *name*

Example:

```
Router(config)# pseudowire-class ts1
```

Establishes a pseudowire class with a name that you specify and enters pseudowire configuration mode.

Step 4 **encapsulation mpls**

Example:

```
Router(config-pw)# encapsulation mpls
```

Specifies the tunneling encapsulation. For AToM, the encapsulation type is mpls.

Step 5 **preferred-path** {**interface tunnel** *tunnel-number* | **peer**{*ip-address* | *host-name*}} [**disable-fallback**]

Example:

```
Router(config-pw)# preferred path peer 10.18.18.18
```

Specifies the MPLS traffic engineering tunnel or IP address or hostname to be used as the preferred path.

Step 6 **exit**

Example:

```
Router(config-pw)# exit
```

Exits from pseudowire configuration mode and enables the Tunnel Selection feature.

Step 7 **interface** *type slot* / *subslot* / *port*

Example:

```
Router(config)# interface atm1/1/0
```

Specifies an interface type and enters interface configuration mode.

Step 8 **encapsulation** *encapsulation-type*

Example:

```
Router(config-if)# encapsulation aal5
```

Specifies the encapsulation for the interface.

Step 9 **xconnect** *peer-router-id vcid* **pw-class** *name*

Example:

```
Router(config-if)# xconnect 10.0.0.1 123 pw-class ts1
```

Binds the attachment circuit to a pseudowire VC.

Examples

In the following sample output from the **show mpls l2transport vc** command includes the following information about the VCs:

- VC 101 has been assigned a preferred path called Tunnel1. The default path is disabled, because the preferred path specified that the default path should not be used if the preferred path fails.
- VC 150 has been assigned an IP address of a loopback address on PE2. The default path can be used if the preferred path fails.

Command output that is in boldface font shows the preferred path information.

```
Router# show mpls l2transport vc detail
Local interface: Gi0/0/0.1 up, line protocol up, Eth VLAN 222 up
  Destination address: 10.16.16.16, VC ID: 101, VC status: up
    Preferred path: Tunnel1,  active
```

```
    Default path: disabled
    Tunnel label: 3, next hop point2point
    Output interface: Tu1, imposed label stack {17 16}
  Create time: 00:27:31, last status change time: 00:27:31
  Signaling protocol: LDP, peer 10.16.16.16:0 up
    MPLS VC labels: local 25, remote 16
    Group ID: local 0, remote 6
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 10, send 10
    byte totals:   receive 1260, send 1300
    packet drops:  receive 0, send 0
Local interface: ATM1/0/0 up, line protocol up, ATM AAL5 0/50 up
  Destination address: 10.16.16.16, VC ID: 150, VC status: up
    Preferred path: 10.18.18.18, active
    Default path: ready
    Tunnel label: 3, next hop point2point
    Output interface: Tu2, imposed label stack {18 24}
  Create time: 00:15:08, last status change time: 00:07:37
  Signaling protocol: LDP, peer 10.16.16.16:0 up
    MPLS VC labels: local 26, remote 24
    Group ID: local 2, remote 0
    MTU: local 4470, remote 4470
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

Troubleshooting Tips

To debug ATM cell packing, issue the **debug atm cell-packing** command.

Configuring Tunnel Selection Using Commands Associated with L2VPN Protocol-Based Feature

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **template type pseudowire** *name*

Example:

```
Router(config)# template type pseudowire ts1
```

Creates a template pseudowire with a name that you specify and enters pseudowire configuration mode.

Step 4 **encapsulation mpls**

Example:

```
Router(config-pw)# encapsulation mpls
```

Specifies the tunneling encapsulation. For AToM, the encapsulation type is mpls.

Step 5 **preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *hostname*}} [**disable-fallback**]

Example:

```
Router(config-pw)# preferred path peer 10.18.18.18
```

Specifies the MPLS traffic engineering tunnel or IP address or hostname to be used as the preferred path.

Step 6 **exit**

Example:

```
Router(config-pw)# exit
```

Exits from pseudowire configuration mode and enables the Tunnel Selection feature.

Step 7 **interface** *type slot* / *subslot* / *port*[**.** *subinterface*]

Example:

```
Router(config)# interface atm1/1/0
```

Specifies an interface type and enters interface configuration mode.

Step 8 **encapsulation** *encapsulation-type*

Example:

```
Router(config-if)# encapsulation aal5
```

Specifies the encapsulation for the interface.

Step 9 **end**

Example:

```
Router(config-if)# end
```

Exits to privileged EXEC mode.

Step 10 **interface pseudowire** *number*

Example:

```
Router(config)# interface pseudowire 100
```

Specifies the pseudowire interface and enters interface configuration mode.

Step 11 **source template type pseudowire** *name*

Example:

```
Router(config-if)# source template type pseudowire ts1
```

Configures the source template of type pseudowire named ts1.

Step 12 **neighbor** *peer-address vcid-value*

Example:

```
Router(config-if)#  neighbor 10.0.0.1 123
```

Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.

Step 13 **end**

Example:

```
Router(config-if)# end
```

Exits to privileged EXEC mode.

Step 14 **l2vpn xconnect context** *context-name*

Example:

```
Router(config)# l2vpn xconnect context con1
```

Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

Step 15 **member pseudowire** *interface-number*

Example:

```
Router(config-xconnect)# member pseudowire 100
```

Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

Step 16 **member** *ip-address vc-id* **encapsulation mpls**

Example:

```
Router(config-xconnect)# member 10.0.0.1 123 encapsulation mpls
```

Creates the VC to transport the Layer 2 packets.

Step 17 **end**

Example:

```
Router(config-xconnect)# end
```

Exits to privileged EXEC mode.

Troubleshooting Tips using the commands associated with the L2VPN Protocol-Based CLIs feature

You can use the **debug l2vpn atom vc event** command to troubleshoot tunnel selection. For example, if the tunnel interface that is used for the preferred path is shut down, the default path is enabled. The **debug l2vpn atom vc event** command provides the following output:

```
AToM SMGR [10.2.2.2, 101]: Processing imposition update, vc_handle 62091860, update_action
 3, remote_vc_label 16
AToM SMGR [10.2.2.2, 101]: selected route no parent rewrite: tunnel not up
AToM SMGR [10.2.2.2, 101]: Imposition Programmed, Output Interface: Et3/2
```

Setting Experimental Bits with AToM

Note Only EoMPLS and CEM is supported .

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **class-map** *class-name*

Example:

```
Router(config)# class-map class1
```

Specifies the user-defined name of the traffic class and enters class map configuration mode.

Step 4 **match any**

Example:

```
Router(config-cmap)# match any
```

Specifies that all packets will be matched. Use only the **any** keyword. Other keywords might cause unexpected results.

Step 5 **policy-map** *policy-name*

Example:

```
Router(config-cmap)# policy-map policy1
```

Specifies the name of the traffic policy to configure and enters policy-map configuration mode.

Step 6 **class** *class-name*

Example:

```
Router(config-pmap)# class class1
```

Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy and enters policy-map class configuration mode.

Step 7 **set mpls experimental** *value*

Example:

```
Router(config-pmap-c)# set mpls experimental 7
```

Designates the value to which the MPLS bits are set if the packets match the specified policy map.

Step 8 **exit**

Example:

```
Router(config-pmap-c)# exit
```

Exits policy-map class configuration mode.

Step 9 **exit**

Example:

```
Router(config-pmap)# exit
```

Exits policy-map configuration mode.

Step 10 **interface** *type slot / subslot / port*

Example:

```
Router(config)# interface atm1/0/0
```

Specifies the interface type and enters interface configuration mode.

Step 11 **service-policy input** *policy-name*

Example:

```
Router(config-if)# service-policy input policy1
```

Attaches a traffic policy to an interface.

Step 12 **end**

Example:

```
Router(config-if)# end
```

Exits to privileged EXEC mode.

Step 13 **show policy-map interface** *interface-name* [*vc* [*vpi /*] *vci*] [**dlci** *dlci*] [**input** | **output**]

Example:

```
Router# show policy-map interface serial3/0/0
```

Displays the traffic policy attached to an interface.

Enabling the Control Word

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class cw_enable**
4. **encapsulation mpls**
5. **control-word**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class cw_enable** **Example:** `Router(config)# pseudowire-class cw_enable`	Enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is MPLS.
Step 5	**control-word** **Example:** `Router(config-pw-class)# control-word`	Enables the control word.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-pw-class)# end`	Exits to privileged EXEC mode.

Enabling the Control Word using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pseudowire** *number*
4. **encapsulation mpls**
5. **control-word include**
6. **neighbor** *peer-address vcid-value*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 1`	Creates an interface pseudowire with a value that you specify and enters pseudowire configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is mpls.
Step 5	**control-word include** **Example:**	Enables the control word.

	Command or Action	Purpose
	`Router(config-pw)# control-word include`	
Step 6	**neighbor** *peer-address vcid-value* **Example:** `Router(config-pw)#  neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 7	**end** **Example:** `Router(config-pw)# end`	Exits to privileged EXEC mode.

Configuring MPLS AToM Remote Ethernet Port Shutdown

Note The Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown feature is automatically enabled by default when an image with the feature supported is loaded on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **exit**
6. **xconnect** *peer-ip-address vc-id pw-class pw-class-name*
7. **no remote link failure notification**
8. **remote link failure notification**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config)# pseudowire-class eompls`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies that MPLS is used as the data encapsulation method for tunneling Layer 2 traffic over the pseudowire.
Step 5	**exit** **Example:** `Router(config-pw)# exit`	Exits to global configuration mode.
Step 6	**xconnect** *peer-ip-address vc-id pw-class pw-class-name* **Example:** `Router(config-if)# xconnect 10.1.1.1 1 pw-class eompls`	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.
Step 7	**no remote link failure notification** **Example:** `Router(config-if-xconn)# remote link failure notification`	Disables MPLS AToM remote link failure notification and shutdown.
Step 8	**remote link failure notification** **Example:** `Router(config-if-xconn)# remote link failure notification`	Enables MPLS AToM remote link failure notification and shutdown.
Step 9	**end** **Example:** `Router(config-if-xconn)# end`	Exits to privileged EXEC mode.

Configuring MPLS AToM Remote Ethernet Port Shutdown using the commands associated with the L2VPN Protocol-Based CLIs feature

Note The Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown feature is automatically enabled by default when an image with the feature supported is loaded on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** [*pseudowire-name*]
4. **encapsulation mpls**
5. **exit**
6. **interface** *type slot / subslot / port*
7. **interface pseudowire** *number*
8. **source template type pseudowire**
9. **neighbor** *peer-address vcid-value*
10. **end**
11. **l2vpn xconnect context** *context-name*
12. **no remote link failure notification**
13. **remote link failure notification**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** [*pseudowire-name*] **Example:** `Device(config)# template type pseudowire eompls`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw)# encapsulation mpls`	Specifies that MPLS is used as the data encapsulation method for tunneling Layer 2 traffic over the pseudowire.
Step 5	**exit** **Example:** `Device(config-pw)# exit`	Exits to global configuration mode.
Step 6	**interface** *type slot / subslot / port* **Example:**	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface GigabitEthernet1/0/0`	
Step 7	**interface pseudowire** *number* **Example:** `Device(config-if)# interface pseudowire 100`	Specifies the pseudowire interface.
Step 8	**source template type pseudowire** **Example:** `Device(config-if)# source template type pseudowire eompls`	Configures the source template of type pseudowire named eompls.
Step 9	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)#  neighbor 10.1.1.1 1`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.
Step 11	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 12	**no remote link failure notification** **Example:** `Device(config-xconnect)# no remote link failure notification`	Disables MPLS AToM remote link failure notification and shutdown.
Step 13	**remote link failure notification** **Example:** `Device(config-xconnect)# remote link failure notification`	Enables MPLS AToM remote link failure notification and shutdown.
Step 14	**end** **Example:** `Device(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring AToM Load Balancing with Single PW

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *pw-class-name*
4. **encapsulation mpls**
5. **load-balance flow**
6. **xconnect** *url* **pw-class** *pw-class-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *pw-class-name* **Example:** `Router(config)# pseudowire-class ecmp-class`	Establishes a pseudowire class with a name that you specify, and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is mpls.
Step 5	**load-balance flow** **Example:** `Router(config-pw-class)# load-balance flow`	Enables the AToM Load Balancing with Single PW feature so that load balancing is done on a per-flow basis.
Step 6	**xconnect** *url* **pw-class** *pw-class-name* **Example:** `Router(config-pw-class)# xconnect 10.0.0.1 pw-class ecmp-class`	Binds the attachment circuit to a pseudowire virtual circuit, and enters xconnect configuration mode. • The syntax for this command is the same as for all other Layer 2 transports.

Configuring AToM Load Balancing with Single PW using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** [*pseudowire-name*]
4. **encapsulation mpls**
5. **load-balance flow**
6. **end**
7. **interface pseudowire** *number*
8. **source template type pseudowire**
9. **neighbor** *peer-address vcid-value*
10. **end**
11. **l2vpn xconnect context** *context-name*
12. **member pseudowire** *interface-number*
13. **member** *ip-address vc-id* **encapsulation mpls**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire eompls`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is mpls.
Step 5	**load-balance flow** **Example:**	Enables the AToM Load Balancing with Single PW feature so that load balancing is done on a per-flow basis.

	Command or Action	Purpose
	`Router(config-pw-class)# load-balance flow`	
Step 6	**end** **Example:** `Router(config-pw-class)# end`	Exits to privileged EXEC mode.
Step 7	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 8	**source template type pseudowire** **Example:** `Router(config-if)# source template type pseudowire ether-pw`	Configures the source template of type pseudowire named ether-pw.
Step 9	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)#  neighbor 10.1.1.1 1`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 10	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 11	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 12	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 13	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 14	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-xconnect)# end`	

Configuring Flow-Aware Transport (FAT) Load Balancing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pseudowire** *name*
4. **encapsulation mpls**
5. **neighbor** *peer-address vcid-value*
6. **signaling protocol ldp**
7. **load-balance flow-label both**
8. **end**
9. **show l2vpn atom vc detail**
10. **show ssm id**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface pseudowire** *name* **Example:** `Device(config)# interface pseudowire 1001`	Establishes a pseudowire with a name that you specify, and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is mpls.
Step 5	**neighbor** *peer-address vcid-value* **Example:** `Device(config-pw-class)# neighbor 10.1.1.200 200`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.

	Command or Action	Purpose
Step 6	**signaling protocol ldp** **Example:** `Device(config-pw-class)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.
Step 7	**load-balance flow-label both** **Example:** `Device(config-pw-class)# load-balance flow-label both`	Enables the Flow-Aware Transport of MPLS Pseudowire feature and specifies how flow labels are used. It is recommended that you use both as the option for flow-label. However, if you choose not to use both, you can either use **load-balance flow-label transmit** or **load-balance flow-label receive** if necessary.
Step 8	**end** **Example:** `Device(config-pw-class)# end`	Exits to privileged EXEC mode.
Step 9	**show l2vpn atom vc detail** **Example:** `Device# show l2vpn atom vc detail`	Displays detailed output that shows information about the flow labels configured for the pseudowire.
Step 10	**show ssm id** **Example:** `Device# show ssm id`	Displays information for all Segment Switching Manager (SSM) IDs.

Examples

The following is sample output from the **show mpls l2transport vc 1 detail** command that shows information about the VC details:

```
Device# show mpls l2transport vc 1 detail

Local interface: Te0/5/2 up, line protocol up, Eth VLAN 1 up
  Interworking type is Ethernet
  Destination address: 4.4.4.4, VC ID: 1, VC status: up
    Output interface: BD12, imposed label stack {23 16}
    Preferred path: not configured
    Default path: active
    Next hop: 12.0.0.2
  Create time: 23:12:54, last status change time: 23:09:05
    Last label FSM state change time: 23:09:02
  Signaling protocol: LDP, peer 4.4.4.4:0 up
    Targeted Hello: 1.1.1.1(LDP Id) -> 4.4.4.4, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
```

```
    Last BFD dataplane     status rcvd: Not sent
    Last BFD peer monitor  status rcvd: No fault
    Last local AC  circuit status rcvd: No fault
    Last local AC  circuit status sent: No fault
    Last local PW i/f circ status rcvd: No fault
    Last local LDP TLV     status sent: No fault
    Last remote LDP TLV    status rcvd: No fault
    Last remote LDP ADJ    status rcvd: No fault
  MPLS VC labels: local 27, remote 16
  Group ID: local 8, remote 8
  MTU: local 9216, remote 9216
  Remote interface description:
Sequencing: receive disabled, send disabled
Control Word: On
SSO Descriptor: 4.4.4.4/1, local label: 27
Dataplane:
  SSM segment/switch IDs: 32854/4116 (used), PWID: 1
VC statistics:
  transit packet totals: receive 0, send 0
  transit byte totals:   receive 0, send 0
  transit packet drops:  receive 0, seq error 0, send 0
```

The following is sample output from the **show ssm id** command that shows information for all Segment Switching Manager (SSM) IDs:

```
Device# show ssm id

SSM Status: 1 switch
  Switch-ID 4096 State: Open
    Segment-ID: 8194 Type: Eth[2]
      Switch-ID:                    4096
      Physical intf:                Local
      Allocated By:                 This CPU
      Locked By:                    SIP     [1]
      Circuit status:               UP      [1]
    Class:                         SSS
      State:                         Active
      AC Switching Context:          Et0/0
      SSS Info : Switch Handle 2583691265 Ckt 0xC36A59E0
      Interworking 0 Encap Len 0 Boardencap Len 0 MTU 1500
      Flow Classification src-dst-mac
      AC Encap [0 bytes]
    Class:                        ADJ
      State:                         Active
      AC Adjacency context:
      adjacency = 0xC36B6100 [complete] RAW Ethernet0/0:0
      AC Encap [0 bytes]
      1stMem: 8194 2ndMem: 0 ActMem: 8194

    Segment-ID: 4097 Type: AToM[17]
      Switch-ID:                    4096
      Allocated By:                 This CPU
      Locked By:                    SIP     [1]
    Class:                         SSS
      State:                         Active
    Class:                        ADJ
      State:                         Active
```

Configuring Flow-Aware Transport (FAT) Load Balancing using a template

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** [*pseudowire-name*]
4. **encapsulation mpls**
5. **load-balance flow**
6. **load-balance flow-label**
7. **end**
8. **interface pseudowire** *number*
9. **source template type pseudowire**
10. **encapsulation mpls**
11. **neighbor** *peer-address vcid-value*
12. **signaling protocol ldp**
13. **end**
14. **show l2vpn atom vc detail**
15. **show ssm id**
16. **show mpls forwarding-table exact-route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** [*pseudowire-name*] **Example:** `Device(config)# template type pseudowire fatpw`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is MPLS.
Step 5	**load-balance flow** **Example:**	Enables the AToM Load Balancing with Single PW feature so that load balancing is done on a per-flow basis.

	Command or Action	Purpose
	`Device(config-pw-class)# load-balance flow`	
Step 6	**load-balance flow-label** **Example:** `Device(config-pw-class)# load-balance flow-label both`	Enables the Flow-Aware Transport of MPLS Pseudowires feature and specifies how flow labels are to be used.
Step 7	**end** **Example:** `Device(config-pw-class)# end`	Exits to privileged EXEC mode.
Step 8	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 9	**source template type pseudowire** **Example:** `Device(config-if)# source template type pseudowire fatpw`	Configures the source template of type pseudowire named fatpw.
Step 10	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is MPLS.
Step 11	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)# neighbor 10.1.1.1 1`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 12	**signaling protocol ldp** **Example:** `Device(config-if)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.
Step 14	**show l2vpn atom vc detail** **Example:**	Displays detailed output that shows information about the flow labels configured for the pseudowire.

	Command or Action	Purpose
	`Device# show l2vpn atom vc detail`	
Step 15	**show ssm id** **Example:** `Device# show ssm id`	Displays information for all Segment Switching Manager (SSM) IDs.
Step 16	**show mpls forwarding-table exact-route** **Example:** `Device# show mpls forwarding-table exact-route label 32 ethernet source 001d.e558.5c1a dest 000e.8379.1c1b detail`	Displays the exact path for the source and destination address pair.

Examples

The following is sample output from the **show l2vpn atom vc detail** command that shows information about the flow labels configured for the pseudowire:

```
Device# show l2vpn atom vc detail

pseudowire100001 is up, VC status is up PW type: Ethernet
  Create time: 00:01:47, last status change time: 00:01:29
    Last label FSM state change time: 00:01:29
  Destination address: 10.1.1.151 VC ID: 100
    Output interface: Se3/0, imposed label stack {1001 100}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
    Load Balance: Flow
    flow classification: ethernet src-dst-mac
  Member of xconnect service Et0/0-2, group right
    Associated member Et0/0 is up, status is up
    Interworking type is Like2Like
    Service id: 0xcf000001
  Signaling protocol: LDP, peer 10.1.1.151:0 up
    Targeted Hello: 10.1.1.152(LDP Id) -> 10.1.1.151, LDP is UP
    Graceful restart: not configured and not enabled
    Non stop routing: not configured and not enabled
    PWid FEC (128), VC ID: 100
    Status TLV support (local/remote)         : enabled/supported
      LDP route watch                         : enabled
      Label/status state machine              : established, LruRru
      Local dataplane status received         : No fault
      BFD dataplane status received           : Not sent
      BFD peer monitor status received        : No fault
      Status received from access circuit     : No fault
      Status sent to access circuit           : No fault
      Status received from pseudowire i/f     : No fault
      Status sent to network peer             : No fault
      Status received from network peer       : No fault
      Adjacency status of remote peer         : No fault
  Sequencing: receive disabled, send disabled
  Bindings
    Parameter    Local                          Remote
```

```
  ------------ ------------------------------ ------------------------------
  Label        200                            100
  Group ID     0                              0
  Interface
  MTU          1500                           1500
  Control word on (configured: autosense)     on
  PW type      Ethernet                       Ethernet
  VCCV CV type 0x12                           0x12
               LSPV [2], BFD/Raw [5]          LSPV [2], BFD/Raw [5]
  VCCV CC type 0x07                           0x07
               CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
  Status TLV   enabled                        supported
  Flow label   enabled, T=1, R=0              enabled, T=1, R=1
Dataplane:
  SSM segment/switch IDs: 4097/4096 (used), PWID: 1
Rx Counters
  28 input transit packets, 2602 bytes
  0 drops, 0 seq err
Tx Counters
  31 output transit packets, 3694 bytes
  0 drops
```

The following is sample output from the **show ssm id** command that shows information for all Segment Switching Manager (SSM) IDs:

```
Device# show ssm id

SSM Status: 1 switch
  Switch-ID 4096 State: Open
    Segment-ID: 8194 Type: Eth[2]
      Switch-ID:                  4096
      Physical intf:              Local
      Allocated By:               This CPU
      Locked By:                  SIP     [1]
      Circuit status:             UP      [1]
    Class:                       SSS
      State:                       Active
      AC Switching Context:        Et0/0
      SSS Info : Switch Handle 2583691265 Ckt 0xC36A59E0
      Interworking 0 Encap Len 0 Boardencap Len 0 MTU 1500
      Flow Classification src-dst-mac
      AC Encap [0 bytes]
    Class:                       ADJ
      State:                       Active
      AC Adjacency context:
      adjacency = 0xC36B6100 [complete] RAW Ethernet0/0:0
      AC Encap [0 bytes]
      1stMem: 8194 2ndMem: 0 ActMem: 8194

    Segment-ID: 4097 Type: AToM[17]
      Switch-ID:                  4096
      Allocated By:               This CPU
      Locked By:                  SIP     [1]
    Class:                       SSS
      State:                       Active
    Class:                       ADJ
      State:                       Active
```

The following is sample output from the **show mpls forwarding-table exact-route** command that shows the exact path for the source and destination address pair:

```
Device# show mpls forwarding-table exact-route label 32 ethernet source 001d.e558.5c1a dest
 000e.8379.1c1b detail
```

```
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
32         No Label   l2ckt(66)        1163          Gi1/0/4    point2point
        MAC/Encaps=0/0, MRU=0, Label Stack{}
        No output feature configured
        Flow label: 227190
```

Configuration Examples for Any Transport over MPLS

Example: ATM over MPLS

The table below shows the configuration of ATM over MPLS on two PE routers.

Table 102: ATM over MPLS Configuration Example

PE1	PE2
mpls label protocol ldp mpls ldp router-id Loopback0 force ! interface Loopback0 ip address 10.16.12.12 255.255.255.255 ! interface ATM4/0/0 pvc 0/100 l2transport encapsulation aal0 xconnect 10.13.13.13 100 encapsulation mpls ! interface ATM4/0/0.300 point-to-point no ip directed-broadcast no atm enable-ilmi-trap pvc 0/300 l2transport encapsulation aal0 xconnect 10.13.13.13 300 encapsulation mpls	mpls label protocol ldp mpls ldp router-id Loopback0 force ! interface Loopback0 ip address 10.13.13.13 255.255.255.255 interface ATM4/0/0 pvc 0/100 l2transport encapsulation aal0 xconnect 10.16.12.12 100 encapsulation mpls ! interface ATM4/0/0.300 point-to-point no ip directed-broadcast no atm enable-ilmi-trap pvc 0/300 l2transport encapsulation aal0 xconnect 10.16.12.12 300 encapsulation mpls

Example: ATM over MPLS Using Commands Associated with L2VPN Protocol-Based Feature

The table below shows the configuration of ATM over MPLS on two PE routers.

Table 103: ATM over MPLS Configuration Example

PE1	PE2

PE1	PE2
mpls label protocol ldp	mpls label protocol ldp
mpls ldp router-id Loopback0 force	mpls ldp router-id Loopback0 force
!	!
interface Loopback0	interface Loopback0
ip address 10.16.12.12 255.255.255.255	ip address 10.13.13.13 255.255.255.255
!	
interface ATM4/0/0	interface ATM4/0/0
pvc 0/100 l2transport	pvc 0/100 l2transport
encapsulation aal0	encapsulation aal0
interface pseudowire 100	interface pseudowire 100
encapsulation mpls	encapsulation mpls
neighbor 10.0.0.1 123	neighbor 10.0.0.1 123
!	!
l2vpn xconnect context A	l2vpn xconnect context A
member pseudowire 100	member pseudowire 100
member atm 100	member atm 100
!	!
interface ATM4/0/0.300 point-to-point	interface ATM4/0/0.300 point-to-point
no atm enable-ilmi-trap	no ip directed-broadcast
pvc 0/300 l2transport	no atm enable-ilmi-trap
encapsulation aal0	pvc 0/300 l2transport
interface pseudowire 300	encapsulation aal0
encapsulation mpls	interface pseudowire 300
neighbor 10.0.0.1 123	encapsulation mpls

PE1	PE2
!	neighbor 10.0.0.1 123
l2vpn xconnect context A	!
member pseudowire 300	l2vpn xconnect context A
member atm 300	member pseudowire 300
	member atm 300

Example: Configuring ATM AAL5 over MPLS in VC Class Configuration Mode

The following example configures ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm aal5class
encapsulation aal5
interface atm1/0/0
class-int aal5class
pvc 1/200 l2transport
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example configures ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm aal5class
encapsulation aal5
interface atm1/0/0
pvc 1/200 l2transport
class-vc aal5class
xconnect 10.13.13.13 100 encapsulation mpls
```

Example: Configuring ATM AAL5 over MPLS in VC Class Configuration Mode Using Commands Associated with L2VPN Protocol-Based Feature

The following example configures ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm aal5class
encapsulation aal5
interface atm1/0/0
class-int aal5class
pvc 1/200 l2transport
interface pseudowire 100
encapsulation mpls
```

```
neighbor 10.0.0.1 123
exit
l2vpn xconnect context A
member pseudowire 100
member atm 100
exit
```

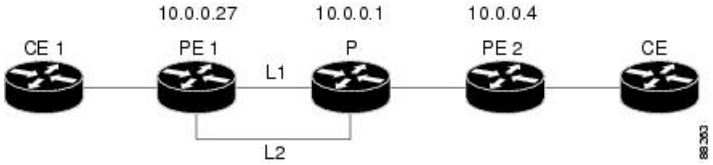
Example: Ethernet over MPLS with MPLS Traffic Engineering Fast Reroute

The following configuration example and the figure show the configuration of Ethernet over MPLS with fast reroute on AToM PE routers.

Routers PE1 and PE2 have the following characteristics:

- A TE tunnel called Tunnel41 is configured between PE1and PE2, using an explicit path through a link called L1. AToM VCs are configured to travel through the FRR-protected tunnel Tunnel41.
- The link L1 is protected by FRR, the backup tunnel is Tunnel1.
- PE2 is configured to forward the AToM traffic back to PE1 through the L2 link.

Figure 53: Fast Reroute Configuration



PE1 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback1 force
!
pseudowire-class T41
 encapsulation mpls
 preferred-path interface Tunnel41 disable-fallback
!
pseudowire-class IP1
 encapsulation mpls
 preferred-path peer 10.4.0.1 disable-fallback
!
interface Loopback1
 ip address 10.0.0.27 255.255.255.255
!
interface Tunnel1
 ip unnumbered Loopback1
 tunnel destination 10.0.0.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 10000
 tunnel mpls traffic-eng path-option 1 explicit name FRR
!
interface Tunnel41
 ip unnumbered Loopback1
 tunnel destination 10.0.0.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
```

```
 tunnel mpls traffic-eng path-option 1 explicit name name-1
 tunnel mpls traffic-eng fast-reroute
!
interface POS0/0/0
 description pe1name POS8/0/0
 ip address 10.1.0.2 255.255.255.252
 mpls traffic-eng tunnels
 mpls traffic-eng backup-path Tunnel1
 crc 16
 clock source internal
 pos ais-shut
 pos report lrdi
 ip rsvp bandwidth 155000 155000
!
interface POS0/3/0
 description pe1name POS10/1/0
 ip address 10.1.0.14 255.255.255.252
 mpls traffic-eng tunnels
 crc 16
 clock source internal
 ip rsvp bandwidth 155000 155000
!
interface gigabitethernet3/0/0.1
 encapsulation dot1Q 203
 xconnect 10.0.0.4 2 pw-class IP1
!
interface gigabitethernet3/0/0.2
 encapsulation dot1Q 204
 xconnect 10.0.0.4 4 pw-class T41
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
!
ip classless
ip route 10.4.0.1 255.255.255.255 Tunnel41
!
ip explicit-path name xxxx-1 enable
 next-address 10.4.1.2
 next-address 10.1.0.10
```

P Configuration

```
ip cef
mpls traffic-eng tunnels
!
interface Loopback1
 ip address 10.0.0.1 255.255.255.255
!
interface FastEthernet1/0/0
 ip address 10.4.1.2 255.255.255.0
 mpls traffic-eng tunnels
 ip rsvp bandwidth 10000 10000
!
interface POS8/0/0
 description xxxx POS0/0
 ip address 10.1.0.1 255.255.255.252
 mpls traffic-eng tunnels
 pos ais-shut
 pos report lrdi
 ip rsvp bandwidth 155000 155000
!
```

```
interface POS10/1/0
 description xxxx POS0/3
 ip address 10.1.0.13 255.255.255.252
 mpls traffic-eng tunnels
 ip rsvp bandwidth 155000 155000
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
```

PE2 Configuration

```
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback1 force
!
interface Loopback1
 ip address 10.0.0.4 255.255.255.255
!
interface loopback 2
ip address 10.4.0.1 255.255.255.255
!
interface Tunnel27
 ip unnumbered Loopback1
 tunnel destination 10.0.0.27
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name xxxx-1
!
interface FastEthernet0/0/0.2
 encapsulation dot1Q 203
 xconnect 10.0.0.27 2 encapsulation mpls
!
interface FastEthernet0/0/0.3
 encapsulation dot1Q 204
 xconnect 10.0.0.27 4 encapsulation mpls
!
interface FastEthernet1/1/0
 ip address 10.4.1.1 255.255.255.0
 mpls traffic-eng tunnels
 ip rsvp bandwidth 10000 10000
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
!
ip explicit-path name xxxx-1 enable
 next-address 10.4.1.2
 next-address 10.1.0.10
```

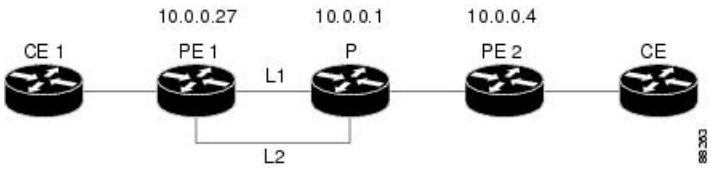
Example: Ethernet over MPLS with MPLS Traffic Engineering Fast Reroute Using Commands Associated with L2VPN Protocol-Based Feature

The following configuration example and the figure show the configuration of Ethernet over MPLS with fast reroute on AToM PE routers.

Routers PE1 and PE2 have the following characteristics:

- A TE tunnel called Tunnel41 is configured between PE1and PE2, using an explicit path through a link called L1. AToM VCs are configured to travel through the FRR-protected tunnel Tunnel41.
- The link L1 is protected by FRR, the backup tunnel is Tunnel1.
- PE2 is configured to forward the AToM traffic back to PE1 through the L2 link.

Figure 54: Fast Reroute Configuration



PE1 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback1 force
!
template type pseudowire T41
 encapsulation mpls
 preferred-path interface Tunnel41 disable-fallback
!
template type pseudowire IP1
 encapsulation mpls
 preferred-path peer 10.4.0.1 disable-fallback
!
interface Loopback1
 ip address 10.0.0.27 255.255.255.255
!
interface Tunnel1
 ip unnumbered Loopback1
 tunnel destination 10.0.0.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 10000
 tunnel mpls traffic-eng path-option 1 explicit name FRR
!
interface Tunnel41
 ip unnumbered Loopback1
 tunnel destination 10.0.0.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name name-1
 tunnel mpls traffic-eng fast-reroute
!
interface POS0/0/0
 description pe1name POS8/0/0
 ip address 10.1.0.2 255.255.255.252
 mpls traffic-eng tunnels
 mpls traffic-eng backup-path Tunnel1
 crc 16
 clock source internal
 pos ais-shut
 pos report lrdi
 ip rsvp bandwidth 155000 155000
```

```
!
interface POS0/3/0
 description pe1name POS10/1/0
 ip address 10.1.0.14 255.255.255.252
 mpls traffic-eng tunnels
 crc 16
 clock source internal
 ip rsvp bandwidth 155000 155000
!
interface gigabitethernet3/0/0.1
 encapsulation dot1Q 203
 interface pseudowire 100
 source template type pseudowire T41
 neighbor 10.0.0.4 2
!
l2vpn xconnect context con1
!
interface gigabitethernet3/0/0.2
 encapsulation dot1Q 204
 interface pseudowire 100
 source template type pseudowire IP1
 neighbor 10.0.0.4 4
!
l2vpn xconnect context con2
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
!
ip classless
ip route 10.4.0.1 255.255.255.255 Tunnel41
!
ip explicit-path name xxxx-1 enable
 next-address 10.4.1.2
 next-address 10.1.0.10
```

P Configuration

```
ip cef
mpls traffic-eng tunnels
!
interface Loopback1
 ip address 10.0.0.1 255.255.255.255
!
interface FastEthernet1/0/0
 ip address 10.4.1.2 255.255.255.0
 mpls traffic-eng tunnels
 ip rsvp bandwidth 10000 10000
!
interface POS8/0/0
 description xxxx POS0/0
 ip address 10.1.0.1 255.255.255.252
 mpls traffic-eng tunnels
 pos ais-shut
 pos report lrdi
 ip rsvp bandwidth 155000 155000
!
interface POS10/1/0
 description xxxx POS0/3
 ip address 10.1.0.13 255.255.255.252
 mpls traffic-eng tunnels
 ip rsvp bandwidth 155000 155000
```

```
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
```

PE2 Configuration

```
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback1 force
!
interface Loopback1
 ip address 10.0.0.4 255.255.255.255
!
interface loopback 2
ip address 10.4.0.1 255.255.255.255
!
interface Tunnel27
 ip unnumbered Loopback1
 tunnel destination 10.0.0.27
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name xxxx-1
!
interface FastEthernet0/0/0.2
 encapsulation dot1Q 203
 interface pseudowire 100
 encapsulation mpls
 neighbor 10.0.0.1 123
!
l2vpn xconnect context A
 member pseudowire 100
 member gigabitethernet 0/0/0.1
!
interface FastEthernet0/0/0.3
 encapsulation dot1Q 204
 interface pseudowire 100
 encapsulation mpls
 neighbor 10.0.0.1 123
!
l2vpn xconnect context A
 member pseudowire 100
 member gigabitethernet 0/0/0.1
!
interface FastEthernet1/1/0
 ip address 10.4.1.1 255.255.255.0
 mpls traffic-eng tunnels
 ip rsvp bandwidth 10000 10000
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
!
ip explicit-path name xxxx-1 enable
 next-address 10.4.1.2
 next-address 10.1.0.10
```

Example: Configuring OAM Cell Emulation

The following example shows how to enable OAM cell emulation on an ATM PVC:

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
xconnect 10.13.13.13 100 encapsulation mpls
oam-ac emulation-enable
oam-pvc manage
```

The following example shows how to set the rate at which an AIS cell is sent every 30 seconds:

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
xconnect 10.13.13.13 100 encapsulation mpls
oam-ac emulation-enable 30
oam-pvc manage
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0/0
class-int oamclass
pvc 1/200 l2transport
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0/0
pvc 1/200 l2transport
class-vc oamclass
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface. One PVC is configured with OAM cell emulation at an AIS rate of 10. That PVC uses the AIS rate of 10 instead of 30.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
```

```
interface atm1/0/0
class-int oamclass
pvc 1/200 l2transport
oam-ac emulation-enable 10
xconnect 10.13.13.13 100 encapsulation mpls
```

Example: Configuring OAM Cell Emulation using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to enable OAM cell emulation on an ATM PVC:

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
interface pseudowire 100
encapsulation mpls
neighbor 10.0.0.1 123
!
l2vpn xconnect context A
member pseudowire 100
member gigabitethernet 0/0/0.1
!
oam-ac emulation-enable
oam-pvc manage
```

The following example shows how to set the rate at which an AIS cell is sent every 30 seconds:

```
interface ATM 1/0/0
pvc 1/200 l2transport
encapsulation aal5
interface pseudowire 100
encapsulation mpls
neighbor 10.0.0.1 123
!
l2vpn xconnect context A
member pseudowire 100
member gigabitethernet 0/0/0.1
!
oam-ac emulation-enable 30
oam-pvc manage
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0/0
class-int oamclass
pvc 1/200 l2transport
interface pseudowire 100
encapsulation mpls
neighbor 10.0.0.1 123
!
l2vpn xconnect context A
```

```
member pseudowire 100
member gigabitethernet 0/0/0.1
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0/0
pvc 1/200 l2transport
class-vc oamclass
interface pseudowire 100
encapsulation mpls
neighbor 10.0.0.1 123
!
l2vpn xconnect context A
member pseudowire 100
member gigabitethernet 0/0/0.1
```

The following example shows how to configure OAM cell emulation for ATM AAL5 over MPLS in VC class configuration mode. The VC class is then applied to an interface. One PVC is configured with OAM cell emulation at an AIS rate of 10. That PVC uses the AIS rate of 10 instead of 30.

```
enable
configure terminal
vc-class atm oamclass
encapsulation aal5
oam-ac emulation-enable 30
oam-pvc manage
interface atm1/0/0
class-int oamclass
pvc 1/200 l2transport
oam-ac emulation-enable 10
interface pseudowire 100
encapsulation mpls
neighbor 10.0.0.1 123
!
l2vpn xconnect context A
member pseudowire 100
member gigabitethernet 0/0/0.1
```

Example: Configuring ATM Cell Relay over MPLS

The following example shows how to configure ATM cell relay over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm cellrelay
encapsulation aal0
interface atm1/0/0
class-int cellrelay
pvc 1/200 l2transport
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure ATM cell relay over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm cellrelay
encapsulation aal0
interface atm1/0/0
pvc 1/200 l2transport
class-vc cellrelay
xconnect 10.13.13.13 100 encapsulation mpls
```

The following example shows how to configure a pseudowire class to transport single ATM cells over a virtual path:

```
pseudowire-class vp-cell-relay
encapsulation mpls
interface atm 5/0
atm pvp 1 l2transport
xconnect 10.0.0.1 123 pw-class vp-cell-relay
```

Example: Configuring ATM Cell Relay over MPLS using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to configure ATM cell relay over MPLS in VC class configuration mode. The VC class is then applied to an interface.

```
enable
configure terminal
vc-class atm cellrelay
encapsulation aal0
interface atm1/0/0
class-int cellrelay
pvc 1/200 l2transport
interface pseudowire 100
encapsulation mpls
neighbor 10.13.13.13 100
!
l2vpn xconnect context A
member pseudowire 100
member gigabitethernet 0/0/0.1
```

The following example shows how to configure ATM cell relay over MPLS in VC class configuration mode. The VC class is then applied to a PVC.

```
enable
configure terminal
vc-class atm cellrelay
encapsulation aal0
interface atm1/0/0
pvc 1/200 l2transport
class-vc cellrelay
interface pseudowire 100
encapsulation mpls
neighbor 10.13.13.13 100
!
l2vpn xconnect context A
```

```
member pseudowire 100
member gigabitethernet 0/0/0.1
```

The following example shows how to configure a pseudowire class to transport single ATM cells over a virtual path:

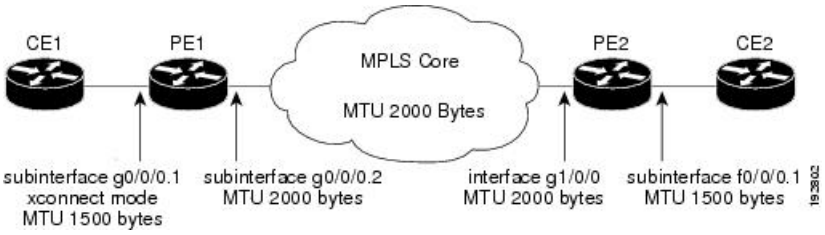
```
template type pseudowire vp-cell-relay
encapsulation mpls
interface atm 5/0
atm pvp 1 l2transport
interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.0.0.1 123
!
l2vpn xconnect context con1
```

Example: Configuring per-Subinterface MTU for Ethernet over MPLS

The figure below shows a configuration that enables matching MTU values between VC endpoints.

As shown in the figure, PE1 is configured in xconnect subinterface configuration mode with an MTU value of 1500 bytes in order to establish an end-to-end VC with PE2, which also has an MTU value of 1500 bytes. If PE1 was not set with an MTU value of 1500 bytes, in xconnect subinterface configuration mode, the subinterface would inherit the MTU value of 2000 bytes set on the interface. This would cause a mismatch in MTU values between the VC endpoints, and the VC would not come up.

Figure 55: Configuring MTU Values in xconnect Subinterface Configuration Mode



The following examples show the router configurations in the figure above:

CE1 Configuration

```
interface gigabitethernet0/0/0
 mtu 1500
 no ip address
!
interface gigabitethernet0/0/0.1
 encapsulation dot1Q 100
 ip address 10.181.182.1 255.255.255.0
```

PE1 Configuration

```
interface gigabitethernet0/0/0
 mtu 2000
 no ip address
!
interface gigabitethernet0/0/0.1
 encapsulation dot1Q 100
```

```
 xconnect 10.1.1.152 100 encapsulation mpls
  mtu 1500
!
interface gigabitethernet0/0/0.2
 encapsulation dot1Q 200
 ip address 10.151.100.1 255.255.255.0
 mpls ip
```

PE2 Configuration

```
interface gigabitethernet1/0/0
 mtu 2000
 no ip address
!
interface gigabitethernet1/0/0.2
 encapsulation dot1Q 200
 ip address 10.100.152.2 255.255.255.0
 mpls ip
!
interface fastethernet0/0/0
 no ip address
!
interface fastethernet0/0/0.1
 description default MTU of 1500 for FastEthernet
 encapsulation dot1Q 100
 xconnect 10.1.1.151 100 encapsulation mpls
```

CE2 Configuration

```
interface fastethernet0/0/0
 no ip address
interface fastethernet0/0/0.1
 encapsulation dot1Q 100
 ip address 10.181.182.2 255.255.255.0
```

The **show mpls l2transport binding** command, issued from router PE1, shows a matching MTU value of 1500 bytes on both the local and remote routers:

```
Router# show mpls l2transport binding
Destination Address: 10.1.1.152,  VC ID: 100
    Local Label: 100
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 202
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
```

```
Router# show mpls l2transport vc detail
Local interface: Gi0/0/0.1 up, line protocol up, Eth VLAN 100 up
  Destination address: 10.1.1.152, VC ID: 100, VC status: up
    Output interface: Gi0/0/0.2, imposed label stack {202}
    Preferred path: not configured
    Default path: active
    Next hop: 10.151.152.2
  Create time: 1d11h, last status change time: 1d11h
  Signaling protocol: LDP, peer 10.1.1.152:0 up
```

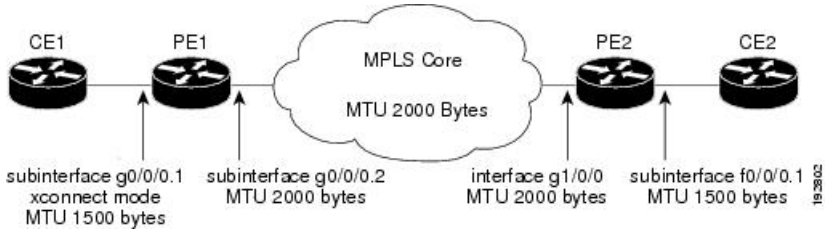
```
    Targeted Hello: 10.1.1.151(LDP Id) -> 10.1.1.152
    MPLS VC labels: local 100, remote 202
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 41, send 39
    byte totals:   receive 4460, send 5346
    packet drops:  receive 0, send 0
```

Example: Configuring per-Subinterface MTU for Ethernet over MPLS using the commands associated with the L2VPN Protocol-Based CLIs feature

The figure below shows a configuration that enables matching MTU values between VC endpoints.

As shown in the figure, PE1 is configured in xconnect subinterface configuration mode with an MTU value of 1500 bytes in order to establish an end-to-end VC with PE2, which also has an MTU value of 1500 bytes. If PE1 was not set with an MTU value of 1500 bytes, in xconnect subinterface configuration mode, the subinterface would inherit the MTU value of 2000 bytes set on the interface. This would cause a mismatch in MTU values between the VC endpoints, and the VC would not come up.

Figure 56: Configuring MTU Values in xconnect Subinterface Configuration Mode



The following examples show the router configurations in the figure above:

CE1 Configuration

```
interface gigabitethernet0/0/0
 mtu 1500
 no ip address
!
interface gigabitethernet0/0/0.1
 encapsulation dot1Q 100
 ip address 10.181.182.1 255.255.255.0
```

PE1 Configuration

```
interface gigabitethernet0/0/0
 mtu 2000
 no ip address
!
interface gigabitethernet0/0/0.1
 encapsulation dot1Q 100
 interface pseudowire 100
encapsulation mpls
neighbor 10.0.0.1 123
mtu 1500
```

```
!
l2vpn xconnect context A
member pseudowire 100
member gigabitethernet 0/0/0.1
!
interface gigabitethernet0/0/0.2
 encapsulation dot1Q 200
 ip address 10.151.100.1 255.255.255.0
 mpls ip
```

PE2 Configuration

```
interface gigabitethernet1/0/0
 mtu 2000
 no ip address
!
interface gigabitethernet1/0/0.2
 encapsulation dot1Q 200
 ip address 10.100.152.2 255.255.255.0
 mpls ip
!
interface fastethernet0/0/0
 no ip address
!
interface fastethernet0/0/0.1
 description default MTU of 1500 for FastEthernet
 encapsulation dot1Q 100
 interface pseudowire 100
 encapsulation mpls
 neighbor 10.0.0.1 123
mtu 1500
!
l2vpn xconnect context A
 member pseudowire 100
 member gigabitethernet 0/0/0.1
```

CE2 Configuration

```
interface fastethernet0/0/0
 no ip address
interface fastethernet0/0/0.1
 encapsulation dot1Q 100
 ip address 10.181.182.2 255.255.255.0
```

The **show l2vpn atom binding** command, issued from router PE1, shows a matching MTU value of 1500 bytes on both the local and remote routers:

```
Device# show l2vpn atom binding
Destination Address: 10.1.1.152,  VC ID: 100
    Local Label: 100
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 202
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
```

Example: Configuring Tunnel Selection

The following example shows how to set up two preferred paths for PE1. One preferred path specifies an MPLS traffic engineering tunnel. The other preferred path specifies an IP address of a loopback address on PE2. There is a static route configured on PE1 that uses a TE tunnel to reach the IP address on PE2.

PE1 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
tag-switching tdp router-id Loopback0
pseudowire-class pw1
 encapsulation mpls
 preferred-path interface Tunnel1 disable-fallback
!
pseudowire-class pw2
 encapsulation mpls
 preferred-path peer 10.18.18.18
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Tunnel1
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.16.16.16
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1500
 tunnel mpls traffic-eng path-option 1 explicit name path-tu1
!
interface Tunnel2
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.16.16.16
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1500
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface gigabitethernet0/0/0
 no ip address
 no ip directed-broadcast
 no negotiation auto
!
interface gigabitethernet0/0/0.1
 encapsulation dot1Q 222
 no ip directed-broadcast
 xconnect 10.16.16.16 101 pw-class pw1
!
interface ATM1/0/0
 no ip address
 no ip directed-broadcast
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
 pvc 0/50 l2transport
  encapsulation aal5
  xconnect 10.16.16.16 150 pw-class pw2
!
```

```
interface FastEthernet2/0/1
 ip address 10.0.0.1 255.255.255.0
 no ip directed-broadcast
 tag-switching ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 15000 15000
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 network 10.2.2.2 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
ip route 10.18.18.18 255.255.255.255 Tunnel2
!
ip explicit-path name path-tu1 enable
 next-address 10.0.0.1
 index 3 next-address 10.0.0.1
```

PE2 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback0
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Loopback2
 ip address 10.18.18.18 255.255.255.255
 no ip directed-broadcast
!
interface FastEthernet1/1/0
 ip address 10.0.0.2 255.255.255.0
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 no cdp enable
 ip rsvp bandwidth 15000 15000
!
interface FastEthernet1/1/1
 no ip address
 no ip directed-broadcast
 no cdp enable
!
interface FastEthernet1/1/1.1
 encapsulation dot1Q 222
 no ip directed-broadcast
 no cdp enable
 mpls l2transport route 10.2.2.2 101
!
interface ATM5/0/0
 no ip address
 no ip directed-broadcast
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
 pvc 0/50 l2transport
  encapsulation aal5
  xconnect 10.2.2.2 150 encapsulation mpls
!
router ospf 1
```

```
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 network 10.16.16.16 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
```

Example: Configuring Tunnel Selection Using Commands Associated with L2VPN Protocol-Based Feature

The following example shows how to set up two preferred paths for PE1. One preferred path specifies an MPLS traffic engineering tunnel. The other preferred path specifies an IP address of a loopback address on PE2. There is a static route configured on PE1 that uses a TE tunnel to reach the IP address on PE2.

PE1 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
tag-switching tdp router-id Loopback0
template type pseudowire pw1
 encapsulation mpls
 preferred-path interface Tunnel1 disable-fallback
!
template type pseudowire pw2
 encapsulation mpls
 preferred-path peer 10.18.18.18
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Tunnel1
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.16.16.16
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1500
 tunnel mpls traffic-eng path-option 1 explicit name path-tu1
!
interface Tunnel2
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.16.16.16
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1500
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface gigabitethernet0/0/0
 no ip address
 no ip directed-broadcast
 no negotiation auto
!
interface gigabitethernet0/0/0.1
 encapsulation dot1Q 222
 no ip directed-broadcast
 interface pseudowire 100
 source template type pseudowire pw1
```

```
  neighbor 10.16.16.16 101
!
l2vpn xconnect context con1
!
interface ATM1/0/0
 no ip address
 no ip directed-broadcast
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
 pvc 0/50 l2transport
  encapsulation aal5
 interface pseudowire 100
 source template type pseudowire pw2
 neighbor 10.16.16.16 150
!
l2vpn xconnect context con1
!
interface FastEthernet2/0/1
 ip address 10.0.0.1 255.255.255.0
 no ip directed-broadcast
 tag-switching ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 15000 15000
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 network 10.2.2.2 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
ip route 10.18.18.18 255.255.255.255 Tunnel2
!
ip explicit-path name path-tu1 enable
 next-address 10.0.0.1
 index 3 next-address 10.0.0.1
```

PE2 Configuration

```
mpls label protocol ldp
mpls traffic-eng tunnels
mpls ldp router-id Loopback0
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface Loopback2
 ip address 10.18.18.18 255.255.255.255
 no ip directed-broadcast
!
interface FastEthernet1/1/0
 ip address 10.0.0.2 255.255.255.0
 no ip directed-broadcast
 mpls traffic-eng tunnels
 mpls ip
 no cdp enable
 ip rsvp bandwidth 15000 15000
!
interface FastEthernet1/1/1
 no ip address
 no ip directed-broadcast
 no cdp enable
```

```
!
interface FastEthernet1/1/1.1
 encapsulation dot1Q 222
 no ip directed-broadcast
 no cdp enable
 mpls l2transport route 10.2.2.2 101
!
interface ATM5/0/0
 no ip address
 no ip directed-broadcast
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
 pvc 0/50 l2transport
  encapsulation aal5
  interface pseudowire 100
   encapsulation mpls
   neighbor 10.2.2.2 150
!
l2vpn xconnect context A
  member pseudowire 100
  member GigabitEthernet0/0/0.1
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 network 10.16.16.16 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
```

Example: Configuring MTU Values in xconnect Configuration Mode for L2VPN Interworking

The following example shows an L2VPN Interworking example. The PE1 router has a serial interface configured with an MTU value of 1492 bytes. The PE2 router uses xconnect configuration mode to set a matching MTU of 1492 bytes, which allows the two routers to form an interworking VC. If the PE2 router did not set the MTU value in xconnect configuration mode, the interface would be set to 1500 bytes by default and the VC would not come up.

Note L2VPN interworking is not supported on Cisco ASR 900 RSP3 Module.

PE1 Configuration

```
pseudowire-class atom-ipiw
 encapsulation mpls
 interworking ip
!
interface Loopback0
 ip address 10.1.1.151 255.255.255.255
!
interface Serial2/0/0
 mtu 1492
 no ip address
 encapsulation ppp
 no fair-queue
 serial restart-delay 0
```

```
 xconnect 10.1.1.152 123 pw-class atom-ipiw
!
interface Serial4/0/0
 ip address 10.151.100.1 255.255.255.252
 encapsulation ppp
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.151 0.0.0.0 area 0
 network 10.151.100.0 0.0.0.3 area 0
!
mpls ldp router-id Loopback0
```

PE2 Configuration

```
pseudowire-class atom-ipiw
 encapsulation mpls
 interworking ip
!
interface Loopback0
 ip address 10.1.1.152 255.255.255.255
!
interface FastEthernet0/0/0
 no ip address
 xconnect 10.1.1.151 123 pw-class atom-ipiw
  mtu 1492
!
interface Serial4/0/0
 ip address 10.100.152.2 255.255.255.252
 encapsulation ppp
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.152 0.0.0.0 area 0
 network 10.100.152.0 0.0.0.3 area 0
!
mpls ldp router-id Loopback0
```

The **show mpls l2transport binding** command shows that the MTU value for the local and remote routers is 1492 bytes.

PE1

```
Router# show mpls l2transport binding
Destination Address: 10.1.1.152,  VC ID: 123
    Local Label: 105
        Cbit: 1,    VC Type: PPP,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 205
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
Router# show mpls l2transport vc detail
Local interface: Serial2/0/0 up, line protocol up, PPP up
```

```
  MPLS VC type is PPP, interworking type is IP
  Destination address: 10.1.1.152, VC ID: 123, VC status: up
    Output interface: Serial4/0/0, imposed label stack {1003 205}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:25:29, last status change time: 00:24:54
  Signaling protocol: LDP, peer 10.1.1.152:0 up
    Targeted Hello: 10.1.1.151(LDP Id) -> 10.1.1.152
    Status TLV support (local/remote)   : enabled/supported
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: no fault
      Last local SSS circuit status rcvd: no fault
      Last local SSS circuit status sent: no fault
      Last local  LDP TLV    status sent: no fault
      Last remote LDP TLV    status rcvd: no fault
    MPLS VC labels: local 105, remote 205
    Group ID: local n/a, remote 0
    MTU: local 1492, remote 1492
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 30, send 29
    byte totals:   receive 2946, send 3364
    packet drops:  receive 0, send 0
```

PE2

```
Router# show mpls l2transport binding
Destination Address: 10.1.1.151,  VC ID: 123
    Local Label: 205
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
    Remote Label: 105
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
Router# show mpls l2transport vc detail
Local interface: Fe0/0/0 up, line protocol up, FastEthernet up
  MPLS VC type is FastEthernet, interworking type is IP
  Destination address: 10.1.1.151, VC ID: 123, VC status: up
    Output interface: Se4/0/0, imposed label stack {1002 105}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:25:19, last status change time: 00:25:19
  Signaling protocol: LDP, peer 10.1.1.151:0 up
    Targeted Hello: 10.1.1.152(LDP Id) -> 10.1.1.151
    Status TLV support (local/remote)   : enabled/supported
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: no fault
      Last local SSS circuit status rcvd: no fault
      Last local SSS circuit status sent: no fault
      Last local  LDP TLV    status sent: no fault
      Last remote LDP TLV    status rcvd: no fault
    MPLS VC labels: local 205, remote 105
    Group ID: local n/a, remote 0
    MTU: local 1492, remote 1492
    Remote interface description:
  Sequencing: receive disabled, send disabled
```

```
  VC statistics:
    packet totals: receive 29, send 30
    byte totals:   receive 2900, send 3426
    packet drops:  receive 0, send 0
```

Example: Configuring MTU Values in xconnect Configuration Mode for L2VPN Interworking Using Commands Associated with L2VPN Protocol-Based Feature

The following example shows an L2VPN Interworking example. The PE1 router has a serial interface configured with an MTU value of 1492 bytes. The PE2 router uses xconnect configuration mode to set a matching MTU of 1492 bytes, which allows the two routers to form an interworking VC. If the PE2 router did not set the MTU value in xconnect configuration mode, the interface would be set to 1500 bytes by default and the VC would not come up.

PE1 Configuration

```
template type pseudowire atom-ipiw
 encapsulation mpls
 interworking ip
!
interface Loopback0
 ip address 10.1.1.151 255.255.255.255
!
interface Serial2/0/0
 mtu 1492
 no ip address
 encapsulation ppp
 no fair-queue
 serial restart-delay 0
interface pseudowire 100
 source template type pseudowire atom-ipiw
 neighbor 10.1.1.152 123
!
l2vpn xconnect context con1
 member <ac_int>
 member pseudowire 100
!
interface Serial4/0/0
 ip address 10.151.100.1 255.255.255.252
 encapsulation ppp
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.151 0.0.0.0 area 0
 network 10.151.100.0 0.0.0.3 area 0
!
mpls ldp router-id Loopback0
```

PE2 Configuration

```
template type pseudowire atom-ipiw
 encapsulation mpls
 interworking ip
!
interface Loopback0
 ip address 10.1.1.152 255.255.255.255
```

```
!
interface FastEthernet0/0/0
 no ip address
 interface pseudowire 100
 source template type pseudowire atom-ipiw
 neighbor 10.1.1.151 123
!
l2vpn xconnect context con1
 member <ac_int>
 member pseudowire1
!
interface Serial4/0/0
 ip address 10.100.152.2 255.255.255.252
 encapsulation ppp
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.152 0.0.0.0 area 0
 network 10.100.152.0 0.0.0.3 area 0
!
mpls ldp router-id Loopback0
```

The **show l2vpn atom binding** command shows that the MTU value for the local and remote routers is 1492 bytes.

PE1

```
Device# show l2vpn atom binding
Destination Address: 10.1.1.152,  VC ID: 123
    Local Label: 105
        Cbit: 1,    VC Type: PPP,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 205
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
Device# show l2vpn atom vc detail
Local interface: Serial2/0/0 up, line protocol up, PPP up
  MPLS VC type is PPP, interworking type is IP
  Destination address: 10.1.1.152, VC ID: 123, VC status: up
    Output interface: Serial4/0/0, imposed label stack {1003 205}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:25:29, last status change time: 00:24:54
  Signaling protocol: LDP, peer 10.1.1.152:0 up
    Targeted Hello: 10.1.1.151(LDP Id) -> 10.1.1.152
    Status TLV support (local/remote)   : enabled/supported
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: no fault
      Last local SSS circuit status rcvd: no fault
      Last local SSS circuit status sent: no fault
      Last local  LDP TLV    status sent: no fault
      Last remote LDP TLV    status rcvd: no fault
    MPLS VC labels: local 105, remote 205
    Group ID: local n/a, remote 0
    MTU: local 1492, remote 1492
    Remote interface description:
```

```
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 30, send 29
    byte totals:   receive 2946, send 3364
    packet drops:  receive 0, send 0
```

PE2

```
Device# show l2vpn atom binding
Destination Address: 10.1.1.151,  VC ID: 123
    Local Label: 205
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: RA [2]
              CV Type: LSPV [2]
    Remote Label: 105
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1492,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
Device# show l2vpn atom vc detail
Local interface: Fe0/0/0 up, line protocol up, FastEthernet up
  MPLS VC type is FastEthernet, interworking type is IP
  Destination address: 10.1.1.151, VC ID: 123, VC status: up
    Output interface: Se4/0/0, imposed label stack {1002 105}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:25:19, last status change time: 00:25:19
  Signaling protocol: LDP, peer 10.1.1.151:0 up
    Targeted Hello: 10.1.1.152(LDP Id) -> 10.1.1.151
    Status TLV support (local/remote)   : enabled/supported
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: no fault
      Last local SSS circuit status rcvd: no fault
      Last local SSS circuit status sent: no fault
      Last local  LDP TLV    status sent: no fault
      Last remote LDP TLV    status rcvd: no fault
    MPLS VC labels: local 205, remote 105
    Group ID: local n/a, remote 0
    MTU: local 1492, remote 1492
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 29, send 30
    byte totals:   receive 2900, send 3426
    packet drops:  receive 0, send 0
```

Examples: Configuring Any Transport over MPLS (AToM) Remote Ethernet Port Shutdown

The following example shows how to enable remote Ethernet port shutdown:

```
configure terminal
!
pseudowire-class eompls
 encapsulation mpls
!
interface GigabitEthernet1/0/0
```

```
 xconnect 10.1.1.1 1 pw-class eompls
  remote link failure notification
```

The following example shows how to disable remote Ethernet port shutdown:

```
configure terminal
!
pseudowire-class eompls
 encapsulation mpls
!
interface GigabitEthernet1/0/0
 xconnect 10.1.1.1 1 pw-class eompls
  no remote link failure notification
```

The related **show** command output reports operational status for all remote L2 Tunnels by interface.

```
Router# show interface G1/0/0
GigabitEthernet1/0/0 is L2 Tunnel remote down, line protocol is up
Hardware is GigMac 4 Port GigabitEthernet, address is 0003.ff4e.12a8 (bia 0003.ff4e.12a8)
  Internet address is 10.9.9.2/16
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec, rely 255/255, load 1/255
Router# show ip interface brief
Interface              IP-Address      OK? Method Status Protocol
GigabitEthernet2/0/0 unassigned      YES NVRAM  L2 Tunnel remote down up
GigabitEthernet2/1/0 unassigned      YES NVRAM  administratively down down
```

Note Remote Ethernet port shutdown is enabled by default when EVC "default encapsulation" is configured.

Examples: Configuring Any Transport over MPLS (AToM) Remote Ethernet Port Shutdown Using Commands Associated with L2VPN Protocol-Based Feature

The following example shows how to enable remote Ethernet port shutdown:

```
configure terminal
!
template type pseudowire eompls
 encapsulation mpls
!
interface GigabitEthernet1/0/0
interface pseudowire 100
 source template type pseudowire eompls
 neighbor 10.1.1.1 1
!
l2vpn xconnect context con1
 remote link failure notification
```

The following example shows how to disable remote Ethernet port shutdown:

```
configure terminal
!
template type pseudowire eompls
 encapsulation mpls
!
interface GigabitEthernet1/0/0
 interface pseudowire 100
 source template type pseudowire eompls
```

```
 neighbor 10.1.1.1 1
!
l2vpn xconnect context con1
 no remote link failure notification
```

The related **show** command output reports operational status for all remote L2 Tunnels by interface.

```
Router# show interface G1/0/0
GigabitEthernet1/0/0 is L2 Tunnel remote down, line protocol is up
Hardware is GigMac 4 Port GigabitEthernet, address is 0003.ff4e.12a8 (bia 0003.ff4e.12a8)
  Internet address is 10.9.9.2/16
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec, rely 255/255, load 1/255
Router# show ip interface brief
Interface              IP-Address      OK? Method Status Protocol
GigabitEthernet2/0/0 unassigned      YES NVRAM  L2 Tunnel remote down up
GigabitEthernet2/1/0 unassigned      YES NVRAM  administratively down down
```

Additional References for Any Transport over MPLS

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Any Transport over MPLS

ß

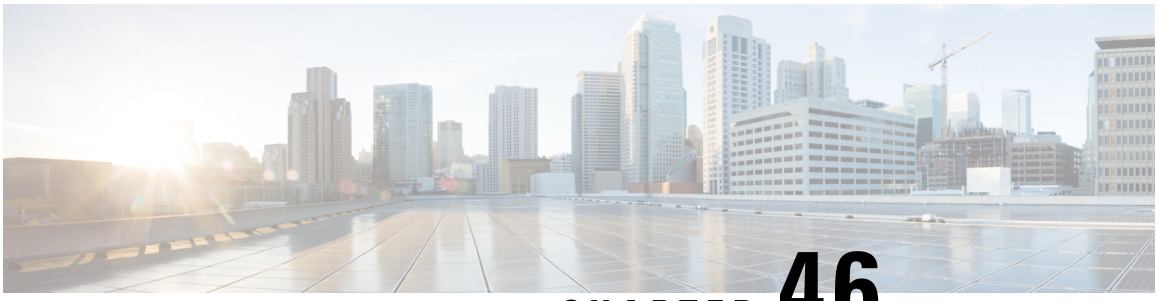
Table 104: Feature Information for Any Transport over MPLS

Feature Name	Releases	Feature Information
Any Transport over MPLS (AToM): ATM AAL5 over MPLS (AAL5oMPLS)	Cisco IOS XE Release 3.2S Cisco IOS XE Release 3.6S	In Cisco IOS XE Release 3.2S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE Release 3.6S, support was added for the Cisco ASR 903 Router. This feature introduced no new or modified commands.
Any Transport over MPLS (AToM): ATM Cell Relay over MPLS: Packed Cell Relay	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
Any Transport over MPLS (AToM): ATM OAM Emulation	Cisco IOS XE Release 3.2S	In Cisco IOS XE Release 3.2S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. This feature introduced no new or modified commands.
	Cisco IOS XE Release 2.5	This feature provides capability to support sequencing of AToM) data plane packets.
Any Transport over MPLS (AToM): Ethernet over MPLS (EoMPLS)	Cisco IOS XE Release 2.4 Cisco IOS XE Release 3.5S	This feature allows you to transport Layer 2 Ethernet VLAN packets from various sources over an MPLS backbone. Ethernet over MPLS extends the usability of the MPLS backbone by enabling it to offer Layer 2 services in addition to already existing Layer 3 services. You can enable the MPLS backbone network to accept Layer 2 VLAN packets by configuring the PE routers at the both ends of the MPLS backbone. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Routers. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
Any Transport over MPLS (AToM): Ethernet over MPLS: Port Mode (EoMPLS)	Cisco IOS XE Release 2.4	Ethernet over MPLS (EoMPLS) is the transport of Ethernet frames across an MPLS core. It transports all frames received on a particular Ethernet or virtual LAN (VLAN) segment, regardless of the destination Media Access Control (MAC) information. It does not perform MAC learning or MAC look up for forwarding packets from the Ethernet interface. Port mode allows a frame coming into an interface to be packed into an MPLS packet and transported over the MPLS backbone to an egress interface. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Routers.

Feature Name	Releases	Feature Information
Any Transport over MPLS-Ethernet over MPLS Enhancements: Fast Reroute	Cisco IOS XE Release 2.4	AToM can use MPLS traffic engineering (TE) tunnels with fast reroute (FRR) support. This features enhances FRR functionality for Ethernet over MPLS (EoMPLS). In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Routers.
Any Transport over MPLS (AToM): Frame Relay over MPLS (FRoMPLS)	Cisco IOS XE Release 3.2.1S	In Cisco IOS XE Release 3.2.1S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. This feature introduced no new or modified commands.
Any Transport over MPLS (AToM): HDLC over MPLS (HDLCoMPLS)	Cisco IOS XE Release 3.2S	In Cisco IOS XE Release 3.2S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. This feature introduced no new or modified commands.
Any Transport over MPLS (AToM): Layer 2 Quality of Service (QoS)	Cisco IOS XE Release 2.3	This feature provides support for quality of service (QoS) features such as traffic policing, traffic shaping, packet marking, and mapping of the packets. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Routers.
Any Transport over MPLS (AToM): PPP over MPLS (PPPoMPLS)	Cisco IOS XE Release 3.2S	In Cisco IOS XE Release 3.2S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. This feature introduced no new or modified commands.
Any Transport over MPLS (AToM): Remote Ethernet Port Shutdown	Cisco IOS XE Release 2.4	This feature allows a service provider edge (PE) router on the local end of an Ethernet over MPLS (EoMPLS) pseudowire to detect a remote link failure and cause the shutdown of the Ethernet port on the local customer edge (CE) router. Because the Ethernet port on the local CE router is shut down, the router does not lose data by continuously sending traffic to the failed remote link. This is beneficial if the link is configured as a static IP route. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Routers.
ATM Port Mode Packed Cell Relay over MPLS	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
ATM VC Class Support	Cisco IOS XE Release 2.3	The ATM VC Class Support feature allows you to specify AAL5 and AAL0 encapsulations as part of a VC class. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Routers.

Feature Name	Releases	Feature Information
AToM Tunnel Selection	Cisco IOS XE Release 2.3	The AToM Tunnel Selection feature allows you to specify the path that traffic uses. You can specify either an MPLS TE tunnel or destination IP address or domain name server (DNS) name. You also have the option of specifying whether the VCs should use the default path (the path LDP uses for signaling) if the preferred path is unreachable. This option is enabled by default; you must explicitly disable it. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
AToM: ATM Cell Relay over MPLS: VP Mode	Cisco IOS XE Release 2.3	The AToM: ATM Cell Relay over MPLS: VP Mode feature allows you to insert one ATM cell in each MPLS packet in VP mode. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Routers.
AToM: Single Cell Relay-VC Mode	Cisco IOS XE Release 2.3	The AToM Single Cell Relay-VC Mode feature allows you to insert one ATM cell in each MPLS packet in VC mode. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Routers.
MPLS MTU Command for GRE Tunnels	Cisco IOS XE Release 2.6	This feature allows you to set the MPLS MTU size in GRE tunnels to the maximum size besides the current default size. The following command was modified for this release: **mpls mtu**.
MPLS L2VPN Clear Xconnect Command	Cisco IOS XE Release 3.1S	These features enable you to: • Reset a VC associated with an interface, a peer address, or on all the configured xconnect circuit attachments • Set the control word on dynamic pseudowires (L2VPN pseudowire control word configuration • Enable ATM cell packing for static pseudowires. The following commands were introduced or modified by these features: **cell-packing**, **clear xconnect**, **control-word**, **encapsulation**(Any Transport over MPLS), **oam-ac emulation-enable**.

Feature Name	Releases	Feature Information
Per-Subinterface MTU for Ethernet over MPLS (EoMPLS)	Cisco IOS XE Release 2.4	This feature provides you with the ability to specify maximum transmission unit (MTU) values in xconnect subinterface configuration mode. When you use xconnect subinterface configuration mode to set the MTU value, you establish a pseudowire connection for situations where the interfaces have different MTU values that cannot be changed. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. No commands were new or modified for this release.
VLAN ID Rewrite	Cisco IOS XE Release 2.4	The VLAN ID rewrite feature enables you to use VLAN interfaces with different VLAN IDs at both ends of the tunnel. In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Routers.
AToM Load Balancing with Single PW	Cisco IOS XE Release 3.4S	The AToM Load Balancing with Single PW feature enables load balancing for packets within the same pseudowire by further classifying packets within the same pseudowire into different flows based on some field in the packet received on attachment circuit. In Cisco IOS XE Release 3.4S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
Flow-Aware Transport of MPLS Pseudowires	Cisco IOS XE Release 3.11S	The Flow-Aware Transport of MPLS Pseudowires feature enables load balancing of packets within the same pseudowire by further classifying the packets into different flows by adding a flow label at the bottom of the MPLS label stack.
EoMPLS over IPv6 GRE Tunnel	Cisco IOS XE Release 3.15S	The EoMPLS over IPv6 GRE Tunnel feature supports tunneling of EoMPLS traffic via an IPv6 network by using GRE tunnels.



CHAPTER 46

L2VPN Interworking

Interworking is a transforming function that is required to interconnect two heterogeneous attachment circuits (ACs). Several types of interworking functions exist. The function that is used would depend on the type of ACs being used, the type of data being carried, and the level of functionality required. The two main Layer 2 Virtual Private Network (L2VPN) interworking functions supported in Cisco IOS XE software are bridged and routed interworking.

Layer 2 (L2) transport over multiprotocol label switching (MPLS) and IP already exists for like-to-like ACs, such as Ethernet-to-Ethernet or Point-to-Point Protocol (PPP)-to-PPP. L2VPN Interworking builds on this functionality by allowing disparate ACs to be connected. An interworking function facilitates the translation between different L2 encapsulations.

Prerequisites for L2VPN Interworking

Before you configure L2VPN interworking on a device you must enable Cisco Express Forwarding.

HDLC-to-Ethernet Interworking

- Ensure that the serial controller and interface on the High-Level Data Link Control (HDLC) customer edge (CE) and provider edge (PE) devices are configured.

```
enable
 configure terminal
  controller e1 2/0
   channel-group 0 timeslots 1
   no shutdown
!
interface Serial 2/0:0
 no shutdown
end
```

- Before configuring HDLC-to-Ethernet bridged interworking, ensure that bridging is configured on the HDLC CE device.

```
enable
 configure terminal
  bridge irb
  bridge 1 protocol ieee
  bridge 1 route ip
!
interface Serial 2/0:0
 no bridge-group 1
 no ip address
!
interface BVI1
 no ip address
 ip address 192.0.2.1 255.255.255.0
 no shutdown
!
interface Serial 2/0:0
 no ip address
 encapsulation hdlc
 bridge-group 1
  no shutdown
end
```

- Before configuring HDLC-to-Ethernet routed interworking, ensure that an IP address is configured on the HDLC CE device.

```
interface Serial 2/0:0
 ip address 192.0.2.1 255.255.255.0
 encapsulation hdlc
 no shutdown
end
```

Restrictions for L2VPN Interworking

General Restrictions for L2VPN Interworking

This section lists general restrictions that apply to L2VPN interworking. Other restrictions that are platform-specific or device-specific are listed in the following sections.

- MTU configured on the AC should not exceed the MTU in the core of the network because fragmentation is not supported.
- The interworking type on one provider edge (PE) router must match the interworking type on the peer PE router.
- IP interworking with native VLANs is not supported.
- Ethernet VLAN (Type 4) interworking is not supported.
- Only the following Quality of Service (QoS) features are supported with L2VPN interworking:
  - Static IP type of service (ToS) or MPLS experimental bit (EXP) setting in tunnel header.
  - One-to-one mapping of VLAN priority bits to MPLS EXP bits.

- VRF-aware Layer 2 Tunneling Protocol Version 3 (L2TPv3) is not supported on Cisco ASR 1000 platforms.

Restrictions for Routed Interworking

Routed interworking has the following restrictions:

- Multipoint Frame Relay (FR) is not supported.
- QoS classification on IP ToS, DSCP and other IP header fields is not supported.
- Security access control list (ACL) and other features based on IP header fields parsing are not supported.
- In routed mode, only one customer edge (CE) router can be attached to an Ethernet PE router.
- There must be a one-to-one relationship between an AC and the pseudowire. Point-to-multipoint or multipoint-to-point configurations are not supported.
- You must configure routing protocols for point-to-point operation on the CE routers when configuring an Ethernet to non-Ethernet setup.
- In the IP interworking mode, the IPv4 (0800) translation is supported. The PE router captures Address Resolution Protocol (ARP) (0806) packets and responds with its own MAC address (proxy ARP). Everything else is dropped.
- The Ethernet must contain only two IP devices: PE router and CE router. The PE router performs proxy ARP and responds to all ARP requests it receives. Therefore, only one CE router and one PE router should be on the Ethernet segment.
- If the CE routers are doing static routing, you can perform the following tasks:
  - The PE router needs to learn the MAC address of the CE router to correctly forward traffic to it. The Ethernet PE router sends an Internet Control Message Protocol (ICMP) Router Discovery Protocol (RDP) solicitation message with the source IP address as zero. The Ethernet CE router responds to this solicitation message. To configure the Cisco CE router's Ethernet interface to respond to the ICMP RDP solicitation message, issue the **ip irdp** command in interface configuration mode. If you do not configure the CE router, traffic is dropped until the CE router sends traffic toward the PE router.
  - To disable the CE routers from running the router discovery protocol, issue the **ip irdp maxadvertinterval 0** command in interface configuration mode.
- When you change the interworking configuration on an Ethernet PE router, clear the ARP entry on the adjacent CE router so that it can learn the new MAC address. Otherwise, you might experience traffic drops.

Restrictions for PPP Interworking

The following restrictions apply to PPP interworking:

- There must be a one-to-one relationship between a PPP session and the pseudowire. Multiplexing of multiple PPP sessions over the pseudowire is not supported.
- Only IP (IPv4 (0021) interworking is supported. Link Control Protocol (LCP) packets and Internet Protocol Control Protocol (IPCP) packets are terminated at the PE router. Everything else is dropped.

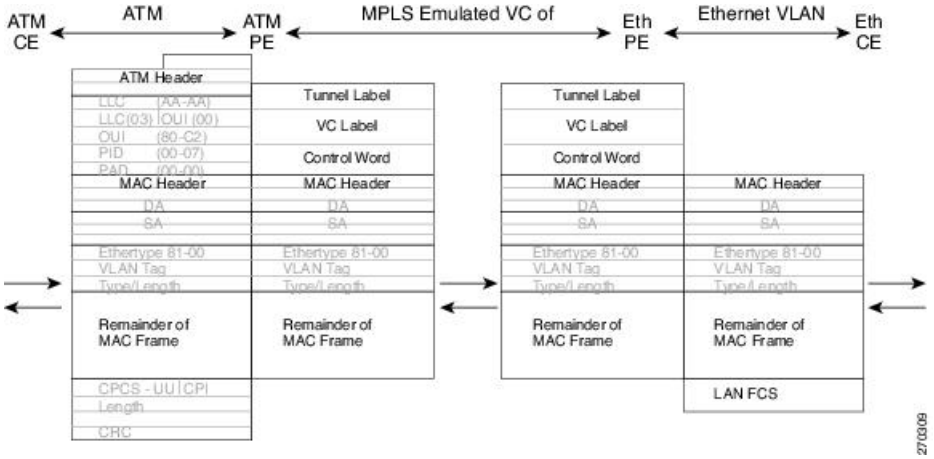
- By default, the PE router assumes that the CE router knows the remote CE router's IP address.
- Password Authentication Protocol (PAP) and Challenge-Handshake Authentication Protocol (CHAP) authentication are supported.

Restrictions for Ethernet/VLAN-to-ATM AAL5 Interworking

The Ethernet/VLAN to ATM AAL5 Any Transport over MPLS (AToM) has the following restrictions:

- Only the following translations are supported; other translations are dropped:
  - Ethernet without LAN FCS (AAAA030080C200070000)
  - Spanning tree (AAAA030080C2000E)
- The ATM encapsulation type supported for bridged interworking is aal5snap. However, ATM encapsulation types supported for routed interworking are aal5snap and aal5mux.
- The existing QoS functionality for ATM is supported, including setting the ATM CLP bit.
- Only ATM AAL5 VC mode is supported. ATM VP and port mode are not supported.
- SVCs are not supported.
- Individual AAL5 ATM cells are assembled into frames before being sent across the pseudowire.
- Non-AAL5 traffic, (such as Operation, Administration, and Maintenance (OAM) cells) is punted to be processed at the route processor (RP) level. A VC that has been configured with OAM cell emulation on the ATM PE router (using the **oam-ac emulation-enable** CLI command) can send end-to-end F5 loopback cells at configured intervals toward the CE router.
- When the pseudowire is down, an F5 end-to-end segment alarm indication signal/remote defect indication (AIS/RDI) is sent from the PE router to the CE router.
- If the Ethernet frame arriving from the Ethernet CE router includes a 802.1Q header (VLAN header), due to the type of endpoint attachment (Ethernet port mode), the VLAN header stays in the frame across the pseudowire (see the figure below).

Figure 57: Protocol Stack for ATM-to-Ethernet AToM Bridged Interworking--with VLAN Header

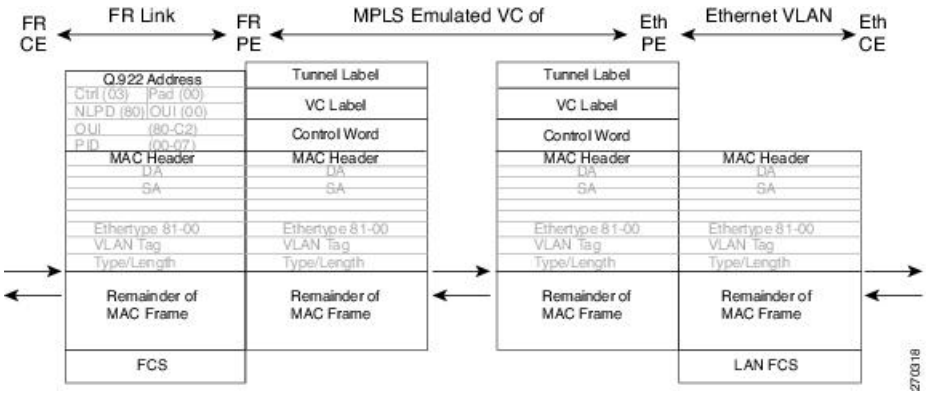


Restrictions for Ethernet/VLAN-to-Frame Relay Interworking

The Ethernet/VLAN-to-Frame Relay AToM has the following restrictions:

- Only the following translations are supported; other translations are dropped:
  - Ethernet without LAN FCS (0300800080C20007)
  - Spanning tree (0300800080C2000E)
- The PE router automatically supports translation of both Cisco and IETF Frame Relay encapsulation types coming from the CE router, but translates only to IETF when sending to the CE router. This is not a problem for the Cisco CE router, because it can manage IETF encapsulation upon receipt even if it is configured to send a Cisco encapsulation.
- The PVC status signaling works the same way as in the like-to-like case. The PE router reports the PVC status to the CE router based upon the availability of the pseudowire.
- The AC maximum transmission unit (MTU) must be within the supported range of MTUs when connected over MPLS.
- Only Frame Relay DLCI mode is supported. Frame Relay port mode is not supported.
- If the Ethernet frame includes a 802.1Q header (VLAN header), due to the type of endpoint attachment (Ethernet port mode), the VLAN header stays in the frame across the pseudowire (see the figure below).
- Frame Relay encapsulation types supported for routed interworking are Cisco and IETF for incoming traffic. However, IETF is also supported for outgoing traffic traveling to the CE router.

Figure 58: Protocol Stack for Frame Relay-to-Ethernet AToM Bridged Interworking--with VLAN Header



Restrictions for HDLC-to-Ethernet Interworking

- The “none CISCO” High-Level Data Link Control (HDLC) encapsulation is not supported.
- IPv6 is not supported in routed mode.

Information About L2VPN Interworking

Overview of L2VPN Interworking

L2 transport over MPLS and IP already exists for like-to-like ACs, such as Ethernet-to-Ethernet or PPP-to-PPP. L2VPN Interworking builds on this functionality by allowing disparate ACs to be connected. An interworking function facilitates the translation between the different L2 encapsulations.

Only the following interworking combinations are supported:

- ATM-to-Ethernet - Routed interworking
- ATM-to-Ethernet - Bridged interworking
- Frame relay-to-Ethernet - Bridged interworking
- PPP-to-Ethernet - Routed interworking
- HDLC-to-Ethernet - Bridged and Routed interworking

L2VPN Interworking Modes

L2VPN interworking works in either Ethernet (bridged) mode or IP (routed) mode. L2VPN interworking does not support Ethernet VLAN (Type 4) mode. You specify the mode in the following ways:

- If using the older legacy CLI commands, you can use the **interworking** {**ethernet** | **ip**} command in pseudowire-class configuration mode.
- If using the newer L2VPN protocol-based CLI commands, you can use the **interworking** {**ethernet** | **ip**} command in xconnect configuration mode.

The **interworking** command causes the ACs to be terminated locally. The two keywords perform the following functions:

- The **ethernet** keyword causes Ethernet frames to be extracted from the AC and sent over the pseudowire. Ethernet end-to-end transmission is resumed. AC frames that are not Ethernet are dropped. In the case of VLAN, the VLAN tag is removed, leaving an untagged Ethernet frame.
- The **ip** keyword causes IP packets to be extracted from the AC and sent over the pseudowire. AC frames that do not contain IPv4 packets are dropped.

The following sections explain more about Ethernet and IP interworking modes.

Ethernet or Bridged Interworking

Ethernet interworking is also called bridged interworking. Ethernet frames are bridged across the pseudowire. The CE routers could be natively bridging Ethernet or could be routing using a bridged encapsulation model, such as Bridge Virtual Interface (BVI) or Routed Bridge Encapsulation (RBE). The PE routers operate in Ethernet like-to-like mode.

This mode is used to offer the following services:

- LAN services--An example is an enterprise that has several sites, where some sites have Ethernet connectivity to the service provider (SP) network and others have ATM connectivity. If the enterprise wants LAN connectivity to all its sites, traffic from the Ethernet or VLAN of one site can be sent through the IP/MPLS network and encapsulated as bridged traffic over an ATM VC of another site.
- Connectivity services--An example is an enterprise that has different sites that are running an Internal Gateway Protocol (IGP) routing protocol, which has incompatible procedures on broadcast and nonbroadcast links. The enterprise has several sites that are running an IGP, such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS), between the sites. In this scenario, some of the procedures (such as route advertisement or designated router) depend on the underlying L2 protocol and are different for a point-to-point ATM connection versus a broadcast Ethernet connection. Therefore, the bridged encapsulation over ATM can be used to achieve homogenous Ethernet connectivity between the CE routers running the IGP.

IP or Routed Interworking

IP interworking is also called routed interworking. The CE routers encapsulate the IP on the link between the CE router and PE router. A new VC type is used to signal the IP pseudowire in MPLS. Translation between the L2 and IP encapsulations across the pseudowire is required. Special consideration needs to be given to the address resolution and routing protocol operation, because these are handled differently on different L2 encapsulations.

This mode is used to provide IP connectivity between sites, regardless of the L2 connectivity to these sites. It is different from a Layer 3 VPN because it is point-to-point in nature and the service provider does not maintain any customer routing information.

Address resolution is encapsulation dependent:

- Ethernet uses Address Resolution Protocol (ARP)
- ATM uses inverse ARP
- PPP uses IP Control Protocol (IPCP)
- HDLC uses Serial Line ARP (SLARP)

Therefore, address resolution must be terminated on the PE router. End-to-end address resolution is not supported. Routing protocols operate differently over broadcast and point-to-point media. For Ethernet, the CE routers must either use static routing or configure the routing protocols to treat the Ethernet side as a point-to-point network.

In routed interworking, IP packets that are extracted from the ACs are sent over the pseudowire. The pseudowire works in the IP Layer 2 transport (VC type 0x000B) like-to-like mode. The interworking function at network service provider's (NSP) end performs the required adaptation based on the AC technology. Non-IPv4 packets are dropped.

In routed interworking, the following considerations are to be kept in mind:

- Address resolution packets (ARP), inverse ARP, and IPCP are punted to the routing protocol. Therefore, NSP at the PE router must provide the following functionality for address resolution:
  - Ethernet--PE device acts as a proxy-ARP server to all ARP requests from the CE router. The PE router responds with the MAC address of its local interface.
  - ATM and Frame Relay point-to-point--By default, inverse ARP does not run in the point-to-point Frame Relay or ATM subinterfaces. The IP address and subnet mask define the connected prefix; therefore, configuration is not required in the CE devices.

- Interworking requires that the MTUs in both ACs match for the pseudowire to come up. The default MTU in one AC should match with the MTU of other AC. The table below lists the range of MTUs that can be configured for different ACs.

Table 105: Range of MTUs for Different ACs

AC type	Range of MTUs supported
ATM	64 to 17940
Gigabit Ethernet	1500 to 4470
POS	64to 9102
Fast Ethernet	64to 9192

Note The MTU configured on the AC should not exceed the MTU in the core network. This ensures that the traffic is not fragmented.

- The CE routers with Ethernet attachment VCs running OSPF must be configured with the **ospfIfType**option so that the OSPF protocol treats the underlying physical broadcast link as a P2P link.

Ethernet VLAN-to-ATM AAL5 Interworking

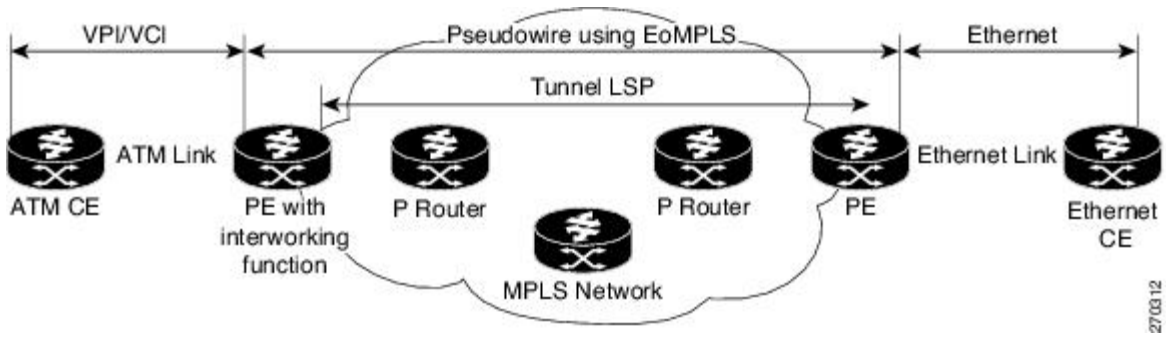
The following topics are covered in this section:

ATM AAL5-to-Ethernet Port AToM--Bridged Interworking

This interworking type provides interoperability between the ATM attachment VC and Ethernet attachment VC connected to different PE routers. Bridged encapsulation corresponding to the bridged (Ethernet) interworking mechanism is used.

The interworking function is performed at the PE router connected to the ATM attachment VC based on multiprotocol encapsulation over ATM AAL5 (see the figure below).

Figure 59: Network Topology for ATM-to-Ethernet AToM Bridged Interworking



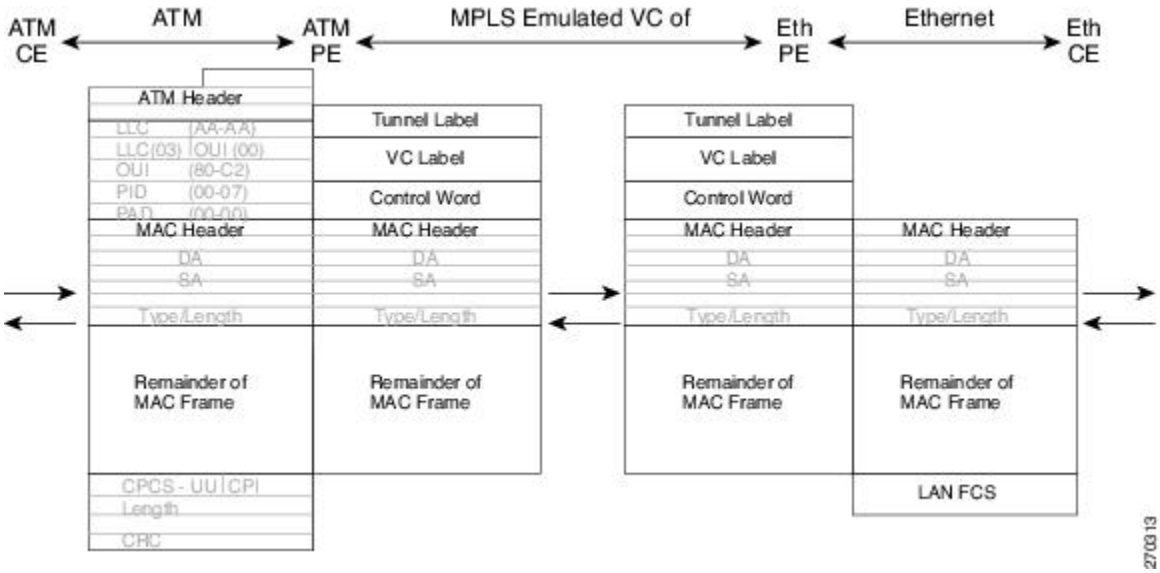
The advantage of this architecture is that the Ethernet PE router (connected to the Ethernet segment) operates similarly to Ethernet like-to-like services.

On the PE router with interworking function, in the direction from the ATM segment to MPLS cloud, the bridged encapsulation (ATM/subnetwork access protocol (SNAP) header) is discarded and the Ethernet frame is encapsulated with the labels required to go through the pseudowire using the VC type 5 (Ethernet) (see the figure below).

In the opposite direction, after the label disposition from the MPLS cloud, Ethernet frames are encapsulated over AAL5 using bridged encapsulation.

The figure below shows the protocol stack for ATM-to-Ethernet AToM bridged interworking. The ATM side has an encapsulation type of aal5snap.

Figure 60: Protocol Stack for ATM-to-Ethernet AToM Bridged Interworking--without VLAN Header



ATM AAL5-to-Ethernet VLAN 802.1Q AToM--Bridged Interworking

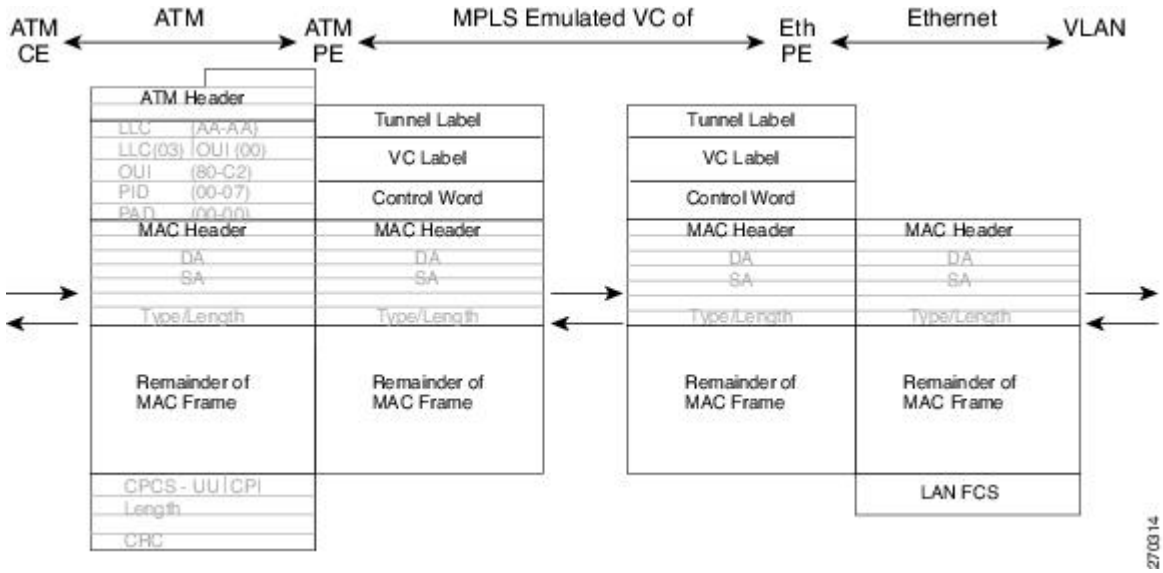
This interworking type provides interoperability between the ATM attachment VC and Ethernet VLAN attachment VC connected to different PE routers. Bridged encapsulation corresponding to the bridged (Ethernet) interworking mechanism is used.

The interworking function is performed in the same way as for the ATM-to-Ethernet port case, implemented on the PE router connected to the ATM attachment VC. The implementation is based on multiprotocol encapsulation over ATM AAL5 (see the figure below).

For the PE router connected to the Ethernet side, one major difference exists due the existence of the VLAN header in the incoming packet. The PE router discards the VLAN header of the incoming frames from the VLAN CE router, and the PE router inserts a VLAN header into the Ethernet frames traveling from the MPLS cloud. The frames sent on the pseudowire (with VC type 5) are Ethernet frames without the VLAN header.

Encapsulation over ATM AAL5 is shown in the figure below.

Figure 61: Protocol Stack for ATM -to-VLAN AToM Bridged Interworking



ATM-to-Ethernet--Routed Interworking

To perform routed interworking, both the ATM PE router and Ethernet PE router must be configured. The figure below shows the routed interworking between ATM to Ethernet. The IP encapsulation over the pseudowire is performed on the ATM packets arriving from the ATM CE router.

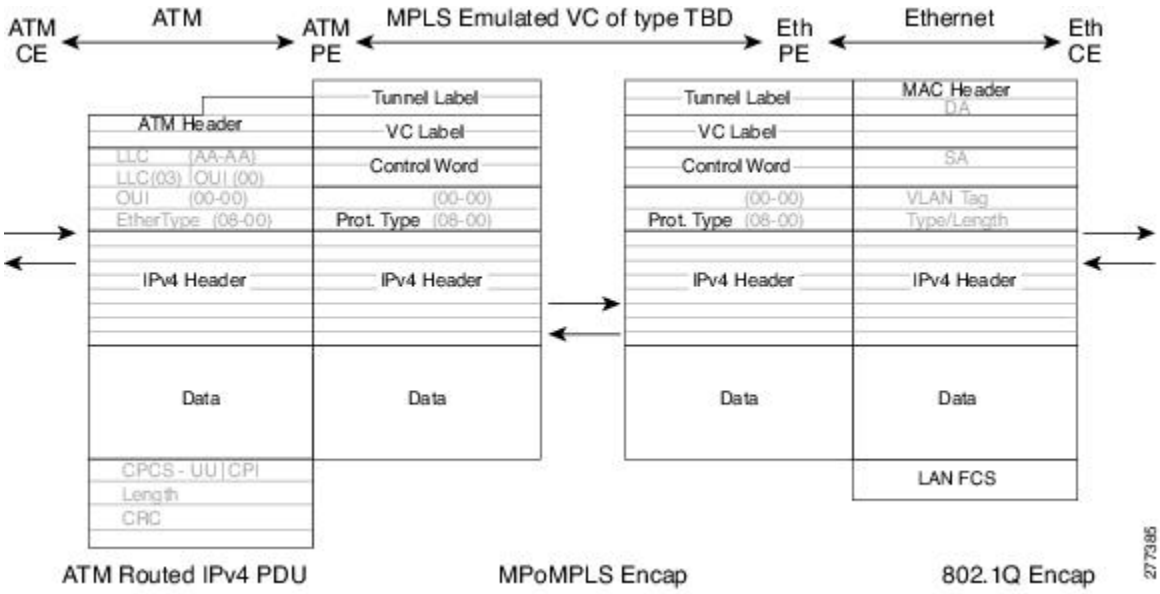
The address resolution is done at the ATM PE router; it is required when the ATM CE router does an inverse ARP. It is not required when the ATM CE router is configured using Point-to-Point (P2P) subinterfaces or static maps.

When packets arrive from the Ethernet CE router, the Ethernet PE router removes the L2 frame tag, and then forwards the IP packet to the egress PE router, using IPoMPLS encapsulation over the pseudowire. The Ethernet PE router makes the forwarding decision based on the L2 circuit ID, the VLAN ID, or port ID, of the incoming L2 frame. At the ATM PE router, after label disposition, the IP packets are encapsulated over the AAL5 using routed encapsulation based on RFC 2684.

The address resolution at the Ethernet PE router can be done when the Ethernet CE router configures the static ARP, or by the proxy ARP on the Ethernet PE router. If the proxy ARP is used, the IP address of the remote CE router can be learned dynamically.

Routing protocols need to be configured to operate in the P2P mode on the Ethernet CE router.

Figure 62: Protocol Stack for ATM-to-Ethernet--Routed Interworking



Ethernet VLAN-to-Frame Relay Interworking

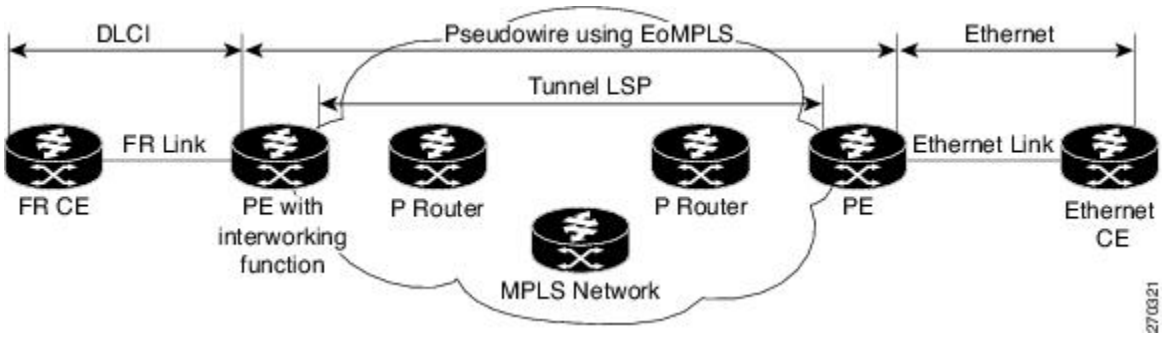
The following topics are covered in this section:

Frame Relay DLCI-to-Ethernet Port AToM--Bridged Interworking

This interworking type provides interoperability between the Frame Relay attachment VC and Ethernet attachment VC connected to different PE routers. Bridged encapsulation corresponding to the bridged (Ethernet) interworking mechanism is used.

For an FR-to-Ethernet port case, the interworking function is performed at the PE router connected to the FR attachment VC based on multiprotocol interconnect over Frame Relay (see the figure below). The interworking is implemented similar to an ATM-to-Ethernet case.

Figure 63: Network Topology for FR-to-Ethernet AToM Bridged Interworking



The advantage of this architecture is that the Ethernet PE router (connected to the Ethernet segment) operates similar to Ethernet like-to-like services: a pseudowire label is assigned to the Ethernet port and then the remote Label Distribution Protocol (LDP) session distributes the labels to its peer PE router. Ethernet frames are carried through the MPLS network using Ethernet over MPLS (EoMPLS).

On the PE router with interworking function, in the direction from the Frame Relay segment to the MPLS cloud, the bridged encapsulation (FR/SNAP header) is discarded and the Ethernet frame is encapsulated with the labels required to go through the pseudowire using the VC type 5 (Ethernet) (see the figure below).

In the opposite direction, after the label disposition from the MPLS cloud, Ethernet frames are encapsulated over Frame Relay using bridged encapsulation.

The following translations are supported:

- Ethernet without LAN FCS (0300800080C20007)
- Spanning tree (0300800080C2000E)

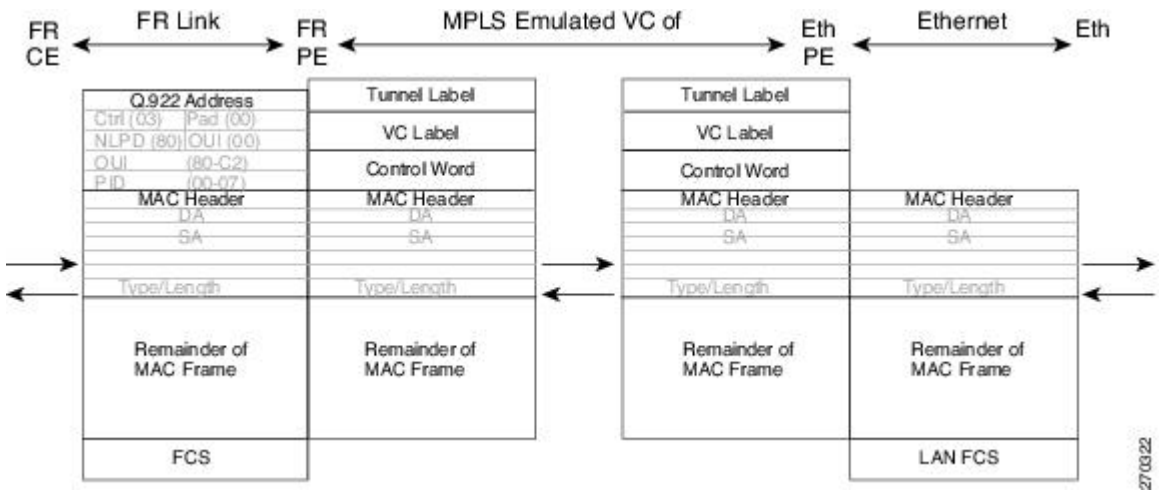
The PE router automatically supports translation of both Cisco and IETF Frame Relay encapsulation types coming from the CE, but translates only to IETF when sending to the CE router. This is not a problem for the Cisco CE router, because it can handle IETF encapsulation on receipt even if it is configured to send Cisco encapsulation.

The existing QoS functionality for Frame Relay is supported. The PVC status signaling works the same way as in the like-to-like case. The PE router reports the PVC status to the CE router, based on the availability of the pseudo wire.

The AC MTU must match when connected over MPLS. Only Frame Relay DLCI mode is supported; Frame Relay port mode is not supported in the bridged interworking.

The figure below shows the protocol stack for FR-to-Ethernet bridged interworking.

Figure 64: Protocol Stack for FR-to-Ethernet AToM Bridged Interworking--without VLAN Header



Frame Relay DLCI-to-Ethernet VLAN 802.1Q AToM--Bridged Interworking

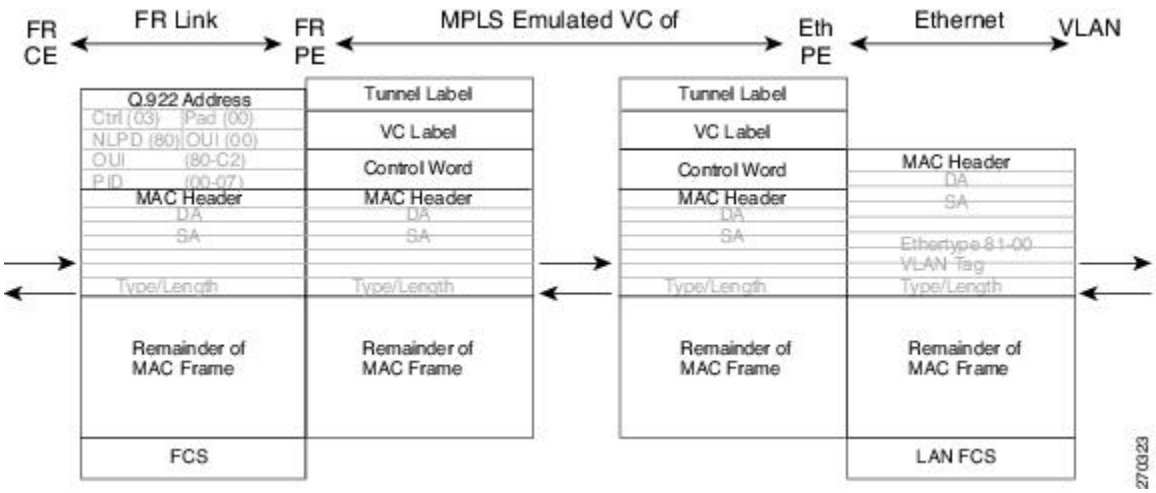
This interworking type provides interoperability between the Frame Relay attachment VC and Ethernet VLAN Attachment VC connected to different PE routers. The bridged encapsulation corresponding to the bridged (Ethernet) interworking mechanism is used.

The interworking function is performed in the same way as it is done for the Frame Relay to Ethernet port case; it is implemented on the PE router connected to the Frame Relay attachment VC, based upon a multiprotocol interconnect over Frame Relay (see the figure above).

As in the ATM-to-VLAN case, one difference exists on the Ethernet side due the existence of the VLAN header in the incoming packet. The PE router on the VLAN side discards the VLAN header of the incoming frames from the VLAN CE router, and the PE router inserts a VLAN header into the Ethernet frames traveling from the MPLS cloud. The frames sent on the pseudowire (with VC type 5) are Ethernet frames without the VLAN header.

The figure below shows the protocol stack for FR-to-VLAN AToM bridged interworking.

Figure 65: Protocol Stack for FR-to-VLAN AToM Bridged Interworking



Frame Relay DLCI-to-Ethernet VLAN Qot1Q QinQ AToM - Bridged Interworking

This interworking type provides interoperability between the Frame Relay Attachment VC and Ethernet VLAN Attachment VC connected to different PE routers. The bridged encapsulation corresponding to bridged (Ethernet) interworking mechanism is used.

The interworking function is done in the same way as it is done for FR-to-Ethernet port case; it is implemented on the PE router connected to the Frame Relay attachment VC, based on RFC 2427(Multiprotocol Interconnect over Frame Relay).

When compared with Frame Relay DLCI-to-Ethernet port AToM, there is one major difference on the Ethernet access side, due the existence of the VLAN header in the incoming packet. The PE router on the VLAN side will discard the VLAN header of the incoming frames form the VLAN CE router, and it will insert a VLAN header into the Ethernet frames coming from the MPLS cloud. So the frames sent on the pseudo wire (with VC type 5) will be Ethernet frames without the VLAN header.

The following translations are supported on the Frame Relay PE router:

- Ethernet without LAN FCS (0300800080C20007)
- Spanning tree (0300800080C2000E)

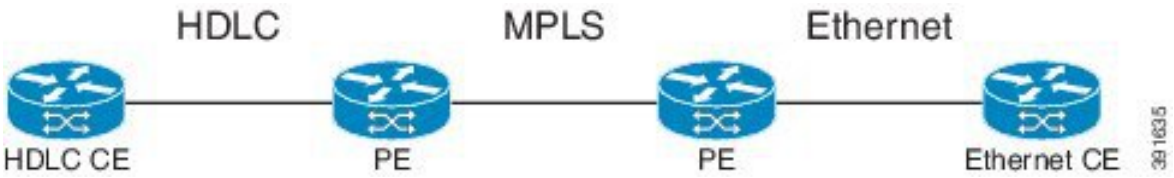
Frame Relay encapsulation types supported for bridged interworking: Cisco and IETF for incoming traffic, IETF only for outgoing traffic towards CE router.

HDLC-to-Ethernet Interworking

High-Level Data Link Control (HDLC) and Ethernet are two independent data link layer transport protocols that utilize the Any Transport over MPLS (AToM) framework to communicate with each other. The interworking function enables translation between two heterogeneous Layer 2 encapsulations over a Multiprotocol Label Switching (MPLS) backbone.

The figure below depicts a simple HDLC-to-Ethernet interworking topology.

Figure 66: HDLC-to-Ethernet interworking topology



HDLC-to-Ethernet interworking supports the following:

- Ethernet or bridged interworking
- IP or routed interworking
- HDLC encapsulation type: CISCO
- Ethernet encapsulation types: IEEE 802.1Q, QinQ, port mode

The HDLC pass-through feature is not affected in any way by HDLC-to-Ethernet interworking.

HDLC-to-Ethernet interworking supports two interworking modes:

- HDLC-to-Ethernet — Ethernet or Bridged interworking
- HDLC-to-Ethernet — IP or Routed interworking

HDLC-to-Ethernet — Ethernet or Bridged Interworking

HDLC-to-Ethernet bridged interworking provides interoperability between the HDLC attachment virtual circuit (VC) and Ethernet VLAN attachment VC connected to different provider edge (PE) devices. Bridged encapsulation corresponding to the bridged (Ethernet) interworking mechanism is used.

When packets arrive from the HDLC customer edge (CE) device, they consist of the HDLC header, the Ethernet MAC header, and the payload. At the HDLC PE device, the HDLC header is removed, and MPLS labels are inserted. The frames are then routed over the pseudowire to the Ethernet PE device, where the MPLS labels are removed. On the Ethernet side, there are two possibilities. The attachment circuit (AC) is either Ethernet or VLAN.

For an Ethernet attachment circuit (AC), the packets are forwarded to the Ethernet CE device, as is. For a VLAN AC, VLAN headers are added at the VLAN/QinQ subinterface’s AC. The Ethernet VLAN frame is then forwarded to the VLAN CE device.

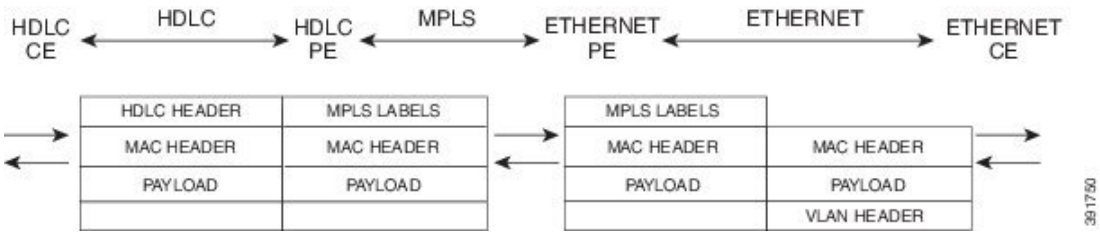
In the opposite direction (Ethernet / VLAN to HDLC), the VLAN header is present in the incoming packet, if the AC is VLAN. So, when packets arrive from the VLAN CE device, they consist of the VLAN header, the Ethernet MAC header, and the payload. At the Ethernet PE device, the VLAN header is removed at the VLAN/QinQ subinterface's AC, and MPLS labels are inserted. The frames are then routed over the pseudowire

to the HDLC PE device, where the MPLS labels are removed. The HDLC header is added before the Ethernet MAC header. The HDLC frame is then forwarded to the HDLC CE device.

If the AC is Ethernet, packets arriving from the Ethernet CE device consist of the Ethernet MAC header and the payload. At the Ethernet PE device, MPLS labels are inserted at the VLAN/QinQ subinterface's AC. The frames are then routed over the pseudowire to the HDLC PE device, where the MPLS labels are removed. The HDLC header is added before the Ethernet MAC header. The HDLC frame is then forwarded to the HDLC CE device.

The figure below shows the bridged interworking mode of HDLC-to-Ethernet interworking, with a VLAN AC on the Ethernet side.

Figure 67: HDLC-to-Ethernet — Ethernet or Bridged Interworking



HDLC-to-Ethernet — IP or Routed Interworking

To perform routed interworking, both the HDLC PE device and Ethernet PE device must be configured. The IP encapsulation over the pseudowire is performed on HDLC packets that arrive from the HDLC CE device. The address resolution is done at the HDLC PE device.

When packets arrive from the HDLC CE device, they consist of the HDLC header, the IPv4 header, and the payload. At the HDLC PE device, the HDLC header is removed, and MPLS labels are inserted. The frames are then routed over the pseudowire to the Ethernet PE device, where the MPLS labels are removed. On the Ethernet side, there are two possibilities. The attachment circuit (AC) is either Ethernet or VLAN.

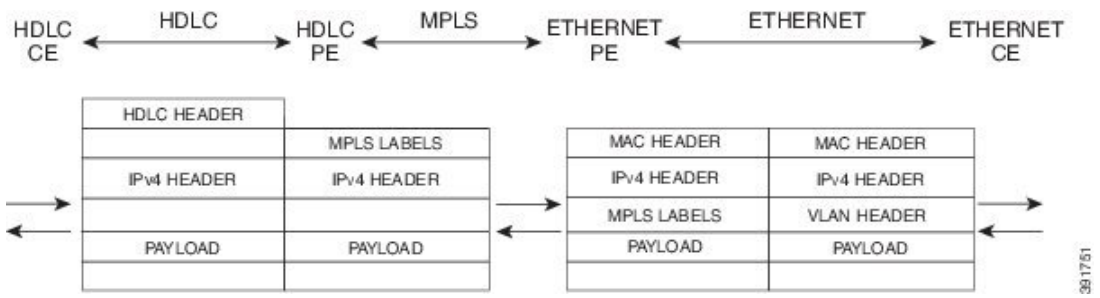
For an Ethernet attachment circuit (AC), the packets are forwarded to the Ethernet CE device, as is. For a VLAN AC, VLAN headers are added at the VLAN/QinQ subinterface’s AC. The Ethernet VLAN frame is then forwarded to the VLAN CE device.

In the opposite direction (Ethernet / VLAN to HDLC), the VLAN header is present in the incoming packet, if the AC is VLAN. So, when packets arrive from the VLAN CE device, they consist of the VLAN header, the Ethernet MAC header, and the payload. At the Ethernet PE device, the MAC header is removed, the VLAN header is removed at the VLAN/QinQ subinterface's AC, and MPLS labels are inserted. The frames are then routed over the pseudowire to the HDLC PE device, where the MPLS labels are removed. The HDLC header is added before the IPv4 header. The HDLC frame is then forwarded to the HDLC CE device.

If the AC is Ethernet, packets arriving from the Ethernet CE device consist of the Ethernet MAC header and the payload. At the Ethernet PE device, the MAC header is removed, and MPLS labels are inserted. The frames are then routed over the pseudowire to the HDLC PE device, where the MPLS labels are removed. The HDLC header is added before the IPv4 header. The HDLC frame is then forwarded to the HDLC CE device.

The figure below shows the routed interworking mode of HDLC-to-Ethernet interworking, with a VLAN AC on the Ethernet side.

Figure 68: HDLC-to-Ethernet — IP or Routed interworking



ATM Local Switching

- ATM like-to-like local switching allows switching data between two physical interfaces where both the segments are of ATM type. The two interfaces must be on the same PE router. The table below lists the supported ATM local switching combinations.

Table 106: ATM local switching - supported combinations

	Same port Point-to-Point	Different port Point-to-Point	Same Port Multipoint	Different Port Multipoint
Port Mode	No	No	No	No
VC-to-VC AAL0	Yes	Yes	Yes	Yes
VC-to-VC AAL5	Yes	Yes	Yes	Yes
VP-to-VP AAL0	No	No	Yes	Yes
VP-to-VP AAL5	No	No	No	No

VC-to-VC Local Switching

VC-to-VC local switching transports cells between two ATM attachment VCs on the same or different port on the PE router. The cells coming to the PE router can be AAL0 or AAL5 encapsulated ATM packets. ATM VC-to-VC local switching can be configured either on point-to-point interface or on multipoint interface.

There are two operation modes for managing OAM cells over ATM local switching interfaces:

- OAM transparent mode: In this mode, the PE router transports F5 OAM cells transparently across local switching interfaces.
- OAM local emulation mode: In this mode, the PE router does not transport OAM cells across local switching interfaces. Instead, the interfaces locally terminate and process F5 OAM cells.

In ATM single cell relay AAL0, the ATM virtual path identifier/virtual channel identifier (VPI/VCI) values of the ingress and egress ATM interfaces of a router must match. If L2 local switching is desired between two ATM VPIs and VCIs, which are on two different interfaces and have values that do not match, ATM AAL5 should be selected. However, if ATM AAL5 uses OAM transparent mode, the VPI and VCI values must match.

ATM OAM can be configured on ATM VC mode local switching AC using the **oam-ac emulation-enable**and **oam-pvc manage** commands. When emulation is enabled on the AC, all OAM cells going through the AC are punted to RP for local processing. The ATM common component processes OAM cells and forwards the cells towards the local CE router. This helps to detect the failures on the PE router by monitoring the response at the CE router end. When the **oam-pvc manage** command is enabled on the AC, the PVC generates end-to-end OAM loopback cells that verify connectivity on the VC.

The following example shows a sample configuration on the ATM PE router:

```
configure terminal
interface atm 4/0.50 multipoint
 no ip address
  no atm enable-ilmi-trap
pvc 100/100 l2transport
 encapsulation aal5
 oam-ac emulation-enable
 oam-pvc manage
interface atm 5/0.100 multipoint
 no ip address
 no atm enable-ilmi-trap
 pvc 100/100 l2transport
  encapsulation aal5
  oam-ac emulation-enable
  oam-pvc manage
connect atm_ls atm 4/0 100/100 atm 5/0 100/100
```

VP-to-VP Local Switching

VP-to-VP local switching transports cells between two VPs on the same port or different ports on the PE router. The cells coming to the PE router can be AAL0 encapsulated ATM packets only. ATM VP-to-VP local switching can be configured only on multipoint interfaces.

There are two operation modes for managing OAM cells over ATM local switching interfaces:

- OAM transparent mode: In this mode, the PE router transports F4 OAM cells transparently across local switching interfaces.
- OAM local emulation mode: In this mode, the PE router do not transport OAM cells across local switching interfaces. Instead, the interfaces locally terminate and process F4 OAM cells.

In ATM single cell relay AAL0, the ATM VPI values of the ingress and egress ATM interfaces on a router must match. If L2 switching is desired between two ATM VPIs which are on two different interfaces and have values that do not match, ATM AAL5 should be selected. If ATM AAL5 uses OAM transparent mode, the VPI value must match. Currently, the ATM VP-to-VP local switching supports only AAL0 encapsulation.

The following example shows a sample configuration on the ATM PE router:

```
configure terminal
interface atm 4/0.100 multipoint
 no ip address
 no atm enable-ilmi-trap
atm pvp 100 l2transport
interface atm 5/0.100 multipoint
 no ip address
 no atm enable-ilmi-trap
atm pvp 100 l2transport
connect atm_ls atm 4/0 100 atm 5/0 100
```

PPP-to-Ethernet AToM-Routed Interworking

In this interworking type, one of the ACs is Ethernet and the other is PPP. Each link is terminated locally on the corresponding PE routers and the extracted layer 3 (L3) packets are transported over a pseudowire.

The PE routers connected to Ethernet and PPP ACs terminate their respective L2 protocols. The PPP session is terminated for both the LCP and the Network Control Protocol (NCP) layers. On the ingress PE router, after extracting L3 packets, each PE router forwards the packets over the already established pseudowire using MPoMPLS encapsulation. On the egress PE router, after performing label disposition, the packets are encapsulated based on the corresponding link layer and are sent to the respective CE router. This interworking scenario requires the support of MPoMPLS encapsulation by the PE routers.

In PPP-to-Ethernet AToM routed interworking mode IPCP is supported. Proxy IPCP is automatically enabled on the PE router when IP interworking is configured on the pseudowire. By default, the PE router gets the IP address it needs to use from the CE router. The PE router accomplishes this by sending an IPCP confreq with the IP address 0.0.0.0. The local CE router has the remote CE router's IP address configured on it. The following example shows a sample configuration on the PPP CE router:

```
interface serial2/0
 ip address 168.65.32.13 255.255.255.0
 encapsulation ppp
peer default ip address 168.65.32.14  *
```

If the remote CE router's IP address cannot be configured on the local CE router, then the remote CE router's IP address can be configured on the PE router using the **ppp ipcp address proxy** *ip address* command on the xconnect PPP interface of PE router. The following example shows a sample configuration on the PPP PE router:

```
pseudowire-class mp
 encapsulation mpls
 protocol ldp
 interworking ip
!
int se2/0
 encap ppp
 xconnect 10.0.0.2 200 pw-class mp
 ppp ipcp address proxy 168.65.32.14
```

PPP-to-Ethernet AToM-Routed Interworking using the commands associated with the L2VPN Protocol-Based CLIs feature

In this interworking type, one of the ACs is Ethernet and the other is PPP. Each link is terminated locally on the corresponding PE routers and the extracted layer 3 (L3) packets are transported over a pseudowire.

The PE routers connected to Ethernet and PPP ACs terminate their respective L2 protocols. The PPP session is terminated for both the LCP and the Network Control Protocol (NCP) layers. On the ingress PE router, after extracting L3 packets, each PE router forwards the packets over the already established pseudowire using MPoMPLS encapsulation. On the egress PE router, after performing label disposition, the packets are encapsulated based on the corresponding link layer and are sent to the respective CE router. This interworking scenario requires the support of MPoMPLS encapsulation by the PE routers.

In PPP-to-Ethernet AToM routed interworking mode IPCP is supported. Proxy IPCP is automatically enabled on the PE router when IP interworking is configured on the pseudowire. By default, the PE router gets the IP address it needs to use from the CE router. The PE router accomplishes this by sending an IPCP confreq with

the IP address 0.0.0.0. The local CE router has the remote CE router's IP address configured on it. The following example shows a sample configuration on the PPP CE router:

```
interface serial2/0
 ip address 168.65.32.13 255.255.255.0
 encapsulation ppp
peer default ip address 168.65.32.14  *
```

If the remote CE router's IP address cannot be configured on the local CE router, then the remote CE router's IP address can be configured on the PE router using the **ppp ipcp address proxy** *ip address* command on the xconnect PPP interface of PE router. The following example shows a sample configuration on the PPP PE router:

```
template type pseudowire mp
 encapsulation mpls
 protocol ldp
 interworking ip
!
int se2/0
 encap ppp
interface pseudowire 100
 source template type pseudowire mp
 neighbor 33.33.33.33 1
!
l2vpn xconnect context con1
 ppp ipcp address proxy 168.65.32.14
```

Static IP Addresses for L2VPN Interworking for PPP

If the PE router needs to perform address resolution with the local CE router for PPP, configure the remote CE router’s IP address on the PE router. Use the **ppp ipcp address proxy** command with the remote CE router’s IP address on the PE router’s xconnect PPP interface. The following example shows a sample configuration:

```
pseudowire-class ip-interworking
 encapsulation mpls
 interworking ip
interface Serial2/0
 encapsulation ppp
 xconnect 10.0.0.2 200 pw-class ip-interworking
 ppp ipcp address proxy 10.65.32.14
```

You can also configure the remote CE router’s IP address on the local CE router with the **peer default ip address** command if the local CE router performs address resolution.

Static IP Addresses for L2VPN Interworking for PPP using the commands associated with the L2VPN Protocol-Based CLIs feature

If the PE router needs to perform address resolution with the local CE router for PPP, configure the remote CE router’s IP address on the PE router. Use the **ppp ipcp address proxy** command with the remote CE router’s IP address on the PE router’s xconnect PPP interface. The following example shows a sample configuration:

```
template type pseudowire ip-interworking
```

```
 encapsulation mpls
 interworking ip
interface Serial2/0
 encapsulation ppp
interface pseudowire 100
 source template type pseudowire ip-interworking
 neighbor 10.0.0.2 200
!
l2vpn xconnect context con1
 ppp ipcp address proxy 10.65.32.14
```

You can also configure the remote CE router's IP address on the local CE router with the **peer default ip address** command if the local CE router performs address resolution.

How to Configure L2VPN Interworking

Configuring L2VPN Interworking

L2VPN interworking allows you to connect disparate ACs. Configuring L2VPN interworking feature requires that you add the **interworking** command to the list of commands that make up the pseudowire. The steps for configuring the pseudowire for L2VPN interworking are included in this section. You use the **interworking**command as part of the overall AToM configuration. For specific instructions on configuring AToM, see the Any Transport over MPLS document.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation** {**mpls** | **l2tpv3**}
5. **interworking** {**ethernet** | **ip**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *name* **Example:**	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.

	Command or Action	Purpose
	`Router(config)# pseudowire-class class1`	
Step 4	**encapsulation** {**mpls** | **l2tpv3**} **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation, which is either **mpls** or **l2tpv3**.
Step 5	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 6	**end** **Example:** `Router(config-pw)# end`	Exits pseudowire class configuration mode and returns to privileged EXEC mode.

Verifying the L2VPN Configuration

You can verify L2VPN configuration using the following steps:

- You can issue the **show arp** command between the CE routers to ensure that data is being sent:

```
Router# show arp
Protocol   Address       Age (min)   Hardware Addr    Type    Interface
Internet   10.1.1.5           134    0005.0032.0854   ARPA    FastEthernet0/0/0
Internet   10.1.1.7             -    0005.0032.0000   ARPA    FastEthernet0/0/0
```

- You can issue the **ping** command between the CE routers to ensure that data is being sent:

```
Router# ping 10.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

- You can verify the AToM configuration by using the **show mpls l2transport vc detail** command.

Configuring L2VPN Interworking using the commands associated with the L2VPN Protocol-Based CLIs feature

L2VPN Interworking allows you to connect disparate attachment circuits. Configuring the L2VPN Interworking feature requires that you add the **interworking** command to the list of commands that make up the pseudowire. The steps for configuring the pseudowire for L2VPN Interworking are included in this section. You use the **interworking**command as part of the overall AToM or L2TPv3 configuration. For specific instructions on configuring AToM or L2TPv3, see the following documents:

- Layer 2 Tunnel Protocol Version 3

- Any Transport over MPLS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hw-module slot** *slot-number* **np mode feature**
4. **interface pseudowire** *number*
5. **encapsulation** {**mpls** | **l2tpv3**}
6. **interworking** {**ethernet** | **ip**}
7. **neighbor** *peer-address vcid-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**hw-module slot** *slot-number* **np mode feature** **Example:** `Router(config)# hw-module slot 3 np mode feature`	(Optional) Enables L2VPN Interworking functionality on the Cisco 12000 series router. **Note** Enter this command only on a Cisco 12000 series Internet router if you use L2TPv3 for L2VPN Interworking on an ISE (Engine 3) or Engine 5 interface. In this case, you must first enable the L2VPN feature bundle on the line card by entering the **hw-module slot** *slot-number* **np mode feature** command.
Step 4	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 1`	Establishes an interface pseudowire with a value that you specify and enters pseudowire class configuration mode.
Step 5	**encapsulation** {**mpls**	**l2tpv3**} **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation, which is either **mpls** or **l2tpv3**.
Step 6	**interworking** {**ethernet**	**ip**} **Example:**	Specifies the type of pseudowire and the type of traffic that can flow across it.

	Command or Action	Purpose
	`Router(config-pw)# interworking ip`	**Note** On the Cisco 12000 series Internet router, Ethernet (bridged) interworking is not supported for L2TPv3. After you configure the L2TPv3 tunnel encapsulation for the pseudowire using the **encapsulation l2tpv3**command, you cannot enter the **interworking ethernet** command.
Step 7	**neighbor** *peer-address vcid-value* **Example:** `Router(config-pw)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.

Verifying the L2VPN Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

You can verify L2VPN configuration using the following commands:

- You can issue the **show arp** command between the CE routers to ensure that data is being sent:

```
Device# show arp
Protocol   Address       Age (min)   Hardware Addr    Type    Interface
Internet   10.1.1.5           134    0005.0032.0854   ARPA    FastEthernet0/0/0
Internet   10.1.1.7             -    0005.0032.0000   ARPA    FastEthernet0/0/0
```

- You can issue the **ping** command between the CE routers to ensure that data is being sent:

```
Device# ping 10.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

- You can verify the AToM configuration by using the **show l2vpn atom vc detail** command.

Configuring Ethernet VLAN-to-ATM AAL5 Interworking

This section explains the following AToM configurations:

ATM AAL5-to-Ethernet Port

You can configure the ATM AAL5-to-Ethernet Port feature on a PE1 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*

5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface atm** *slot* / *subslot* / *port* **.** *subinterface number*
10. **pvc** [*name*] *vpi* / *vci* **12transport**
11. **encapsulation aal5snap**
12. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.

	Command or Action	Purpose
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface atm** *slot* / *subslot* / *port* **.** *subinterface number* **Example:** `Router(config-pw)# interface atm 2/0/0.1`	Configures an ATM interface and enters interface configuration mode.
Step 10	**pvc** [*name*] *vpi* / *vci* **12transport** **Example:** `Router(config-subif)# pvc 0/200 l2transport`	Assigns a name to an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 11	**encapsulation aal5snap** **Example:** `Router(config-if-atm-member)# encapsulation aal5snap`	Configures the ATM AAL and encapsulation type for an ATM VC.
Step 12	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:** `Router(config-if-atm-member)# xconnect 10.0.0.200 140 pw-class atm-eth`	Binds an AC to a pseudowire and configures an AToM static pseudowire.
Step 13	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

ATM AAL5-to-Ethernet Port using the commands associated with the L2VPN Protocol-Based CLIs feature

You can configure the ATM AAL5-to-Ethernet Port feature on a PE1 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **template type pseudowire** [*pw-class-name*]
7. **encapsulation mpls**

8. **interworking** {**ethernet** | **ip**}
9. **interface atm** *slot* / *subslot* / *port* **.** *subinterface number*
10. **pvc** [*name*] *vpi* / *vci* **12transport**
11. **encapsulation aal5snap**
12. **end**
13. **interface pseudowire** *number*
14. **source template type pseudowire** *template-name*
15. **neighbor** *peer-address vcid-value*
16. **exit**
17. **exit**
18. **l2vpn xconnect context** *context-name*
19. **member pseudowire** *interface-number*
20. **member** *ip-address vc-id* **encapsulation mpls**
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**template type pseudowire** [*pw-class-name*] **Example:**	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.

	Command or Action	Purpose
	`Router(config-if)# template type pseudowire atm-eth`	
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface atm** *slot* / *subslot* / *port* **.** *subinterface number* **Example:** `Router(config-pw)# interface atm 2/0/0.1`	Configures an ATM interface and enters interface configuration mode.
Step 10	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-subif)# pvc 0/200 l2transport`	Assigns a name to an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 11	**encapsulation aal5snap** **Example:** `Router(config-if-atm-member)# encapsulation aal5snap`	Configures the ATM AAL and encapsulation type for an ATM VC.
Step 12	**end** **Example:** `Router(config-if-atm-member)# end`	Exits to privileged EXEC mode.
Step 13	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 14	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire atm-eth`	Configures the source template of type pseudowire named atm-eth.
Step 15	**neighbor** *peer-address vcid-value* **Example:**	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.

	Command or Action	Purpose
	`Router(config-if)#  neighbor 10.0.0.200 140`	
Step 16	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 17	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 18	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 19	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 20	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.200 140 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 21	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

ATM AAL5-to-Ethernet Port on a PE2 Router

You can configure the ATM AAL5-to-Ethernet Port feature on a PE2 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}

9. **interface** *type slot* / *subslot* / *port*
10. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.

	Command or Action	Purpose
Step 9	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface gigabitethernet 5/1/0`	Configure an interface and enters interface configuration mode.
Step 10	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:** `Router(config-if)# xconnect 10.0.0.100 140 pw-class atm-eth`	Binds an AC to a pseudowire and configures an AToM static pseudowire.
Step 11	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configure mode and returns to privileged EXEC mode.

What to do next

Note When configuring bridged interworking, the PE2 router configuration does not include the **interworking ethernet** command because it is treated as like-to-like, and also because the AC is already an Ethernet port. However, when configuring routed interworking, the **interworking ip** command is required.

ATM AAL5-to-Ethernet Port on a PE2 Router using the commands associated with the L2VPN Protocol-Based CLIs feature

You can configure the ATM AAL5-to-Ethernet Port feature on a PE2 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **template type pseudowire** [*pseudowire-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface** *type slot / subslot / port*
10. **end**
11. **interface pseudowire** *number*
12. **source template type pseudowire** *template-name*
13. **neighbor** *peer-address vcid-value*
14. **exit**
15. **l2vpn xconnect context** *context-name*

16. **member pseudowire** *interface-number*
17. **member** *ip-address vc-id* **encapsulation mpls**
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**template type pseudowire** [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire atm-eth`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.

	Command or Action	Purpose
Step 9	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface gigabitethernet 5/1/0`	Configure an interface and enters interface configuration mode.
Step 10	**end** **Example:** `Router(config-pw)# end`	Exits to privileged EXEC mode.
Step 11	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 12	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire atm-eth`	Configures the source template of type pseudowire named atm-eth
Step 13	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)#  neighbor 10.0.0.100 140`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 15	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 16	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 17	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.100 140 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

	Command or Action	Purpose
Step 18	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

What to do next

Note When configuring bridged interworking, the PE2 router configuration does not include the **interworking ethernet** command because it is treated as like-to-like, and also because the AC is already an Ethernet port. However, when configuring routed interworking, the **interworking ip** command is required.

ATM AAL5-to-Ethernet VLAN 802.1Q on a PE1 Router

You can configure the ATM AAL5-to-Ethernet VLAN 802.1Q feature on a PE1 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface atm** *slot* / *subslot* / *port* **.** *subinterface number*
10. **pvc** [*name*] *vpi* / *vci* **12transport**
11. **encapsulation aal5snap**
12. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface atm** *slot* / *subslot* / *port* **.** *subinterface number* **Example:** `Router(config-pw)# interface atm 2/0/0.1`	Configure an ATM interface and enters interface configuration mode.
Step 10	**pvc** [*name*] *vpi* / *vci* **l2transport** **Example:** `Router(config-subif)# pvc 0/200 l2transport`	Assigns a name to an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 11	**encapsulation aal5snap** **Example:**	Configures the ATM AAL and encapsulation type for an ATM VC.

	Command or Action	Purpose
	`Router(config-if-atm-member)# encapsulation aal5snap`	
Step 12	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:** `Router(config-if-atm-member)# xconnect 10.0.0.200 140 pw-class atm-eth`	Binds an AC to a pseudowire and configures an AToM static pseudowire.
Step 13	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

ATM AAL5-to-Ethernet VLAN 802.1Q on a PE1 Router using the commands associated with the L2VPN Protocol-Based CLIs feature

You can configure the ATM AAL5-to-Ethernet VLAN 802.1Q feature on a PE1 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **template type pseudowire** [*pseudowire-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface atm** *slot* / *subslot* / *port* **.** *subinterface number*
10. **pvc** [*name*] *vpi* / *vci* **12transport**
11. **encapsulation aal5snap**
12. **end**
13. **interface pseudowire** *number*
14. **source template type pseudowire** *template-name*
15. **neighbor** *peer-address vcid-value*
16. **exit**
17. **l2vpn xconnect context** *context-name*
18. **member pseudowire** *interface-number*
19. **member** *ip-address vc-id* **encapsulation mpls**
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**template type pseudowire** [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire atm-eth`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface atm** *slot* / *subslot* / *port* **.** *subinterface number* **Example:** `Router(config-pw)# interface atm 2/0/0.1`	Configure an ATM interface and enters interface configuration mode.

	Command or Action	Purpose
Step 10	**pvc** [*name*] *vpi* / *vci* **12transport** **Example:** `Router(config-subif)# pvc 0/200 l2transport`	Assigns a name to an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 11	**encapsulation aal5snap** **Example:** `Router(config-if-atm-member)# encapsulation aal5snap`	Configures the ATM AAL and encapsulation type for an ATM VC.
Step 12	**end** **Example:** `Router(config-if-atm-member)# end`	Exits to privileged EXEC mode.
Step 13	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 14	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire atm-eth`	Configures the source template of type pseudowire named atm-eth
Step 15	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)#  neighbor 10.0.0.200 140`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 16	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 17	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 18	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.

	Command or Action	Purpose
Step 19	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.200 140 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 20	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

ATM AAL5-to-Ethernet VLAN 802.1Q on a PE2 router

You can configure the ATM AAL5-to-Ethernet VLAN 802.1Q feature on a PE2 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface** *type slot* / *subslot* / *port* . *subinterface-number*
10. **encapsulation dot1q** *vlan-id*
11. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:**	Establishes the label distribution protocol for the platform.

	Command or Action	Purpose
	`Router(config)# mpls label protocol ldp`	
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface** *type slot / subslot / port . subinterface-number* **Example:** `Router(config-pw)# interface gigabitethernet 5/1/0.3`	Configures an interface and enters interface configuration mode.
Step 10	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if)# encapsulation dot1q 1525`	Enables IEEE 802.1Q encapsulation of traffic on a specified sub interface in a VLAN.
Step 11	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:** `Router(config-if)# xconnect 10.0.0.100 140 pw-class atm-eth`	Binds an AC to a pseudowire and configures an AToM static pseudowire.
Step 12	**end** **Example:**	Exits xconnect configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if-xconn)# end`	

What to do next

Note In the case of ATM AAl5-to-VLAN, the PE2 router configuration includes the **interworking**command for both bridged and routed interworking.

Note To verify the L2VPN interworking status and check the statistics, refer to the Verifying L2VPN Interworking, on page 962.

ATM AAL5-to-Ethernet VLAN 802.1Q on a PE2 router using the commands associated with the L2VPN Protocol-Based CLIs feature

You can configure the ATM AAL5-to-Ethernet VLAN 802.1Q feature on a PE2 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **template type pseudowire** [*pseudowire-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface** *type slot / subslot / port . subinterface-number*
10. **encapsulation dot1q** *vlan-id*
11. **end**
12. **interface pseudowire** *number*
13. **source template type pseudowire** *template-name*
14. **neighbor** *peer-address vcid-value*
15. **exit**
16. **l2vpn xconnect context** *context-name*
17. **member pseudowire** *interface-number*
18. **member** *ip-address vc-id* **encapsulation mpls**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configure an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**template type pseudowire** [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire atm-eth`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface** *type slot / subslot / port . subinterface-number* **Example:**	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config-pw)# interface gigabitethernet 5/1/0.3`	
Step 10	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if)# encapsulation dot1q 1525`	Enables IEEE 802.1Q encapsulation of traffic on a specified sub interface in a VLAN.
Step 11	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 12	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 13	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire atm-eth`	Configures the source template of type pseudowire named atm-eth
Step 14	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)#  neighbor 10.0.0.100 140`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 16	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 17	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 18	**member** *ip-address vc-id* **encapsulation mpls** **Example:**	Creates the VC to transport the Layer 2 packets.

	Command or Action	Purpose
	`Router(config-xconnect)# member 10.0.0.100 140 encapsulation mpls`	
Step 19	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

What to do next

Note In the case of ATM AAl5-to-VLAN, the PE2 router configuration includes the **interworking**command for both bridged and routed interworking.

Note To verify the L2VPN interworking status and check the statistics, refer to the Verifying L2VPN Interworking, on page 962.

Configuring Ethernet VLAN-to-Frame Relay Interworking

This section explains the following AToM configurations and provides examples. The Network Topology for FR-to-Ethernet AToM Bridged Interworking figure above illustrates different AToM configurations.

Frame Relay DLCI-to-Ethernet Port on a PE1 Router

You can configure the Frame Relay DLCI-to-Ethernet Port feature on a PE1 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking ethernet**
9. **interface** *type slot / subslot / port*
10. **encapsulation frame-relay**
11. **connect** *connection-name interface dlci* {*interface dlci* | **l2transport**}
12. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class fr-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking ethernet** **Example:** `Router(config-pw)# interworking ethernet`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface serial 2/0/0`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 10	**encapsulation frame-relay** **Example:** `Router(config-if)# encapsulation frame-relay`	Enables Frame Relay encapsulation.
Step 11	**connect** *connection-name interface dlci* {*interface dlci* | **l2transport**} **Example:** `Router(config-if)# connect fr-vlan-1 POS2/3/1 151 l2transport`	Defines the connection between Frame Relay PVCs.
Step 12	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:** `Router(config-if)# xconnect 10.0.0.200 151 pw-class pw-class-bridge`	Binds an AC to a pseudowire and configures an AToM static pseudowire.
Step 13	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

Frame Relay DLCI-to-Ethernet Port on a PE1 Router using the commands associated with the L2VPN Protocol-Based CLIs feature

You can configure the Frame Relay DLCI-to-Ethernet Port feature on a PE1 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **template type pseudowire** [*pseudowire-name*]
7. **encapsulation mpls**
8. **interworking ethernet**
9. **interface** *type slot / subslot / port*
10. **encapsulation frame-relay**
11. **connect** *connection-name interface dlci* {*interface dlci* | **l2transport**}
12. **end**
13. **interface pseudowire** *number*
14. **source template type pseudowire** *template-name*
15. **neighbor** *peer-address vcid-value*
16. **exit**

17. **l2vpn xconnect context** *context-name*
18. **member pseudowire** *interface-number*
19. **member** *ip-address vc-id* **encapsulation mpls**
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**template type pseudowire** [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire fr-eth`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking ethernet** **Example:** `Router(config-pw)# interworking ethernet`	Specifies the type of pseudowire and the type of traffic that can flow across it.

	Command or Action	Purpose
Step 9	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface serial 2/0/0`	Configures an interface and enters interface configuration mode.
Step 10	**encapsulation frame-relay** **Example:** `Router(config-if)# encapsulation frame-relay`	Enables Frame Relay encapsulation.
Step 11	**connect** *connection-name interface dlci* {*interface dlci* | **l2transport**} **Example:** `Router(config-if)# connect fr-vlan-1 POS2/3/1 151 l2transport`	Defines the connection between Frame Relay PVCs.
Step 12	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 13	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 14	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire pwclass-bridge`	Configures the source template of type pseudowire named pwclass-bridge.
Step 15	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)#  neighbor 10.0.0.200 151`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 16	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 17	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

	Command or Action	Purpose
Step 18	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 19	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.200 151 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 20	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

Frame Relay DLCI-to-Ethernet Port on a PE2 router

You can configure the Frame Relay DLCI-to-Ethernet Port feature on a PE2 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking ethernet**
9. **interface** *type slot / subslot / port*
10. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking ethernet** **Example:** `Router(config-pw)# interworking ethernet`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface gigabitethernet 2/0/0`	Configures an interface and enters interface configuration mode.
Step 10	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:** `Router(config-if)# xconnect 10.0.0.200 140 pw-class atm-eth`	Binds an AC to a pseudowire and configures an AToM static pseudowire.
Step 11	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

What to do next

Note When configuring bridged interworking, the PE2 router configuration does not include the **interworking ethernet**command because it is treated as like-to-like, and also because the AC is already an Ethernet port. However, when configuring routed interworking, the PE2 router configuration does include the **interworking ip** command.

Frame Relay DLCI-to-Ethernet Port on a PE2 router using the commands associated with the L2VPN Protocol-Based CLIs feature

You can configure the Frame Relay DLCI-to-Ethernet Port feature on a PE2 router using the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **template type pseudowire** [*pseudowire-name*]
7. **encapsulation mpls**
8. **interworking ethernet**
9. **interface** *type slot / subslot / port*
10. **end**
11. **interface pseudowire** *number*
12. **source template type pseudowire** *template-name*
13. **neighbor** *peer-address vcid-value*
14. **exit**
15. **l2vpn xconnect context** *context-name*
16. **member pseudowire** *interface-number*
17. **member** *ip-address vc-id* **encapsulation mpls**
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**template type pseudowire** [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire atm-eth`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking ethernet** **Example:** `Router(config-pw)# interworking ethernet`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface gigabitethernet 2/0/0`	Configures an interface and enters interface configuration mode.
Step 10	**end** **Example:** `Router(config-pw)# end`	Exits to privileged EXEC mode.
Step 11	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.

	Command or Action	Purpose
Step 12	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire atm-eth`	Configures the source template of type pseudowire named atm-eth
Step 13	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.200 140`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 15	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 16	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 17	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.200 140 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 18	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

What to do next

Note When configuring bridged interworking, the PE2 router configuration does not include the **interworking ethernet**command because it is treated as like-to-like, and also because the AC is already an Ethernet port. However, when configuring routed interworking, the PE2 router configuration does include the **interworking ip** command.

Frame Relay DLCI-to-Ethernet VLAN 802.1Q on a PE1 Router

To configure the Frame Relay DLCI-to-Ethernet VLAN 802.1Q feature on a PE1 router, use the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **frame-relay switching**
10. **interface** *type slot / subslot / port*
11. **encapsulation frame-relay**
12. **frame-relay intf-type** [**dce**]
13. **connect** *connection-name interface dlci* {*interface dlci* | **l2transport**}
14. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:**	Sets the primary or secondary IP address for an interface.

	Command or Action	Purpose
	`Router(config-if)# ip address 10.0.0.100 255.255.255.255`	
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**frame-relay switching** **Example:** `Router(config-pw)# frame-relay switching`	Enables PVC switching on a Frame Relay DCE device.
Step 10	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface serial 2/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**encapsulation frame-relay** **Example:** `Router(config-if)# encapsulation frame-relay`	Enables Frame Relay encapsulation.
Step 12	**frame-relay intf-type** [**dce**] **Example:** `Router(config-if)# frame-relay intf-type dce`	Configures a Frame Relay switch type.
Step 13	**connect** *connection-name interface dlci* {*interface dlci* | **l2transport**} **Example:** `Router(config-if)# connect one serial0 16 serial1 100`	Defines the connection between Frame Relay PVCs.
Step 14	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:**	Binds an AC to a pseudowire and configures an AToM static pseudowire.

	Command or Action	Purpose
	`Router(config-if)# xconnect 10.0.0.200 140 pw-class atm-eth`	
Step 15	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

Frame Relay DLCI-to-Ethernet VLAN 802.1Q on a PE1 Router using the commands associated with the L2VPN Protocol-Based CLIs feature

To configure the Frame Relay DLCI-to-Ethernet VLAN 802.1Q feature on a PE1 router, use the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **template type pseudowire** [*pseudowire-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **frame-relay switching**
10. **interface** *type slot* / *subslot* / *port*
11. **encapsulation frame-relay**
12. **frame-relay intf-type** [**dce**]
13. **connect** *connection-name interface dlci* {*interface dlci* | **l2transport**}
14. **end**
15. **interface pseudowire** *number*
16. **source template type pseudowire** *template-name*
17. **neighbor** *peer-address vcid-value*
18. **exit**
19. **l2vpn xconnect context** *context-name*
20. **member pseudowire** *interface-number*
21. **member** *ip-address vc-id* **encapsulation mpls**
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**template type pseudowire** [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire atm-eth`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**frame-relay switching** **Example:** `Router(config-pw)# frame-relay switching`	Enables PVC switching on a Frame Relay DCE device.
Step 10	**interface** *type slot / subslot / port* **Example:** `Router(config-pw)# interface serial 2/0/0`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 11	**encapsulation frame-relay** **Example:** `Router(config-if)# encapsulation frame-relay`	Enables Frame Relay encapsulation.
Step 12	**frame-relay intf-type** [**dce**] **Example:** `Router(config-if)# frame-relay intf-type dce`	Configures a Frame Relay switch type.
Step 13	**connect** *connection-name interface dlci* {*interface dlci* | **l2transport**} **Example:** `Router(config-if)# connect one serial0 16 serial1 100`	Defines the connection between Frame Relay PVCs.
Step 14	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 15	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 16	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire atm-eth`	Configures the source template of type pseudowire named atm-eth
Step 17	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)#  neighbor 10.0.0.200 140`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 18	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 19	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.

	Command or Action	Purpose
Step 20	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 21	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.200 140 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 22	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

Frame Relay DLCI-to-Ethernet VLAN 802.1Q on a PE2 Router

To configure the Frame Relay DLCI-to-Ethernet VLAN 802.1Q feature on a PE2 router, use the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface** *type slot* / *subslot* / *port* **.** *subinterface-number*
10. **encapsulation dot1q** *vlan-id*
11. **xconnect** *ip-address vc-id* **pw-class** *pw-class-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.
Step 9	**interface** *type slot / subslot / port . subinterface-number* **Example:** `Router(config-pw)# interface gigabitethernet 5/1/0.3`	Configures an interface and enters interface configuration mode.
Step 10	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if)# encapsulation dot1q 1525`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 11	**xconnect** *ip-address vc-id* **pw-class** *pw-class-name* **Example:**	Binds an AC to a pseudowire and configures an AToM static pseudowire.

	Command or Action	Purpose
	`Router(config-if)# xconnect 10.0.0.100 140 pw-class atm-eth`	
Step 12	**end** **Example:** `Router(config-if-xconn)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

What to do next

Note In the case of an Frame Relay DLCI-to-VLAN, the PE2 router configuration includes the **interworking**command for both bridged and routed interworking.

Note To verify the L2VPN interworking status and check the statistics, refer to the Verifying L2VPN Interworking, on page 962.

Frame Relay DLCI-to-Ethernet VLAN 802.1Q on a PE2 Router using the commands associated with the L2VPN Protocol-Based CLIs feature

To configure the Frame Relay DLCI-to-Ethernet VLAN 802.1Q feature on a PE2 router, use the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **pseudowire-class** [*pw-class-name*]
7. **encapsulation mpls**
8. **interworking** {**ethernet** | **ip**}
9. **interface** *type slot / subslot / port . subinterface-number*
10. **encapsulation dot1q** *vlan-id*
11. **end**
12. **interface pseudowire** *number*
13. **source template type pseudowire** *template-name*
14. **exit**
15. **l2vpn xconnect context** *context-name*
16. **member pseudowire** *interface-number*
17. **member** *ip-address vc-id* **encapsulation mpls**

18. **interworking ip**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Establishes the label distribution protocol for the platform.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 100`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.100 255.255.255.255`	Sets the primary or secondary IP address for an interface.
Step 6	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config-if)# pseudowire-class atm-eth`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 7	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 8	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw)# interworking ip`	Specifies the type of pseudowire and the type of traffic that can flow across it.

	Command or Action	Purpose
Step 9	**interface** *type slot / subslot / port . subinterface-number* **Example:** `Router(config-pw)# interface gigabitethernet 5/1/0.3`	Configures an interface and enters interface configuration mode.
Step 10	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if)# encapsulation dot1q 1525`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 11	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 12	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 13	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire ether-pw`	Configures the source template of type pseudowire named ether-pw.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.
Step 15	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 16	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 17	**member** *ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-xconnect)# member 10.0.0.100 140 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

	Command or Action	Purpose
Step 18	**interworking ip** **Example:** `Router(config-xconnect)# interworking ip`	Establishes an L2VPN cross connect context.
Step 19	**end** **Example:** `Router(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

What to do next

Note In the case of an Frame Relay DLCI-to-VLAN, the PE2 router configuration includes the **interworking**command for both bridged and routed interworking.

Note To verify the L2VPN interworking status and check the statistics, refer to the Verifying L2VPN Interworking, on page 962.

Configuring HDLC-to-Ethernet Interworking

HDLC-to-Ethernet Bridged Interworking on a HDLC PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **interworking ethernet**
6. **interface** *type slot/subslot /port* [**.** *subinterface*]
7. **no ip address** [*ip-address mask*] [**secondary**]
8. **xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class pw-iw-ether`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**interworking ethernet** **Example:** `Device(config-pw-class)# interworking ethernet`	Specifies Ethernet as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 6	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config-pw-class)# interface serial 3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 7	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 8	**xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*] **Example:** `Device(config-if)# xconnect 198.51.100.2 123 pseudowire-class pw-iw-ether`	Creates the virtual circuit (VC) to transport the Layer 2 packets.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Bridged Interworking on a HDLC PE Device Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **template type pseudowire** *name*
4. **encapsulation mpls**
5. **exit**
6. **interface pseudowire** *number*
7. **source template type pseudowire** *name*
8. **encapsulation mpls**
9. **neighbor** *peer-address vc id-value*
10. **signaling protocol ldp**
11. **no shutdown**
12. **exit**
13. **l2vpn xconnect context** *context-name*
14. **interworking ethernet**
15. **member** *interface-type-number*
16. **member pseudowire** *interface-number*
17. **no shutdown**
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** *name* **Example:** `Device# template type pseudowire temp5`	Creates a template pseudowire with a name that you specify and enters template configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-template)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 6	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 107`	Establishes an interface pseudowire with a value that you specify and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire temp5`	Configures the source template of type pseudowire named temp5.
Step 8	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 9	**neighbor** *peer-address vc id-value* **Example:** `Device(config-if)# neighbor 10.0.0.11 107`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 10	**signaling protocol ldp** **Example:** `Device(config-if)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.
Step 11	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the interface pseudowire.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con1`	Creates an L2VPN cross-connect context and enters xconnect configuration mode.
Step 14	**interworking ethernet** **Example:** `Device(config-xconnect)# interworking ethernet`	Specifies Ethernet as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 15	**member** *interface-type-number* **Example:** `Device(config-xconnect)# member serial 0/1/0:0`	Specifies the location of the member interface.
Step 16	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 107`	Specifies a member pseudowire to form an L2VPN cross connect.
Step 17	**no shutdown** **Example:** `Device(config-xconnect)# no shutdown`	Restarts the member interface.

	Command or Action	Purpose
Step 18	**end** **Example:** `Device(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Bridged Interworking (Port Mode) on an Ethernet PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **interworking ethernet**
6. **interface** *type slot/subslot /port* [**.** *subinterface*]
7. **encapsulation mpls**
8. **xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class pw-iw-ether`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**interworking ethernet** **Example:** `Device(config-pw-class)# interworking ethernet`	Specifies Ethernet as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.

	Command or Action	Purpose
Step 6	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config-pw-class)# interface gigabitethernet 4/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.
Step 7	**encapsulation mpls** **Example:** `Device(config-subif)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 8	**xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*] **Example:** `Device(config-subif)# xconnect 198.51.100.2 123 pseudowire-class pw-iw-ether`	Creates the virtual circuit (VC) to transport the Layer 2 packets.
Step 9	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Bridged Interworking (Port Mode) on an Ethernet PE Device Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot /port* [**.** *subinterface*]
4. **encapsulation mpls**
5. **no ip address**
6. **no shutdown**
7. **exit**
8. **template type pseudowire** *name*
9. **encapsulation mpls**
10. **exit**
11. **interface pseudowire** *number*
12. **source template type pseudowire** *name*
13. **encapsulation mpls**
14. **neighbor** *peer-address vc id-value*
15. **signaling protocol ldp**
16. **no shutdown**
17. **exit**

18. **l2vpn xconnect context** *context-name*
19. **interworking ethernet**
20. **member** *interface-type-number*
21. **member pseudowire** *interface-number*
22. **no shutdown**
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config)# interface fastethernet 4/0/0.1`	Specifies the subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.
Step 4	**encapsulation mpls** **Example:** `Device(config-subif)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**no ip address** **Example:** `Device(config-subif)# no ip address`	Disables IP processing.
Step 6	**no shutdown** **Example:** `Device(config-subif)# no shutdown`	Restarts the Fast Ethernet subinterface.
Step 7	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.
Step 8	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire temp4`	Creates a template pseudowire with a name that you specify and enters template configuration mode.

	Command or Action	Purpose
Step 9	**encapsulation mpls** **Example:** `Device(config-template)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 10	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 11	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 109`	Establishes an interface pseudowire with a value that you specify and enters interface configuration mode.
Step 12	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire temp4`	Configures the source template of type pseudowire named temp4.
Step 13	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 14	**neighbor** *peer-address vc id-value* **Example:** `Device(config-if)# neighbor 10.0.0.15 109`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 15	**signaling protocol ldp** **Example:** `Device(config-if)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.
Step 16	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the interface pseudowire.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con2`	Creates an L2VPN cross-connect context and enters xconnect configuration mode.
Step 19	**interworking ethernet** **Example:**	Specifies Ethernet as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.

	Command or Action	Purpose
	`Device(config-xconnect)# interworking ethernet`	
Step 20	**member** *interface-type-number* **Example:** `Device(config-xconnect)# member fastethernet 4/0/0.1`	Specifies the location of the member interface.
Step 21	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 109`	Specifies a member pseudowire to form an L2VPN cross connect.
Step 22	**no shutdown** **Example:** `Device(config-xconnect)# no shutdown`	Restarts the member interface.
Step 23	**end** **Example:** `Device(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Bridged Interworking (dot1q and QinQ Modes) on an Ethernet PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **interworking ethernet**
6. **interface** *type slot/subslot /port* [**.** *subinterface*]
7. **encapsulation dot1q** *vlan-id***second dot1q** *vlan-id*
8. **xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class pw-iw-ether`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**interworking ethernet** **Example:** `Device(config-pw-class)# interworking ethernet`	Specifies Ethernet as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 6	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config-pw-class)# interface gigabitethernet 4/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.
Step 7	**encapsulation dot1q** *vlan-id***second dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100 second dot1q 200`	Defines the matching criteria to map QinQ ingress frames on an interface to the appropriate service instance.
Step 8	**xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*] **Example:** `Device(config-subif)# xconnect 198.51.100.2 123 pseudowire-class pw-iw-ether`	Creates the virtual circuit (VC) to transport the Layer 2 packets.
Step 9	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Bridged Interworking (dot1q and QinQ Modes) on an Ethernet PE Device Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot /port* [**.** *subinterface*]

4. **encapsulation dot1q** *vlan-id* **second dot1q** *vlan-id*
5. **no ip address**
6. **no shutdown**
7. **exit**
8. **template type pseudowire** *name*
9. **encapsulation mpls**
10. **exit**
11. **interface pseudowire** *number*
12. **source template type pseudowire** *name*
13. **encapsulation mpls**
14. **neighbor** *peer-address vc id-value*
15. **signaling protocol ldp**
16. **no shutdown**
17. **exit**
18. **l2vpn xconnect context** *context-name*
19. **interworking ethernet**
20. **member** *interface-type-number*
21. **member pseudowire** *interface-number*
22. **no shutdown**
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config)# interface fastethernet 4/0/0.1`	Specifies the subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.
Step 4	**encapsulation dot1q** *vlan-id* **second dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100 second dot1q 200`	Defines the matching criteria to map QinQ ingress frames on an interface to the appropriate service instance.
Step 5	**no ip address** **Example:**	Disables IP processing.

	Command or Action	Purpose
	`Device(config-subif)# no ip address`	
Step 6	**no shutdown** **Example:** `Device(config-subif)# no shutdown`	Restarts the Fast Ethernet subinterface.
Step 7	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.
Step 8	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire temp4`	Creates a template pseudowire with a name that you specify and enters template configuration mode.
Step 9	**encapsulation mpls** **Example:** `Device(config-template)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 10	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 11	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 109`	Establishes an interface pseudowire with a value that you specify and enters interface configuration mode.
Step 12	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire temp4`	Configures the source template of type pseudowire named temp4.
Step 13	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 14	**neighbor** *peer-address vc id-value* **Example:** `Device(config-if)# neighbor 10.0.0.15 109`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 15	**signaling protocol ldp** **Example:** `Device(config-if)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.

	Command or Action	Purpose
Step 16	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the interface pseudowire.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con2`	Creates an L2VPN cross-connect context and enters xconnect configuration mode.
Step 19	**interworking ethernet** **Example:** `Device(config-xconnect)# interworking ethernet`	Specifies Ethernet as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 20	**member** *interface-type-number* **Example:** `Device(config-xconnect)# member fastethernet 4/0/0.1`	Specifies the location of the member interface.
Step 21	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 109`	Specifies a member pseudowire to form an L2VPN cross connect.
Step 22	**no shutdown** **Example:** `Device(config-xconnect)# no shutdown`	Restarts the member interface.
Step 23	**end** **Example:** `Device(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Routed Interworking on a HDLC PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **interworking ip**
6. **interface** *type slot/subslot /port* [**.** *subinterface*]
7. **no ip address** [*ip-address mask*] [**secondary**]

8. **xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class pw-iw-ip`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**interworking ip** **Example:** `Device(config-pw-class)# interworking ip`	Specifies IP as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 6	**interface** *type slot*/*subslot* /*port* [**.** *subinterface*] **Example:** `Device(config-pw-class)# interface serial 3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 7	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 8	**xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*] **Example:** `Device(config-if)# xconnect 198.51.100.2 123 pseudowire-class pw-iw-ip`	Creates the virtual circuit (VC) to transport the Layer 2 packets.
Step 9	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

HDLC-to-Ethernet Routed Interworking on a HDLC PE Device Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** *name*
4. **encapsulation mpls**
5. **exit**
6. **interface pseudowire** *number*
7. **source template type pseudowire** *name*
8. **encapsulation mpls**
9. **neighbor** *peer-address vc id-value*
10. **signaling protocol ldp**
11. **no shutdown**
12. **exit**
13. **l2vpn xconnect context** *context-name*
14. **interworking ip**
15. **member** *interface-type-number*
16. **member pseudowire** *interface-number*
17. **no shutdown**
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** *name* **Example:** `Device# template type pseudowire temp5`	Creates a template pseudowire with a name that you specify and enters template configuration mode.
Step 4	**encapsulation mpls** **Example:**	Specifies the tunneling encapsulation as MPLS.

	Command or Action	Purpose
	`Device(config-template)# encapsulation mpls`	
Step 5	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 6	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 107`	Establishes an interface pseudowire with a value that you specify and enters interface configuration mode.
Step 7	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire temp5`	Configures the source template of type pseudowire named temp5.
Step 8	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 9	**neighbor** *peer-address vc id-value* **Example:** `Device(config-if)# neighbor 10.0.0.11 107`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 10	**signaling protocol ldp** **Example:** `Device(config-if)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.
Step 11	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the interface pseudowire.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con1`	Creates an L2VPN cross-connect context and enters xconnect configuration mode.
Step 14	**interworking ip** **Example:** `Device(config-xconnect)# interworking ip`	Specifies IP as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.

	Command or Action	Purpose
Step 15	**member** *interface-type-number* **Example:** `Device(config-xconnect)# member serial 0/1/0:0`	Specifies the location of the member interface.
Step 16	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 107`	Specifies a member pseudowire to form an L2VPN cross connect.
Step 17	**no shutdown** **Example:** `Device(config-xconnect)# no shutdown`	Restarts the member interface.
Step 18	**end** **Example:** `Device(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Routed Interworking (Port Mode) on an Ethernet PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **interworking ip**
6. **interface** *type slot/subslot /port* [**.** *subinterface*]
7. **encapsulation mpls**
8. **xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:**	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.

	Command or Action	Purpose
	`Device(config)# pseudowire-class pw-iw-ip`	
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**interworking ip** **Example:** `Device(config-pw-class)# interworking ip`	Specifies IP as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 6	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config-pw-class)# interface gigabitethernet 4/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.
Step 7	**encapsulation mpls** **Example:** `Device(config-subif)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 8	**xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*] **Example:** `Device(config-subif)# xconnect 198.51.100.2 123 pseudowire-class pw-iw-ip`	Creates the virtual circuit (VC) to transport the Layer 2 packets.
Step 9	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Routed Interworking (Port Mode) on an Ethernet PE Device Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot /port* [**.** *subinterface*]
4. **encapsulation mpls**
5. **no ip address**
6. **no shutdown**

7. **exit**
8. **template type pseudowire** *name*
9. **encapsulation mpls**
10. **exit**
11. **interface pseudowire** *number*
12. **source template type pseudowire** *name*
13. **encapsulation mpls**
14. **neighbor** *peer-address vc id-value*
15. **signaling protocol ldp**
16. **no shutdown**
17. **exit**
18. **l2vpn xconnect context** *context-name*
19. **interworking ip**
20. **member** *interface-type-number*
21. **member pseudowire** *interface-number*
22. **no shutdown**
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config)# interface fastethernet 4/0/0.1`	Specifies the Fast Ethernet subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.
Step 4	**encapsulation mpls** **Example:** `Device(config-subif)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**no ip address** **Example:** `Device(config-subif)# no ip address`	Disables IP processing.

	Command or Action	Purpose
Step 6	**no shutdown** **Example:** `Device(config-subif)# no shutdown`	Restarts the Fast Ethernet subinterface.
Step 7	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.
Step 8	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire temp4`	Creates a template pseudowire with a name that you specify and enters template configuration mode.
Step 9	**encapsulation mpls** **Example:** `Device(config-template)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 10	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 11	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 109`	Establishes an interface pseudowire with a value that you specify and enters interface configuration mode.
Step 12	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire temp4`	Configures the source template of type pseudowire named temp4.
Step 13	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 14	**neighbor** *peer-address vc id-value* **Example:** `Device(config-if)# neighbor 10.0.0.15 109`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 15	**signaling protocol ldp** **Example:** `Device(config-if)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.
Step 16	**no shutdown** **Example:**	Restarts the interface pseudowire.

	Command or Action	Purpose
	`Device(config-if)# no shutdown`	
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con2`	Creates an L2VPN cross-connect context and enters xconnect configuration mode.
Step 19	**interworking ip** **Example:** `Device(config-xconnect)# interworking ip`	Specifies IP as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 20	**member** *interface-type-number* **Example:** `Device(config-xconnect)# member fastethernet 4/0/0.1`	Specifies the location of the member interface.
Step 21	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 109`	Specifies a member pseudowire to form an L2VPN cross connect.
Step 22	**no shutdown** **Example:** `Device(config-xconnect)# no shutdown`	Restarts the member interface.
Step 23	**end** **Example:** `Device(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Routed Interworking (dot1q and QinQ Modes) on an Ethernet PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **interworking ip**
6. **interface** *type slot/subslot /port* [**.** *subinterface*]
7. **encapsulation dot1q** *vlan-id* **second dot1q** *vlan-id*
8. **xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class pw-iw-ip`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 5	**interworking ip** **Example:** `Device(config-pw-class)# interworking ip`	Specifies IP as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 6	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config-pw-class)# interface gigabitethernet 4/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.
Step 7	**encapsulation dot1q** *vlan-id* **second dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100 second dot1q 200`	Defines the matching criteria to map QinQ ingress frames on an interface to the appropriate service instance.
Step 8	**xconnect** *peer-router-id vc id* **pseudowire-class** [*pw-class-name*] **Example:** `Device(config-subif)# xconnect 198.51.100.2 123 pseudowire-class pw-iw-ip`	Creates the virtual circuit (VC) to transport the Layer 2 packets.
Step 9	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

HDLC-to-Ethernet Routed Interworking (dot1q and QinQ Modes) on an Ethernet PE Device Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot /port* [**.** *subinterface*]
4. **encapsulation dot1q** *vlan-id* **second dot1q** *vlan-id*
5. **no ip address**
6. **no shutdown**
7. **exit**
8. **template type pseudowire** *name*
9. **encapsulation mpls**
10. **exit**
11. **interface pseudowire** *number*
12. **source template type pseudowire** *name*
13. **encapsulation mpls**
14. **neighbor** *peer-address vc id-value*
15. **signaling protocol ldp**
16. **no shutdown**
17. **exit**
18. **l2vpn xconnect context** *context-name*
19. **interworking ip**
20. **member** *interface-type-number*
21. **member pseudowire** *interface-number*
22. **no shutdown**
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config)# interface fastethernet 4/0/0.1`	Specifies the subinterface and enters subinterface configuration mode. • Ensure that the subinterface on the adjoining Ethernet CE device is on the same VLAN as this Ethernet PE device.

	Command or Action	Purpose
Step 4	**encapsulation dot1q** *vlan-id* **second dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100 second dot1q 200`	Defines the matching criteria to map QinQ ingress frames on an interface to the appropriate service instance.
Step 5	**no ip address** **Example:** `Device(config-subif)# no ip address`	Disables IP processing.
Step 6	**no shutdown** **Example:** `Device(config-subif)# no shutdown`	Restarts the Fast Ethernet subinterface.
Step 7	**exit** **Example:** `Device(config-subif)# exit`	Exits subinterface configuration mode and returns to global configuration mode.
Step 8	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire temp4`	Creates a template pseudowire with a name that you specify and enters template configuration mode.
Step 9	**encapsulation mpls** **Example:** `Device(config-template)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 10	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 11	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 109`	Establishes an interface pseudowire with a value that you specify and enters interface configuration mode.
Step 12	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire temp4`	Configures the source template of type pseudowire named temp4.
Step 13	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.

	Command or Action	Purpose
Step 14	**neighbor** *peer-address vc id-value* **Example:** `Device(config-if)# neighbor 10.0.0.15 109`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 15	**signaling protocol ldp** **Example:** `Device(config-if)# signaling protocol ldp`	Specifies that the Label Distribution Protocol (LDP) is configured for the pseudowire class.
Step 16	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts the interface pseudowire.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con2`	Creates an L2VPN cross-connect context and enters xconnect configuration mode.
Step 19	**interworking ip** **Example:** `Device(config-xconnect)# interworking ip`	Specifies IP as the type of pseudowire as well as the type of traffic that can flow across the pseudowire.
Step 20	**member** *interface-type-number* **Example:** `Device(config-xconnect)# member fastethernet 4/0/0.1`	Specifies the location of the member interface.
Step 21	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 109`	Specifies a member pseudowire to form an L2VPN cross connect.
Step 22	**no shutdown** **Example:** `Device(config-xconnect)# no shutdown`	Restarts the member interface.
Step 23	**end** **Example:** `Device(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

Verifying HDLC-to-Ethernet Interworking (Port Mode) Configuration on a HDLC PE Device

You can use **show** commands to view information about a HDLC-to-Ethernet interworking (port mode) configuration on a HDLC provider edge (PE) device.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show mpls l2transport vc detail**
3. **show l2vpn atom vc**
4. **show l2vpn atom vc detail**

DETAILED STEPS

Step 1 show mpls l2transport vc

The following is sample output from the **show mpls l2transport vc** command which displays basic information about HDLC-to-Ethernet interworking (port mode) configuration on a HDLC PE device:

Example:

```
Device# show mpls l2transport vc

Local intf  Local circuit  Dest address    VC ID      Status
----------- -------------- --------------- ---------- ----------
Se0/1/0:0   HDLC           10.0.0.1        101        UP
```

Step 2 show mpls l2transport vc detail

The following is sample output from the **show mpls l2transport vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (port mode) configuration on a HDLC PE device:

Example:

```
Device# show mpls l2transport vc detail

Local interface: Se0/1/0:0 up, line protocol up, HDLC up
Interworking type is Ethernet
Destination address: 10.0.0.1, VC ID: 101, VC status: up
Output interface: Fa0/0/1, imposed label stack {20 22}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.10
Create time: 00:00:19, last status change time: 00:00:15
Last label FSM state change time: 00:00:15
Signaling protocol: LDP, peer 10.0.0.1:0 up
Targeted Hello: 203.0.113.1(LDP Id) -> 10.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
```

```
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 33, remote 22
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description: Connect to CE2
Sequencing: receive disabled, send disabled
Control Word: On
SSO Descriptor: 10.0.0.1/101, local label: 33
Dataplane:
SSM segment/switch IDs: 4274/4273 (used), PWID: 26
VC statistics:
transit packet totals: receive 3, send 6
transit byte totals: receive 162, send 366
transit packet drops: receive 0, seq error 0, send 0
```

Step 3 **show l2vpn atom vc**

The following is sample output from the **show l2vpn atom vc** command which displays basic information about HDLC-to-Ethernet interworking (port mode) configuration on a HDLC PE device:

Example:

```
Device# show l2vpn atom vc

                            Service
Interface Peer ID    VC ID  Type   Name  Status
--------- ---------- ------ ------ ----- ----------
pw101     10.0.0.1   101    p2p    101   UP
```

Step 4 **show l2vpn atom vc detail**

The following is sample output from the **show l2vpn atom vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (port mode) configuration on a HDLC PE device:

Example:

```
Device# show l2vpn atom vc detail

pseudowire101 is up, VC status is up PW type: Ethernet
Create time: 00:00:18, last status change time: 00:00:14
Last label FSM state change time: 00:00:14
Destination address: 10.0.0.1 VC ID: 101
Output interface: Fa0/0/1, imposed label stack {16 17}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.10
Member of xconnect service hdlc101
Associated member Se0/1/0:0 is up, status is up
Interworking type is Ethernet
Service id: 0xde000002
Signaling protocol: LDP, peer 10.0.0.1:0 up
Targeted Hello: 203.0.113.1(LDP Id) -> 10.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 101
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Local dataplane status received : No fault
BFD dataplane status received : Not sent
BFD peer monitor status received : No fault
Status received from access circuit : No fault
Status sent to access circuit : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer : No fault
```

```
Status received from network peer : No fault
Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        18                             17
Group ID     0                              0
Interface    Connect to CE1                 Connect to CE2
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x02                           0x02
             LSPV [2]                       LSPV [2]
VCCV CC type 0x07                           0x07
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
SSO Descriptor: 10.0.0.1/101, local label: 18
Dataplane:
SSM segment/switch IDs: 4106/4105 (used), PWID: 2
Rx Counters
3 input transit packets, 162 bytes
0 drops, 0 seq err
Tx Counters
5 output transit packets, 305 bytes
0 drops
```

Verifying HDLC-to-Ethernet Interworking (Port Mode) Configuration on an Ethernet PE Device

You can use **show** commands to view information about a HDLC-to-Ethernet interworking (port mode) configuration on an Ethernet PE device.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show l2vpn atom vc**
3. **show l2vpn atom vc detail**

DETAILED STEPS

Step 1 show mpls l2transport vc

The following is sample output from the **show mpls l2transport vc** command which displays basic information about HDLC-to-Ethernet interworking (port mode) configuration on an Ethernet PE device:

Example:

```
Device# show mpls l2transport vc

Local interface: Gi1/0/0 up, line protocol up, Ethernet up
Destination address: 203.0.113.1, VC ID: 101, VC status: up
Output interface: Fa0/0/1, imposed label stack {19 33}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.11
Create time: 00:00:22, last status change time: 00:00:19
Last label FSM state change time: 00:00:19
```

```
Signaling protocol: LDP, peer 203.0.113.1:0 up
Targeted Hello: 10.0.0.1(LDP Id) -> 203.0.113.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 22, remote 33
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description: Connect to CE1
Sequencing: receive disabled, send disabled
Control Word: On
SSO Descriptor: 203.0.113.1/101, local label: 22
Dataplane:
SSM segment/switch IDs: 4574/4573 (used), PWID: 80
VC statistics:
transit packet totals: receive 9, send 5
transit byte totals: receive 315, send 380
transit packet drops: receive 0, seq error 0, send 0
```

Step 2 **show l2vpn atom vc**

The following is sample output from the **show l2vpn atom vc** command which displays basic information about HDLC-to-Ethernet interworking (port mode) configuration on an Ethernet PE device:

Example:

```
Device# show l2vpn atom vc

                            Service
Interface Peer ID    VC ID  Type   Name  Status
--------- ---------- ------ ------ ----- ----------
pw101     10.0.0.1   101    p2p    101   UP
```

Step 3 **show l2vpn atom vc detail**

The following is sample output from the **show l2vpn atom vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (port mode) configuration on an Ethernet PE device:

Example:

```
Device# show l2vpn atom vc detail

pseudowire101 is up, VC status is up PW type: Ethernet
Create time: 00:00:18, last status change time: 00:00:14
Last label FSM state change time: 00:00:14
Destination address: 10.0.0.1 VC ID: 101
Output interface: Fa0/0/1, imposed label stack {16 17}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.10
Member of xconnect service eth101
Associated member Se0/1/0:0 is up, status is up
Interworking type is Ethernet
Service id: 0xde000002
```

```
Signaling protocol: LDP, peer 10.0.0.1:0 up
Targeted Hello: 203.0.113.1(LDP Id) -> 10.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 101
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Local dataplane status received : No fault
BFD dataplane status received : Not sent
BFD peer monitor status received : No fault
Status received from access circuit : No fault
Status sent to access circuit : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer : No fault
Status received from network peer : No fault
Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        18                             17
Group ID     0                              0
Interface    Connect to CE1                 Connect to CE2
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x02                           0x02
             LSPV [2]                       LSPV [2]
VCCV CC type 0x07                           0x07
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
SSO Descriptor: 10.0.0.1/101, local label: 18
Dataplane:
SSM segment/switch IDs: 4106/4105 (used), PWID: 2
Rx Counters
3 input transit packets, 162 bytes
0 drops, 0 seq err
Tx Counters
5 output transit packets, 305 bytes
0 drops
```

Verifying HDLC-to-Ethernet Interworking (dot1q Mode) Configuration on a HDLC PE Device

You can use **show** commands to view information about a HDLC-to-Ethernet interworking (dot1q mode) configuration on a HDLC PE device.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show mpls l2transport vc detail**
3. **show l2vpn atom vc**
4. **show l2vpn atom vc detail**

DETAILED STEPS

Step 1 show mpls l2transport vc

The following is sample output from the **show mpls l2transport vc** command which displays basic information about HDLC-to-Ethernet interworking (dot1q mode) configuration on a HDLC PE device:

Example:

```
Device# show mpls l2transport vc

Local intf  Local circuit  Dest address    VC ID      Status
----------- -------------- --------------- ---------- ----------
Se0/1/0:0   HDLC           10.0.0.1       101        UP
```

Step 2 show mpls l2transport vc detail

The following is sample output from the **show mpls l2transport vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (dot1q mode) configuration on a HDLC PE device:

Example:

```
Device# show mpls l2transport vc detail

Local interface: Se0/1/0:0 up, line protocol up, HDLC up
Interworking type is Ethernet
Destination address: 10.0.0.1, VC ID: 101, VC status: up
Output interface: Fa0/0/1, imposed label stack {20 22}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.10
Create time: 00:00:19, last status change time: 00:00:15
Last label FSM state change time: 00:00:15
Signaling protocol: LDP, peer 10.0.0.1:0 up
Targeted Hello: 203.0.113.1(LDP Id) -> 10.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 33, remote 22
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description: Connect to CE2
Sequencing: receive disabled, send disabled
Control Word: On
SSO Descriptor: 10.0.0.1/101, local label: 33
Dataplane:
SSM segment/switch IDs: 4274/4273 (used), PWID: 26
VC statistics:
transit packet totals: receive 3, send 6
transit byte totals: receive 162, send 366
transit packet drops: receive 0, seq error 0, send 0
```

Step 3 **show l2vpn atom vc**

The following is sample output from the **show l2vpn atom vc** command which displays basic information about HDLC-to-Ethernet interworking (dot1q mode) configuration on a HDLC PE device:

Example:

```
Device# show l2vpn atom vc

                            Service
Interface Peer ID    VC ID  Type   Name  Status
--------- ---------- ------ ------ ----- ----------
pw101     10.0.0.1  101    p2p    101   UP
```

Step 4 **show l2vpn atom vc detail**

The following is sample output from the **show l2vpn atom vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (dot1q mode) configuration on a HDLC PE device:

Example:

```
Device# show l2vpn atom vc detail

pseudowire101 is up, VC status is up PW type: Ethernet
Create time: 00:00:18, last status change time: 00:00:14
Last label FSM state change time: 00:00:14
Destination address: 10.0.0.1 VC ID: 101
Output interface: Fa0/0/1, imposed label stack {16 17}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.10
Member of xconnect service hdlc101
Associated member Se0/1/0:0 is up, status is up
Interworking type is Ethernet
Service id: 0xde000002
Signaling protocol: LDP, peer 10.0.0.1:0 up
Targeted Hello: 203.0.113.1(LDP Id) -> 10.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 101
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Local dataplane status received : No fault
BFD dataplane status received : Not sent
BFD peer monitor status received : No fault
Status received from access circuit : No fault
Status sent to access circuit : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer : No fault
Status received from network peer : No fault
Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        18                             17
Group ID     0                              0
Interface    Connect to CE1                 Connect to CE2
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x02                           0x02
             LSPV [2]                       LSPV [2]
VCCV CC type 0x07                           0x07
```

```
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
SSO Descriptor: 10.0.0.1/101, local label: 18
Dataplane:
SSM segment/switch IDs: 4106/4105 (used), PWID: 2
Rx Counters
3 input transit packets, 162 bytes
0 drops, 0 seq err
Tx Counters
5 output transit packets, 305 bytes
0 drops
```

Verifying HDLC-to-Ethernet Interworking (dot1q Mode) Configuration on an Ethernet PE Device

You can use **show** commands to view information about a HDLC-to-Ethernet interworking (dot1q mode) configuration on an Ethernet PE device.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show mpls l2transport vc detail**
3. **show l2vpn atom vc**
4. **show l2vpn atom vc detail**

DETAILED STEPS

Step 1 **show mpls l2transport vc**

The following is sample output from the **show mpls l2transport vc** command which displays basic information about HDLC-to-Ethernet interworking (dot1q mode) configuration on an Ethernet PE device:

Example:

```
Device# show mpls l2transport vc

Local intf  Local circuit  Dest address    VC ID      Status
----------- -------------- --------------- ---------- ----------
Gi1/0/0.10  Eth VLAN 10    203.0.113.1     138        UP
```

Step 2 **show mpls l2transport vc detail**

The following is sample output from the **show mpls l2transport vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (dot1q mode) configuration on an Ethernet PE device:

Example:

```
Device# show mpls l2transport vc detail

Local interface: Gi1/0/0.10 up, line protocol up, Eth VLAN 10 up
Interworking type is Ethernet
Destination address: 203.0.113.1, VC ID: 138, VC status: up
Output interface: Fa0/0/1, imposed label stack {19 35}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.11
Create time: 00:00:22, last status change time: 00:00:20
Last label FSM state change time: 00:00:20
```

```
Signaling protocol: LDP, peer 203.0.113.1:0 up
Targeted Hello: 10.0.0.1(LDP Id) -> 203.0.113.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 53, remote 35
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description: Connect to CE1
Sequencing: receive disabled, send disabled
Control Word: On
SSO Descriptor: 203.0.113.1/138, local label: 53
Dataplane:
SSM segment/switch IDs: 4784/4783 (used), PWID: 117
VC statistics:
transit packet totals: receive 6, send 6
transit byte totals: receive 234, send 1276
transit packet drops: receive 0, seq error 0, send 0
```

Step 3 show l2vpn atom vc

The following is sample output from the **show l2vpn atom vc** command which displays basic information about HDLC-to-Ethernet interworking (dot1q mode) configuration on an Ethernet PE device:

Example:

```
Device# show l2vpn atom vc

                            Service
Interface Peer ID    VC ID  Type   Name  Status
--------- ---------- ------ ------ ----- ----------
pw138     203.0.113.1 138    p2p    138   UP
```

Step 4 show l2vpn atom vc detail

The following is sample output from the **show l2vpn atom vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (dot1q mode) configuration on an Ethernet PE device:

Example:

```
Device# show l2vpn atom vc detail

pseudowire138 is up, VC status is up PW type: Ethernet
Create time: 00:00:23, last status change time: 00:00:20
Last label FSM state change time: 00:00:20
Destination address: 203.0.113.1 VC ID: 138
Output interface: Fa0/0/1, imposed label stack {18 20}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.11
Member of xconnect service eth138
Associated member Gi1/0/0.10 is up, status is up
Interworking type is Ethernet
Service id: 0x7b000029
```

```
Signaling protocol: LDP, peer 203.0.113.1:0 up
Targeted Hello: 10.0.0.1(LDP Id) -> 203.0.113.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 138
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Local dataplane status received : No fault
BFD dataplane status received : Not sent
BFD peer monitor status received : No fault
Status received from access circuit : No fault
Status sent to access circuit : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer : No fault
Status received from network peer : No fault
Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        30                             20
Group ID     0                              0
Interface    Connect to CE2                 Connect to CE1
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x02                           0x02
             LSPV [2]                       LSPV [2]
VCCV CC type 0x07                           0x07
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
SSO Descriptor: 203.0.113.1/138, local label: 30
Dataplane:
SSM segment/switch IDs: 4333/4332 (used), PWID: 41
Rx Counters
8 input transit packets, 312 bytes
0 drops, 0 seq err
Tx Counters
5 output transit packets, 380 bytes
0 drops
```

Verifying HDLC-to-Ethernet Interworking (QinQ Mode) Configuration on a HDLC PE Device

You can use **show** commands to view information about a HDLC-to-Ethernet interworking (QinQ mode) configuration on a HDLC PE device.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show mpls l2transport vc detail**
3. **show l2vpn atom vc**
4. **show l2vpn atom vc detail**

DETAILED STEPS

Step 1 show mpls l2transport vc

The following is sample output from the **show mpls l2transport vc** command which displays basic information about HDLC-to-Ethernet interworking (QinQ mode) configuration on a HDLC PE device:

Example:

```
Device# show mpls l2transport vc

Local intf  Local circuit  Dest address    VC ID      Status
----------- -------------- --------------- ---------- ----------
Se0/1/0:0   HDLC           10.0.0.1        145        UP
```

Step 2 show mpls l2transport vc detail

The following is sample output from the **show mpls l2transport vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (QinQ mode) configuration on a HDLC PE device:

Example:

```
Device# show mpls l2transport vc detail

Local interface: Se0/1/0:0 up, line protocol up, HDLC up
Interworking type is Ethernet
Destination address: 10.0.0.1, VC ID: 101, VC status: up
Output interface: Fa0/0/1, imposed label stack {20 22}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.10
Create time: 00:00:19, last status change time: 00:00:15
Last label FSM state change time: 00:00:15
Signaling protocol: LDP, peer 10.0.0.1:0 up
Targeted Hello: 203.0.113.1(LDP Id) -> 10.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 33, remote 22
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description: Connect to CE2
Sequencing: receive disabled, send disabled
Control Word: On
SSO Descriptor: 10.0.0.1/101, local label: 33
Dataplane:
SSM segment/switch IDs: 4274/4273 (used), PWID: 26
VC statistics:
transit packet totals: receive 3, send 6
transit byte totals: receive 162, send 366
transit packet drops: receive 0, seq error 0, send 0
```

Step 3 **show l2vpn atom vc**

The following is sample output from the **show l2vpn atom vc** command which displays basic information about HDLC-to-Ethernet interworking (QinQ mode) configuration on a HDLC PE device:

Example:

```
Device# show l2vpn atom vc

                            Service
Interface Peer ID    VC ID  Type   Name  Status
--------- ---------- ------ ------ ----- ----------
pw145     10.0.0.1   145    p2p    145   UP
```

Step 4 **show l2vpn atom vc detail**

The following is sample output from the **show l2vpn atom vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (QinQ mode) configuration on a HDLC PE device:

Example:

```
Device# show l2vpn atom vc detail

pseudowire145 is up, VC status is up PW type: Ethernet
Create time: 00:00:18, last status change time: 00:00:13
Last label FSM state change time: 00:00:13
Destination address: 10.0.0.1 VC ID: 145
Output interface: Fa0/0/1, imposed label stack {16 33}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.10
Member of xconnect service hdlc145
Associated member Se0/1/0:0 is up, status is up
Interworking type is Ethernet
Service id: 0x2e
Signaling protocol: LDP, peer 10.0.0.1:0 up
Targeted Hello: 203.0.113.1(LDP Id) -> 10.0.0.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 145
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Local dataplane status received : No fault
BFD dataplane status received : Not sent
BFD peer monitor status received : No fault
Status received from access circuit : No fault
Status sent to access circuit : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer : No fault
Status received from network peer : No fault
Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        33                             33
Group ID     0                              0
Interface    Connect to CE1                 Connect to CE2
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x02                           0x02
             LSPV [2]                       LSPV [2]
VCCV CC type 0x07                           0x07
```

```
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
SSO Descriptor: 10.0.0.1/145, local label: 33
Dataplane:
SSM segment/switch IDs: 4345/4344 (used), PWID: 48
Rx Counters
2 input transit packets, 108 bytes
0 drops, 0 seq err
Tx Counters
3 output transit packets, 183 bytes
0 drops
```

Verifying HDLC-to-Ethernet Interworking (QinQ Mode) Configuration on an Ethernet PE Device

You can use **show** commands to view information about a HDLC-to-Ethernet interworking (QinQ mode) configuration on an Ethernet PE device.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show mpls l2transport vc detail**
3. **show l2vpn atom vc**
4. **show l2vpn atom vc detail**

DETAILED STEPS

Step 1 show mpls l2transport vc

The following is sample output from the **show mpls l2transport vc** command which displays basic information about HDLC-to-Ethernet interworking (QinQ mode) configuration on an Ethernet PE device:

Example:

```
Device# show mpls l2transport vc

Local intf  Local circuit  Dest address    VC ID      Status
----------- -------------- --------------- ---------- ----------
Gi1/0/0.10  Eth VLAN 10/20  203.0.113.1    145        UP
```

Step 2 show mpls l2transport vc detail

The following is sample output from the **show mpls l2transport vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (QinQ mode) configuration on an Ethernet PE device:

Example:

```
Device# show mpls l2transport vc detail

Local interface: Gi1/0/0.10 up, line protocol up, Eth VLAN 10/20 up
Interworking type is Ethernet
Destination address: 203.0.113.1, VC ID: 145, VC status: up
Output interface: Fa0/0/1, imposed label stack {19 27}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.11
Create time: 00:00:23, last status change time: 00:00:21
Last label FSM state change time: 00:00:21
```

```
Signaling protocol: LDP, peer 203.0.113.1:0 up
Targeted Hello: 10.0.0.1(LDP Id) -> 203.0.113.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 25, remote 27
Group ID: local 0, remote 0
MTU: local 1500, remote 1500
Remote interface description: Connect to CE1
Sequencing: receive disabled, send disabled
Control Word: On
SSO Descriptor: 203.0.113.1/145, local label: 25
Dataplane:
SSM segment/switch IDs: 4815/4814 (used), PWID: 124
VC statistics:
transit packet totals: receive 10, send 6
transit byte totals: receive 430, send 456
transit packet drops: receive 0, seq error 0, send 0
```

Step 3 **show l2vpn atom vc**

The following is sample output from the **show l2vpn atom vc** command which displays basic information about HDLC-to-Ethernet interworking (QinQ mode) configuration on an Ethernet PE device:

Example:

```
Device# show l2vpn atom vc

                            Service
Interface Peer ID    VC ID  Type   Name  Status
--------- ---------- ------ ------ ----- ----------
pw145     203.0.113.1  145    p2p    145   UP
```

Step 4 **show l2vpn atom vc detail**

The following is sample output from the **show l2vpn atom vc detail** command which displays detailed information about HDLC-to-Ethernet interworking (QinQ mode) configuration on an Ethernet PE device:

Example:

```
Device# show l2vpn atom vc detail

pseudowire145 is up, VC status is up PW type: Ethernet
Create time: 00:00:23, last status change time: 00:00:19
Last label FSM state change time: 00:00:19
Destination address: 203.0.113.1 VC ID: 145
Output interface: Fa0/0/1, imposed label stack {18 33}
Preferred path: not configured
Default path: active
Next hop: 10.0.0.11
Member of xconnect service eth145
Associated member Gi1/0/0.10 is up, status is up
Interworking type is Ethernet
Service id: 0xed000030
```

```
Signaling protocol: LDP, peer 203.0.113.1:0 up
Targeted Hello: 10.0.0.1(LDP Id) -> 203.0.113.1, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 145
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Local dataplane status received : No fault
BFD dataplane status received : Not sent
BFD peer monitor status received : No fault
Status received from access circuit : No fault
Status sent to access circuit : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer : No fault
Status received from network peer : No fault
Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        33                             33
Group ID     0                              0
Interface    Connect to CE2                 Connect to CE1
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x02                           0x02
             LSPV [2]                       LSPV [2]
VCCV CC type 0x07                           0x07
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
SSO Descriptor: 203.0.113.1/145, local label: 33
Dataplane:
SSM segment/switch IDs: 4361/4360 (used), PWID: 48
Rx Counters
8 input transit packets, 344 bytes
0 drops, 0 seq err
Tx Counters
5 output transit packets, 380 bytes
0 drops
```

Verifying L2VPN Interworking

To verify the L2VPN status (in the AToM configuration), use the following commands:

- **show connection** [**all** | **name** | **id** | **elements** | **port**]
- **show xconnect** [**all** | **interface** | **peer**]
- **show mpls l2transport** [**binding** | **checkpoint** | **hw-capability** | **summary** | **vc**]
- **show mpls infrastructure lfd pseudowire vcid**

Verifying L2VPN Interworking using the commands associated with the L2VPN Protocol-Based CLIs feature

To verify the L2VPN status (in the AToM configuration), use the following commands:

- **show connection** [**all** | **name** | **id** | **elements** | **port**]
- **show l2vpn service**[**all** | **interface** | **peer**]
- **show l2vpn atom** [**binding** | **checkpoint** | **hw-capability** | **summary** | **vc**]
- **show mpls infrastructure lfd pseudowire vcid**

Configuration Examples for L2VPN Interworking

Frame Relay DLCI-to-Ethernet VLAN 802.1Q Using Bridged Internetworking Example

The following example shows how to configure the Frame Relay DLCI-to-Ethernet VLAN 802.1Q feature using bridged interworking:

PE1 router	PE2 router
config t mpls label protocol ldp interface Loopback100 ip address 10.0.0.100 255.255.255.255 pseudowire-class fr-vlan encapsulation mpls interworking ethernet frame-relay switching interface serial 2/0/0:1 encapsulation frame-relay frame-relay intf-type dce connect mpls serial 2/0/0:1 567 l2transport xconnect 10.0.0.200 150 pw-class fr-vlan	config t mpls label protocol ldp interface Loopback200 ip address 10.0.0.200 255.255.255.255 pseudowire-class fr-vlan encapsulation mpls interworking ethernet interface gigabitethernet 5/1/0.3 encapsulation dot1q 1525 xconnect 10.0.0.100 150 pw-class fr-vlan

Frame Relay DLCI-to-Ethernet VLAN 802.1Q Using Bridged Internetworking Example using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to configure the Frame Relay DLCI-to-Ethernet VLAN 802.1Q feature using bridged interworking:

PE1 router	PE2 router
config t mpls label protocol ldp interface Loopback100 ip address 10.0.0.100 255.255.255.255 template type pseudowire fr-vlan encapsulation mpls interworking ethernet frame-relay switching interface serial 2/0/0:1 encapsulation frame-relay frame-relay intf-type dce connect mpls serial 2/0/0:1 567 l2transport interface pseudowire 100 source template type pseudowire fr-vlan neighbor 10.0.0.200 150 ! l2vpn xconnect context con1 member pseudowire 100 member 10.0.0.200 150 encapsulation mpls	config t mpls label protocol ldp interface Loopback200 ip address 10.0.0.200 255.255.255.255 template type pseudowire fr-vlan encapsulation mpls interworking ethernet interface gigabitethernet 5/1/0.3 encapsulation dot1q 1525 interface pseudowire 100 source template type pseudowire fr-vlan neighbor 10.0.0.100 150 ! l2vpn xconnect context con1 member pseudowire 100 member 10.0.0.100 150 encapsulation mpls

ATM AAL5-to-Ethernet VLAN 802.1Q Using Bridged Internetworking Example

The following example shows how to configure the ATM AAL5-to-Ethernet VLAN 802.1Q feature using bridged interworking:

PE1 router	PE2 router
config t mpls label protocol ldp interface Loopback100 ip address 10.0.0.100 255.255.255.255 pseudowire-class atm-vlan encapsulation mpls interworking ethernet interface atm 2/0/0 pvc 0/200 l2transport encapsulation aal5snap xconnect 10.0.0.200 140 pw-class atm-vlan	config t mpls label protocol ldp interface Loopback200 ip address 10.0.0.200 255.255.255.255 pseudowire-class atm-vlan encapsulation mpls interworking ethernet interface gigabitethernet 5/1/0.3 encapsulation dot1q 1525 xconnect 10.0.0.100 140 pw-class atm-vlan

ATM AAL5-to-Ethernet VLAN 802.1Q Using Bridged Internetworking Example using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to configure the ATM AAL5-to-Ethernet VLAN 802.1Q feature using bridged interworking:

PE1 router	PE2 router
config t mpls label protocol ldp interface Loopback100 ip address 10.0.0.100 255.255.255.255 template type pseudowire atm-vlan encapsulation mpls interworking ethernet interface atm 2/0/0 pvc 0/200 l2transport encapsulation aal5snap interface pseudowire 100 source template type pseudowire atm-vlan neighbor 10.0.0.200 140 ! l2vpn xconnect context con1 member pseudowire 100 member 10.0.0.200 140 encapsulation mpls	config t mpls label protocol ldp interface Loopback200 ip address 10.0.0.200 255.255.255.255 template type pseudowire atm-vlan encapsulation mpls interworking ethernet interface gigabitethernet 5/1/0.3 encapsulation dot1q 1525 interface pseudowire 100 source template type pseudowire atm-vlan neighbor 10.0.0.100 140 ! l2vpn xconnect context con1 member pseudowire 100 member 10.0.0.200 140 encapsulation mpls

ATM AAL5-to-Ethernet Port Using Routed Interworking Example

The following example shows how to configure the ATM AAL5-to-Ethernet Port feature using routed interworking:

PE1 router	PE2 router
config t mpls label protocol ldp interface Loopback100 ip address 10.0.0.100 255.255.255.255 pseudowire-class atm-eth encapsulation mpls interworking ip interface atm 2/0.1 pvc 0/200 l2transport encapsulation aal5 xconnect 10.0.0.200 140 pw-class atm-eth	config t mpls label protocol ldp interface Loopback200 ip address 10.0.0.200 255.255.255.255 pseudowire-class atm-eth encapsulation mpls interworking ip interface gigabitethernet 5/1/0 xconnect 10.0.0.100 140 pw-class atm-eth

Frame Relay DLCI-to-Ethernet Port Using Routed Interworking Example

The following example shows how to configure the Frame Relay DLCI-to-Ethernet Port feature using routed interworking:

PE1 router	PE2 router
config t mpls label protocol ldp interface Loopback100 ip address 10.0.0.100 255.255.255.255 pseudowire-class fr-eth encapsulation mpls interworking ip frame-relay switching interface serial 2/0/0:1 encapsulation frame-relay frame-relay intf-type dce frame-relay interface-dlci 567 switched connect fr-vlan-1 POS2/3/1 151 l2transport xconnect 10.0.0.200 151 pw-class pw-class-bridge	config t mpls label protocol ldp interface Loopback200 ip address 10.0.0.200 255.255.255.255 pseudowire-class fr-eth encapsulation mpls interworking ip interface gigabitethernet 5/1/0 xconnect 10.0.0.100 150 pw-class fr-eth

Frame Relay DLCI-to-Ethernet Port Using Routed Interworking Example using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to configure the Frame Relay DLCI-to-Ethernet Port feature using routed interworking:

PE1 router	PE2 router
config t mpls label protocol ldp interface Loopback100 ip address 10.0.0.100 255.255.255.255 template type pseudowire fr-eth encapsulation mpls interworking ip frame-relay switching interface serial 2/0/0:1 encapsulation frame-relay frame-relay intf-type dce frame-relay interface-dlci 567 switched connect fr-vlan-1 POS2/3/1 151 l2transport interface pseudowire 100 source template type pseudowire fr-eth neighbor 10.0.0.200 140 ! l2vpn xconnect context con1 member pseudowire 100 member 10.0.0.200 140 encapsulation mpls	config t mpls label protocol ldp interface Loopback200 ip address 10.0.0.200 255.255.255.255 template type pseudowire fr-eth encapsulation mpls interworking ip interface gigabitethernet 5/1/0 interface pseudowire 100 source template type pseudowire fr-eth neighbor 10.0.0.200 140 ! l2vpn xconnect context con1 member pseudowire 100 member 10.0.0.200 140 encapsulation mpls

Ethernet-to-VLAN over AToM--Bridged Example

The following example shows how to configure Ethernet-to-VLAN over AToM in a PE router:

PE1 router	PE2 router
ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! pseudowire-class atom encapsulation mpls ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet1/0 xconnect 10.8.8.8 123 pw-class atom	ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! pseudowire-class atom-eth-iw encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface FastEthernet1/0.1 encapsulation dot1q 100 xconnect 10.9.9.9 123 pw-class atom-eth-iw

Ethernet to VLAN over AToM (Bridged) Example using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows the configuration of Ethernet to VLAN over AToM:

PE1	PE2
ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! template type pseudowire atom-eth-iw encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface FastEthernet1/0.1 encapsulation dot1q 100 interface pseudowire 100 source template type pseudowire atom-eth-iw neighbor 10.8.8.8 123 ! l2vpn xconnect context con1 member pseudowire 100 member 10.8.8.8 123 encapsulation mpls	ip cef ! mpls label protocol ldp mpls ldp router-id Loopback0 force ! template type pseudowire atom encapsulation mpls ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet1/0 interface pseudowire 100 source template type pseudowire ether-pw neighbor 10.9.9.9 123 ! l2vpn xconnect context con1 member pseudowire 100 member 10.9.9.9 123 encapsulation mpls

VLAN-to-ATM AAL5 over AToM (Bridged) Example

The following example shows the configuration of VLAN-to-ATM AAL5 over AToM:

PE1 router	PE2 router

PE1 router	PE2 router
	ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! pseudowire-class inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet0/0.1 encapsulation dot1Q 10 xconnect 10.8.8.8 123 pw-class inter-ether ! router ospf 10 log-adjacency-changes network 10.9.9.9 0.0.0.0 area 0 network 10.1.1.2 0.0.0.0 area 0

PE1 router	PE2 router
ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! pseudowire-class inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface ATM1/0.1 point-to-point pvc 0/100 l2transport encapsulation aal5snap xconnect 10.9.9.9 123 pw-class inter-ether ! interface FastEthernet1/0 xconnect 10.9.9.9 1 pw-class inter-ether ! router ospf 10 log-adjacency-changes network 10.8.8.8 0.0.0.0 area 0	

PE1 router	PE2 router
`network 10.1.1.1 0.0.0.0 area 0`	

VLAN-to-ATM AAL5 over AToM (Bridged) Example using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows the configuration of VLAN-to-ATM AAL5 over AToM:

PE1 router	PE2 router

PE1 router	PE2 router
ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! template type pseudowire inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 ! interface ATM1/0.1 point-to-point pvc 0/100 l2transport encapsulation aal5snap interface pseudowire 100 source template type pseudowire inter-ether neighbor 10.9.9.9 123 ! l2vpn xconnect context con1 ! interface FastEthernet1/0	ip cef ! mpls ip mpls label protocol ldp mpls ldp router-id Loopback0 ! template type pseudowire inter-ether encapsulation mpls interworking ethernet ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 ! interface FastEthernet0/0 no ip address ! interface FastEthernet0/0.1 encapsulation dot1Q 10 interface pseudowire 100 source template type pseudowire inter-ether neighbor 10.8.8.8 123 ! l2vpn xconnect context con1 member pseudowire 100

PE1 router	PE2 router
interface pseudowire 100 source template type pseudowire inter-ether neighbor 10.9.9.9 1 ! l2vpn xconnect context con1 member pseudowire 100 member 10.9.9.9.9 1 encapsulation mpls ! router ospf 10 log-adjacency-changes network 10.8.8.8 0.0.0.0 area 0 network 10.1.1.1 0.0.0.0 area 0	member 10.8.8.8 123 encapsulation mpls ! router ospf 10 log-adjacency-changes network 10.9.9.9 0.0.0.0 area 0 network 10.1.1.2 0.0.0.0 area 0

Ethernet VLAN-to-PPP over AToM (Routed) Example

The following example shows the configuration of Ethernet VLAN-to-PPP over AToM

PE1 router	PE2 router
configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! pseudowire-class ppp-ether encapsulation mpls interworking ip ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 no shutdown ! interface POS2/0/1 no ip address encapsulation ppp no peer default ip address ppp ipcp address proxy 10.10.10.1 xconnect 10.9.9.9 300 pw-class ppp-ether no shutdown	configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! pseudowire-class ppp-ether encapsulation mpls interworking ip ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 no shutdown ! interface GigabitEthernet6/2 xconnect 10.8.8.8 300 pw-class ppp-ether no shutdown

Ethernet VLAN to PPP over AToM (Routed) Example using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows the configuration of Ethernet VLAN to PPP over AToM:

PE1	PE2

PE1	PE2
configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! template type pseudowire ppp-ether encapsulation mpls interworking ip ! interface Loopback0 ip address 10.8.8.8 255.255.255.255 no shutdown ! interface POS2/0/1 no ip address encapsulation ppp no peer default ip address ppp ipcp address proxy 10.10.10.1 interface pseudowire 100 source template type pseudowire ppp-ether neighbor 10.9.9.9 300 ! l2vpn xconnect context con1 member pseudowire 100	configure terminal mpls label protocol ldp mpls ldp router-id Loopback0 mpls ip ! template type pseudowire ppp-ether encapsulation mpls interworking ip ! interface Loopback0 ip address 10.9.9.9 255.255.255.255 no shutdown ! interface vlan300 mtu 4470 no ip address interface pseudowire 100 source template type pseudowire ppp-ether neighbor 10.8.8.8 300 ! l2vpn xconnect context con1 member pseudowire 100 member 10.8.8.8 300 encapsulation mpls no shutdown

PE1	PE2
member 10.9.9.9 300 encapsulation mpls no shutdown	! interface GigabitEthernet6/2 switchport switchport trunk encapsulation dot1q switchport trunk allowed vlan 300 switchport mode trunk no shutdown

ATM VC-to-VC Local Switching (Different Port) Example

The following example shows the configuration of ATM VC-to-VC local switching:

CE1 router	CE2 router	PE router
interface ATM1/0 no ip address atm clock INTERNAL no atm ilmi-keepalive no atm enable-ilmi-trap interface ATM1/0 ip address 10.1.1.1 255.255.255.0 no atm enable-ilmi-trap pvc 0/100 encapsulation aal5snap	interface ATM3/0 no ip address atm clock INTERNAL no atm ilmi-keepalive no atm enable-ilmi-trap ! interface ATM3/0.1 multipoint ip address 10.1.1.2 255.255.255.0 no atm enable-ilmi-trap pvc 0/50 protocol ip 10.1.1.1 encapsulation aal5snap	interface ATM0/1/0 no ip address atm clock INTERNAL no atm enable-ilmi-trap ! interface ATM0/1/0.50 point-to-point no atm enable-ilmi-trap pvc 0/50 l2transport encapsulation aal5 ! ! interface ATM0/1/1 no ip address atm clock INTERNAL no atm enable-ilmi-trap ! interface ATM0/1/1.100 point-to-point no atm enable-ilmi-trap pvc 0/100 l2transport encapsulation aal5 connect con_atm ATM0/1/1 0/100 ATM0/1/0 0/50

ATM VP-to-VP Local Switching (Different Port) Example

The following example shows the configuration of ATM VP-to-VP local switching:

CE1 router	CE2 router	PE router
interface ATM1/0 no ip address atm clock INTERNAL no atm enable-ilmi-trap ! interface ATM1/0.1 point-to-point ip address 10.1.1.1 255.255.255.0 no atm enable-ilmi-trap pvc 100/100 encapsulation aal5snap	interface ATM3/0 no ip address atm clock INTERNAL no atm ilmi-keepalive no atm enable-ilmi-trap ! interface ATM3/0.1 point-to-point ip address 10.1.1.2 255.255.255.0 no atm enable-ilmi-trap pvc 100/100 encapsulation aal5snap	interface ATM0/1/0 no ip address atm clock INTERNAL no atm ilmi-keepalive no atm enable-ilmi-trap ! interface ATM0/1/0.50 multipoint atm pvp 100 l2transport no atm enable-ilmi-trap ! interface ATM0/1/1 no ip address atm clock INTERNAL no atm ilmi-keepalive no atm enable-ilmi-trap ! interface ATM0/1/1.100 multipoint atm pvp 100 l2transport no atm enable-ilmi-trap connect atm_con ATM0/1/1 100 ATM0/1/0 100

Example: Configuring HDLC-to-Ethernet Interworking: Controller Slot on HDLC Devices

The following example shows how to configure the serial controller and interface on HDLC devices:

HDLC CE device	HDLC PE device
enable configure terminal controller E1 2/0 channel-group 0 timeslots 1 no shutdown ! interface serial 2/0:0 no shutdown end	enable configure terminal controller E1 0/1/0 channel-group 0 timeslots 1 no shutdown ! interface serial 0/1/0:0 no shutdown end

Example: Configuring HDLC-to-Ethernet Bridged Interworking on HDLC Devices

The following example shows how to configure HDLC-to-Ethernet bridged interworking on HDLC devices:

HDLC CE device	HDLC PE device
enable configure terminal bridge irb bridge 1 protocol ieee bridge 1 route ip ! interface BVI1 ip address 192.0.2.1 255.255.255.0 no shutdown ! interface serial 2/0:0 encapsulation hdlc bridge-group 1 no shutdown end	enable configure terminal pseudowire-class pw-iw-eth encapsulation mpls interworking Ethernet ! interface serial 0/1/0:0 encapsulation hdlc no ip address xconnect 203.0.113.10 100 pw-class pw-iw-eth no shutdown end

Example: Configuring HDLC-to-Ethernet Bridged Interworking on HDLC Devices Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

The following example shows how to configure HDLC-to-Ethernet bridged interworking on HDLC devices using the commands associated with the L2VPN protocol-based CLIs feature:

HDLC CE device	HDLC PE device
enable configure terminal bridge irb bridge 1 protocol ieee bridge 1 route ip ! interface BVI1 ip address 192.0.2.1 255.255.255.0 no shutdown ! interface serial 2/0:0 encapsulation hdlc bridge-group 1 no shutdown end	enable configure terminal interface serial 0/1/0:0 encapsulation hdlc no ip address no shutdown ! interface pseudowire 101 encapsulation mpls neighbor 203.0.113.10 100 signaling protocol ldp no shutdown ! l2vpn xconnect context hdlc interworking ethernet member Serial 0/1/0:0 member pseudowire 101 no shutdown end

Example: Configuring HDLC-to-Ethernet Bridged Interworking on Ethernet Devices

The following example shows how to configure HDLC-to-Ethernet bridged interworking on Ethernet devices:

Ethernet CE device	Ethernet PE device
enable configure terminal interface GigabitEthernet0/1 ip address 198.51.100.19 255.255.255.0 ip irdp ip irdp maxadvertinterval 4 no shutdown end	enable configure terminal pseudowire-class pw-iw-eth encapsulation mpls interworking Ethernet ! interface GigabitEthernet 1/0/0 no ip address xconnect 203.0.113.20 100 pseudowire-class pw-iw-eth no shutdown end

Example: Configuring HDLC-to-Ethernet Bridged Interworking on Ethernet Devices Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

The following example shows how to configure HDLC-to-Ethernet bridged interworking on Ethernet devices using the commands associated with the L2VPN protocol-based CLIs feature:

Ethernet CE device	Ethernet PE device
enable configure terminal interface GigabitEthernet 0/1 ip address 198.51.100.19 255.255.255.0 ip irdp ip irdp maxadvertinterval 4 no shutdown end	enable configure terminal interface GigabitEthernet 1/0/0 no ip address no shutdown ! interface pseudowire 101 encapsulation mpls neighbor 203.0.113.20 100 signaling protocol ldp no shutdown ! l2vpn xconnect context eth interworking ethernet member GigabitEthernet 1/0/0 member pseudowire101 no shutdown end

Example: Configuring HDLC-to-VLAN Bridged Interworking (Port Mode) on Ethernet Devices

The following example shows how to configure HDLC-to-VLAN bridged interworking (port mode) on Ethernet devices:

Ethernet CE device	Ethernet PE device

enable
 configure terminal
  interface GigabitEthernet 0/1
  no ip address
  no shutdown
!
interface GigabitEthernet 0/1.10
 encapsulation dot1q 10
 ip address 198.51.100.19 255.255.255.0
 	
 ip irdp
 ip irdp maxadvertinterval 4
 no shutdown
end
	enable
	 configure terminal
	  pseudowire-class pw-iw-eth
	  encapsulation mpls
	  interworking Ethernet
	!
	interface GigabitEthernet 1/0/0
	 no ip address
	 no shutdown
	!
	interface GigabitEthernet 1/0/0.10
	 encapsulation dot1Q 10
	 no ip address
	!
	 xconnect 203.0.113.20 100 pseudowire-class
	pw-iw-eth
	  no shutdown
	end

Example: Configuring HDLC-to-VLAN Bridged Interworking on Ethernet Devices Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

The following example shows how to configure HDLC-to-VLAN bridged interworking on Ethernet devices using the commands associated with the L2VPN protocol-based CLIs feature:

Ethernet CE device	Ethernet PE device

enable
 configure terminal
  interface GigabitEthernet 0/1
  no ip address
  no shutdown
!
interface GigabitEthernet 0/1.10
 encapsulation dot1q 10
 ip address 198.51.100.19 255.255.255.0
 	
 ip irdp
 ip irdp maxadvertinterval 4
 no shutdown
end
	enable
	 configure terminal
	  interface GigabitEthernet 1/0/0
	  no ip address
	  no shutdown
	!
	interface GigabitEthernet 1/0/0.10
	
	 encapsulation dot1q 10
	 no ip addres
	 no shutdown
	!
	interface pseudowire 101
	 encapsulation mpls
	 neighbor 203.0.113.20 100
	 signaling protocol ldp
	 no shutdown
	!
	l2vpn xconnect context vlan
	 interworking ethernet
	 member GigabitEthernet 1/0/0.10
	 member pseudowire 101
	 no shutdown
	end

Example: Configuring HDLC-to-VLAN Bridged Interworking (dot1q Mode) Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

The following example shows how to configure HDLC-to-VLAN bridged interworking (dot1q mode) using the commands associated with the L2VPN protocol-based CLIs feature:

HDLC PE device	Ethernet PE device

enable
 configure terminal
  template type pseudowire hdlc-vlan1
  encapsulation mpls
!
interface pseudowire 107
 source template type pseudowire hdlc-vlan1
 	
  encapsulation mpls
  neighbor 203.0.113.10 107
  signaling protocol ldp
  no shutdown
!
l2vpn xconnect context hdlc-vlan1-con
 interworking ethernet
 member Serial 0/2/0:3
 member pseudowire 107
 no shutdown
end
	enable
	 configure terminal
	  interface FastEthernet 0/0/0.16
	  encapsulation dot1q 16
	  no ip addres
	  no shutdown
	!
	template type pseudowire hdlc-vlan1
	 encapsulation mpls
	!
	interface pseudowire 107
	 source template type pseudowire hdlc-vlan1
	
	  encapsulation mpls
	  neighbor 203.0.113.20 107
	  signaling protocol ldp
	  no shutdown
	!
	l2vpn xconnect context hdlc-vlan1-con
	 interworking ethernet
	 member FastEthernet 0/0/0.16
	 member pseudowire 107
	 no shutdown
	end

Example: Configuring HDLC-to-VLAN Bridged Interworking (QinQ Mode) on Ethernet Devices

The following example shows how to configure HDLC-to-VLAN bridged interworking (QinQ mode) on Ethernet devices:

Ethernet CE device	Ethernet PE device

enable
 configure terminal
  interface GigabitEthernet 0/1
  no ip address
  no shutdown
!
interface GigabitEthernet 0/1.10
 encapsulation dot1q 10 second-dot1q 20
 	
 ip address 198.51.100.19 255.255.255.0
 	
 ip irdp
 ip irdp maxadvertinterval 4
 no shutdown
end
	enable
	 configure terminal
	  pseudowire-class pw-iw-eth
	  encapsulation mpls
	  interworking Ethernet
	!
	interface GigabitEthernet 1/0/0
	 no ip address
	 no shutdown
	!
	interface GigabitEthernet 1/0/0.10
	 encapsulation dot1Q 10 second-dot1q 20
	 no ip address
	 xconnect 203.0.113.20 100 pseudowire-class
	pw-iw-eth
	  no shutdown
	end

Example: Configuring HDLC-to-VLAN Bridged Interworking (QinQ Mode) on Ethernet Devices Using the Commands Associated with the L2VPN Protocol-Based CLIs Feature

The following example shows how to configure HDLC-to-VLAN bridged interworking (QinQ mode) on Ethernet devices using the commands associated with the L2VPN protocol-based CLIs feature:

Ethernet CE device	Ethernet PE device
enable configure terminal interface GigabitEthernet 0/1 no ip address no shutdown ! interface GigabitEthernet 0/1.10 encapsulation dot1q 10 second-dot1q 20 ip address 198.51.100.19 255.255.255.0 ip irdp ip irdp maxadvertinterval 4 no shutdown end	enable configure terminal interface GigabitEthernet 1/0/0 no ip address no shutdown ! interface GigabitEthernet 1/0/0.10 encapsulation dot1q 10 second-dot1q 20 no ip address no shutdown ! interface pseudowire 101 encapsulation mpls neighbor 203.0.113.20 100 signaling protocol ldp no shutdown ! l2vpn xconnect context qinq interworking ethernet member GigabitEthernet 1/0/0.10 member pseudowire 101 no shutdown end

Additional References for L2VPN Interworking

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Multiprotocol Label Switching Command Reference
Any Transport over MPLS	Any Transport over MPLS

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2tpext-l2tp-base-03.txt	*Layer Two Tunneling Protocol (Version 3) 'L2TPv3'*

Standard/RFC	Title
draft-martini-l2circuit-trans-mpls-09.txt	*Transport of Layer 2 Frames Over MPLS*
draft-ietf-pwe3-frame-relay-03.txt.	*Encapsulation Methods for Transport of Frame Relay over MPLS Networks*
draft-martini-l2circuit-encap-mpls-04.txt.	*Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks*
draft-ietf-pwe3-ethernet-encap-08.txt.	*Encapsulation Methods for Transport of Ethernet over MPLS Networks*
draft-ietf-pwe3-hdlc-ppp-encap-mpls-03.txt.	*Encapsulation Methods for Transport of PPP/HDLC over MPLS Networks*
draft-ietf-ppvpn-l2vpn-00.txt.	*An Architecture for L2VPNs*
RFC 4618	Encapsulation Methods for Transport of PPP/High-Level Data Link Control (HDLC) over MPLS Networks

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

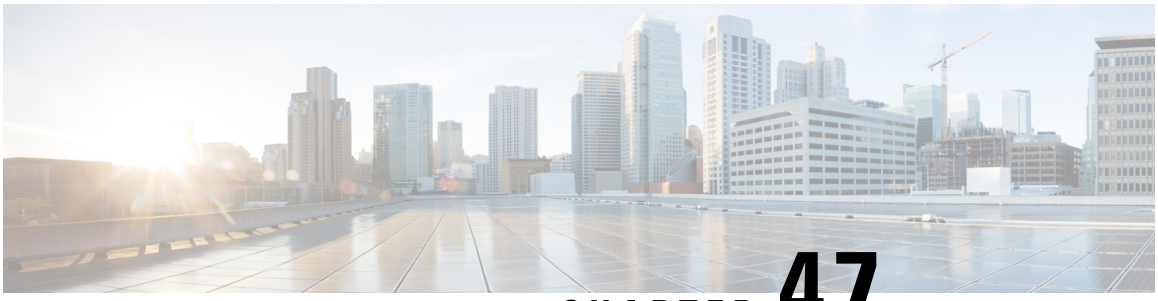
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for L2VPN Interworking

Table 107: Feature Information for L2VPN Interworking

Feature Name	Releases	Feature Information
L2VPN Interworking	Cisco IOS XE Release 2.4 Cisco IOS XE Release 3.3S	This feature allows disparate ACs to be connected. An interworking function facilitates the translation between the different Layer 2 encapsulations. The following commands were introduced or modified: **debug frame-relay pseudowire**, **debug ssm**, **interworking**, **mtu**, **pseudowire-class**, **show l2tun session**, **show l2tun tunnel** , **show mpls l2transport vc**, **show platform**.
L2VPN Interworking: Ethernet to VLAN Interworking	Cisco IOS XE Release 2.4	This feature allows interworking by stripping the VLAN tags and sending them as untagged frames on the remote end.
L2VPN Interworking: Ethernet VLAN to Frame Relay	Cisco IOS XE Release 3.3S	This feature allows interworking of Ethernet VLANs with Frame Relay DLCIs. The following command was modified: **interworking**
L2VPN Interworking: Ethernet VLAN to PPP	Cisco IOS XE Release 3.3S	The L2VPN interworking - Ethernet VLAN-to-PPP feature allows disparate ACs to be connected. An interworking function facilitates the translation between the following Layer 2 encapsulations.
L2VPN Interworking: Frame Relay to ATM (Bridged Mode)	Cisco IOS XE Release 3.6S	This feature allows Frame Relay to ATM Interworking using bridged and routed mode encapsulation.
L2VPN Interworking: HDLC to Ethernet Interworking	Cisco IOS XE Release 3.13S	High-Level Data Link Control (HDLC) and Ethernet are two independent data link layer transport protocols that utilize the Any Transport over MPLS (AToM) framework to communicate with each other. The interworking function enables translation between two heterogeneous Layer 2 encapsulations over a Multiprotocol Label Switching (MPLS) backbone. In Cisco IOS XE Release 3.13S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. This feature introduced no new or modified commands.



CHAPTER 47

L2VPN Pseudowire Preferential Forwarding

The L2VPN: Pseudowire Preferential Forwarding feature allows you to configure the pseudowires so that you can use **ping** and **show** commands to find status information for the pseudowires before, during, and after a switchover.

Prerequisites for L2VPN—Pseudowire Preferential Forwarding

- Before configuring the L2VPN: Pseudowire Preferential Forwarding feature, you should understand the concepts in the following documents:
  - Preferential Forwarding Status Bit Definition (draft-ietf-pwe3-redundancy-bit-xx.txt)
  - *MPLS Pseudowire Status Signaling*
  - *L2VPN Pseudowire Redundancy*
  - *NSF/SSO--Any Transport over MPLS and AToM Graceful Restart*
  - *MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV*
- The PE routers must be configured with the following features:
  - *L2VPN Pseudowire Redundancy*
  - *NSF/SSO--Any Transport over MPLS and AToM Graceful Restart*
- The L2VPN: Pseudowire Preferential Forwarding feature requires that the following mechanisms be in place to enable you to detect a failure in the network:
  - *Label switched paths (LSPs) Ping/Traceroute and Any Transport over MPLS Virtual Circuit Connection Verification (AToM VCCV)*
  - *Local Management Interface (LMI)*
  - *Operation, Administration, and Maintenance (OAM)*

Guidelines and Limitations for L2VPN--Pseudowire Preferential Forwarding

- Only ATM attachment circuits are supported.
- The following features are not supported:
  - Port mode cell relay
  - Any Transport over MPLS: AAL5 over MPLS
  - VC cell packing
  - OAM emulation
  - ILMI/PVC-D
  - Permanent virtual circuit (PVC) Range
  - L2TPv3 Pseudowire Redundancy
  - Local switching
  - Multiple backup pseudowires
  - Static pseudowires

Information About L2VPN--Pseudowire Preferential Forwarding

Overview of L2VPN--Pseudowire Preferential Forwarding

The L2VPN: Pseudowire Preferential Forwarding feature allows you to configure pseudowires so that you can use **ping** , **traceroute** , and **show** commands to find status information before, during, and after a switchover. The implementation of this feature is based on *Preferential Forwarding Status Bit Definition* (draft-ietf-pwe3-redundancy-bit-xx.txt). The L2VPN: Pseudowire Preferential Forwarding feature provides the following enhancements for displaying information about the pseudowires:

- You can issue **ping mpls** commands on the backup pseudowires.
- You can display status of the pseudowires before, during, and after a switchover using the **show xconnect** and **show mpls l2transport vc** commands.

Note In a single-segment pseudowire, the PE routers at each end of the pseudowire serve as the termination points. In multisegment pseudowires, the terminating PE routers serve as the termination points.

Overview of L2VPN—Pseudowire Preferential Forwarding using the commands associated with the L2VPN Protocol-Based CLIs feature

The L2VPN: Pseudowire Preferential Forwarding feature allows you to configure pseudowires so that you can use **ping**, **traceroute**, and **show** commands to find status information before, during, and after a switchover. The implementation of this feature is based on *Preferential Forwarding Status Bit Definition*

(draft-ietf-pwe3-redundancy-bit-xx.txt). The L2VPN: Pseudowire Preferential Forwarding feature provides the following enhancements for displaying information about the pseudowires:

- You can issue **ping mpls** commands on the backup pseudowires.
- You can display status of the pseudowires before, during, and after a switchover using the **show l2vpn service** and **show l2vpn atom vc** commands.

Note In a single-segment pseudowire, the PE routers at each end of the pseudowire serve as the termination points. In multisegment pseudowires, the terminating PE routers serve as the termination points.

How to Configure L2VPN--Pseudowire Preferential Forwarding

Configuring the Pseudowire Connection Between PE Routers

You set up a connection called a pseudowire between the routers to transmit Layer 2 frames between PE routers.

As part of the pseudowire configuration, issue the **status redundancy master** command to make it the master. This enables the L2VPN: Pseudowire Preferential Forwarding feature to display the status of the active and backup pseudowires. By default, the PE router is in slave mode.

Note One pseudowire must be the master, and the other must be the slave. You cannot configure both pseudowires as master or slave.

Note You must specify the **encapsulation mpls** command as part of the pseudowire class in order for the AToM VCs to work properly. If you omit the **encapsulation mpls** command, you receive the following error: % Incomplete command.

Before you begin

The PE routers must be configured for the L2VPN Pseudowire Redundancy and NSF/SSO--Any Transport over MPLS and AToM Graceful Restart features. See the following documents for configuration instructions.

- *L2VPN Pseudowire Redundancy*
- *NSF/SSO--Any Transport over MPLS and AToM Graceful Restart*

SUMMARY STEPS

1. **configure terminal**
2. **pseudowire-class name**
3. **encapsulation mpls**

4. **status redundancy** {**master**| **slave**}
5. **interworking** {**ethernet** | **ip**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**pseudowire-class name** **Example:** `switch(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify, and enters pseudowire class configuration mode.
Step 3	**encapsulation mpls** **Example:** `switch(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation. • For AToM, the encapsulation type is mpls.
Step 4	**status redundancy** {**master**| **slave**} **Example:** `switch(config-pw)# status redundancy master`	Configures the pseudowire as the master or slave. This enables the L2VPN: Pseudowire Preferential Forwarding feature to display the status of the active and backup pseudowires. • By default, the PE router is in slave mode. **Note** One pseudowire must be the master, and the other must be the slave. You cannot configure both pseudowires as master or slave.
Step 5	**interworking** {**ethernet**	**ip**} **Example:** `switch(config-pw)# interworking ip`	(Optional) Enables the translation between the different Layer 2 encapsulations.

Configuring the Pseudowire Connection Between PE Routers

You set up a connection called a pseudowire between the routers to transmit Layer 2 frames between PE routers.

As part of the pseudowire configuration, issue the **status redundancy master** command to make it the master. This enables the L2VPN: Pseudowire Preferential Forwarding feature to display the status of the active and backup pseudowires. By default, the PE router is in slave mode.

Note One pseudowire must be the master, and the other must be the slave. You cannot configure both pseudowires as master or slave.

Note You must specify the **encapsulation mpls** command as part of the pseudowire class in order for the AToM VCs to work properly. If you omit the **encapsulation mpls** command, you receive the following error: % Incomplete command.

Before you begin

The PE routers must be configured for the L2VPN Pseudowire Redundancy and NSF/SSO--Any Transport over MPLS and AToM Graceful Restart features. See the following documents for configuration instructions.

- *L2VPN Pseudowire Redundancy*
- *NSF/SSO--Any Transport over MPLS and AToM Graceful Restart*

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pseudowire** *number*
4. **encapsulation mpls**
5. **neighbor** *peer-address vcid-value*
6. **status redundancy** {**master**| **slave**}
7. **interworking** {**ethernet** | **ip**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 1`	Establishes an interface pseudowire with a value that you specify, and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls**	Specifies the tunneling encapsulation.

	Command or Action	Purpose
	Example: `Device(config-pw)# encapsulation mpls`	• For AToM, the encapsulation type is mpls.
Step 5	**neighbor** *peer-address vcid-value* **Example:** `Router(config-pw)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 6	**status redundancy** {**master**| **slave**} **Example:** `Device(config-pw)# status redundancy master`	Configures the pseudowire as the master or slave. This enables the L2VPN: Pseudowire Preferential Forwarding feature to display the status of the active and backup pseudowires. • By default, the PE router is in slave mode. **Note** One pseudowire must be the master, and the other must be the slave. You cannot configure both pseudowires as master or slave.
Step 7	**interworking** {**ethernet** | **ip**} **Example:** `Device(config-pw)# interworking ip`	(Optional) Enables the translation between the different Layer 2 encapsulations.

Configuration Examples for L2VPN--Pseudowire Preferential Forwarding

Example: L2VPN--Pseudowire Preferential Forwarding Configuration

The following commands configure a PE router with the L2VPN: Pseudowire Preferential Forwarding feature:

```
mpls ldp graceful-restart
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp advertise-labels
!
pseudowire-class mpls
 encapsulation mpls
 status redundancy master
interface ATM0/2/0.1 multipoint
 logging event subif-link-status
 atm pvp 50 l2transport
  xconnect 10.1.1.2 100 pw-class mpls
   backup peer 10.1.1.3 100 encap mpls
end
```

Example: L2VPN--Pseudowire Preferential Forwarding Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

The following commands configure a PE router with the L2VPN: Pseudowire Preferential Forwarding feature:

```
mpls ldp graceful-restart
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp advertise-labels
!
interface pseudowire1
 encapsulation mpls
 status redundancy master
 neighbor 10.0.0.1 123
interface ATM0/2/0.1 multipoint
 logging event subif-link-status
 atm pvp 50 l2transport
  interface pseudowire 100
  encapsulation mpls
  neighbor 10.1.1.2 100
!
l2vpn xconnect context A
 member pseudowire 100
 member atm 100
end
```

Example: Displaying the Status of the Pseudowires

The following examples show the status of the active and backup pseudowires before, during, and after a switchover.

The **show mpls l2transport vc** command on the active PE router displays the status of the pseudowires:

```
Router# show mpls l2transport vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.2         100        UP
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.3         100        STANDBY
```

The **show mpls l2transport vc** command on the backup PE router displays the status of the pseudowires. The active pseudowire on the backup PE router has the HOTSTANDBY status.

```
Router1-standby# show mpls l2transport vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.2         100        HOTSTANDBY
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.3         100        DOWN
```

During a switchover, the status of the active and backup pseudowires changes:

```
Router# show mpls l2transport vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
```

```
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.2         100        RECOVERING
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.3         100        DOWN
```

After the switchover is complete, the recovering pseudowire shows a status of UP:

```
Router# show mpls l2transport vc

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.2         100        UP
AT0/2/0/0.1    ATM VPC CELL 50            10.1.1.3         100        STANDBY
```

The **show xconnect** command displays the standby (SB) state for the backup pseudowire, which is independent of the stateful switchover mode of the router:

```
Router# show xconnect all

Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware
XC ST        Segment 1                                 S1 Segment 2
           S2
------+---------------------------------+--+---------------------------------+---------
UP pri ac   AT1/1/0/0.1/1/1:220/220(ATM V  UP mpls 10.193.193.3:330            UP
IA sec ac   AT1/1/0/0.1/1/1:220/220(ATM V  UP mpls 10.193.193.3:331            SB
```

The **ping mpls** and **traceroute mpls** commands show that the dataplane is active on the backup pseudowire:

```
Router# ping mpls pseudowire 10.193.193.22 331

%Total number of MS-PW segments is less than segment number; Adjusting the segment number
to 1
Sending 5, 100-byte MPLS Echos to 10.193.193.22,
     timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Router# traceroute mpls pseudowire 10.193.193.22 331 segment 1

Tracing MS-PW segments within range [1-1] peer address 10.193.193.22 and timeout 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
! 1 10.193.33.22 4 ms [Labels: 23 Exp: 0]
    local 10.193.193.3 remote 10.193.193.22 vc id 331
```

Additional References

Related Documents

Related Topic	Document Title
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
L2VPN Pseudowires	• *L2VPN Pseudowire Redundancy* • *MPLS Pseudowire Status Signaling*
NSF/SSO for L2VPNs	*NSF/SSO--Any Transport over MPLS and AToM Graceful Restart*
Ping and Traceroute for L2VPNs	*MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV*

Standards

Standard	Title
draft-ietf-pwe3-redundancy-bit-xx.txt	Preferential Forwarding Status Bit Definition

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

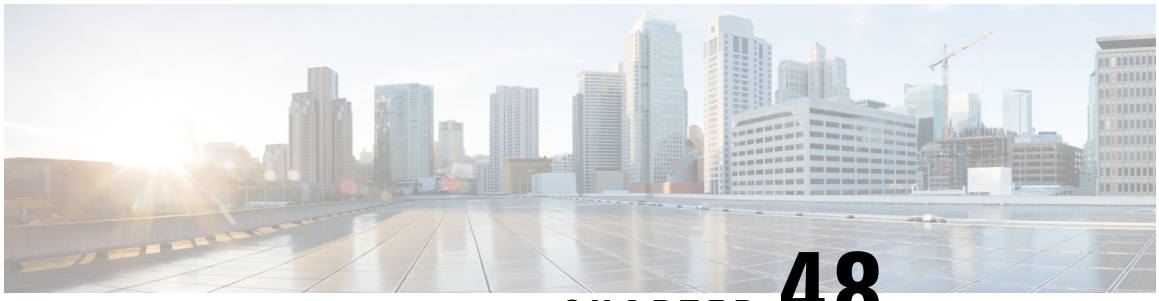
Feature Information for L2VPN--Pseudowire Preferential Forwarding

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 108: Feature Information for L2VPN: Pseudowire Preferential Forwarding

Feature Name	Releases	Feature Information
L2VPN: Pseudowire Preferential Forwarding	Cisco IOS XE Release 2.3	This feature allows you to configure the pseudowires so that you can use **ping** and **show** commands to find status information of the pseudowires before, during, and after a switchover. The following commands were introduced or modified: **show mpls l2transport vc**, **show xconnect**, **status redundancy**.



CHAPTER 48

L2VPN Multisegment Pseudowires

The L2VPN Multisegment Pseudowires feature enables you to configure two or more Layer 2 pseudowire segments that function as a single pseudowire. The L2VPN Multisegment Pseudowires feature span multiple cores or autonomous systems of the same or different carrier networks.

Prerequisites for L2VPN Multisegment Pseudowires

Before configuring this feature, see the following documents:

- Any Transport over MPLS
- *L2VPN Pseudowire Switching*
- MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV
- Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) (RFC 4447)

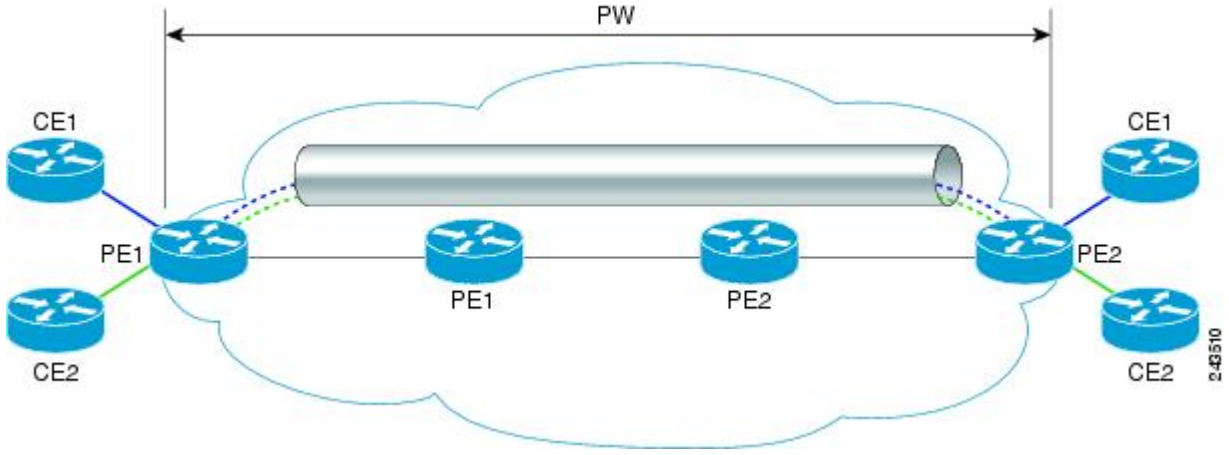
Restrictions for L2VPN Multisegment Pseudowires

- Only Mutliprotocol (MPLS) Layer 2 pseudowires are supported.
- Only manual configuration of the pseudowires (including S-PE and T-PE routers) is supported.
- The L2VPN Pseudowire Switching feature is supported for pseudowires advertised with FEC 128. FEC 129 is not supported.
- The S-PE router is limited to 1600 pseudowires.

Information About L2VPN Multisegment Pseudowires

L2VPN Pseudowire Defined

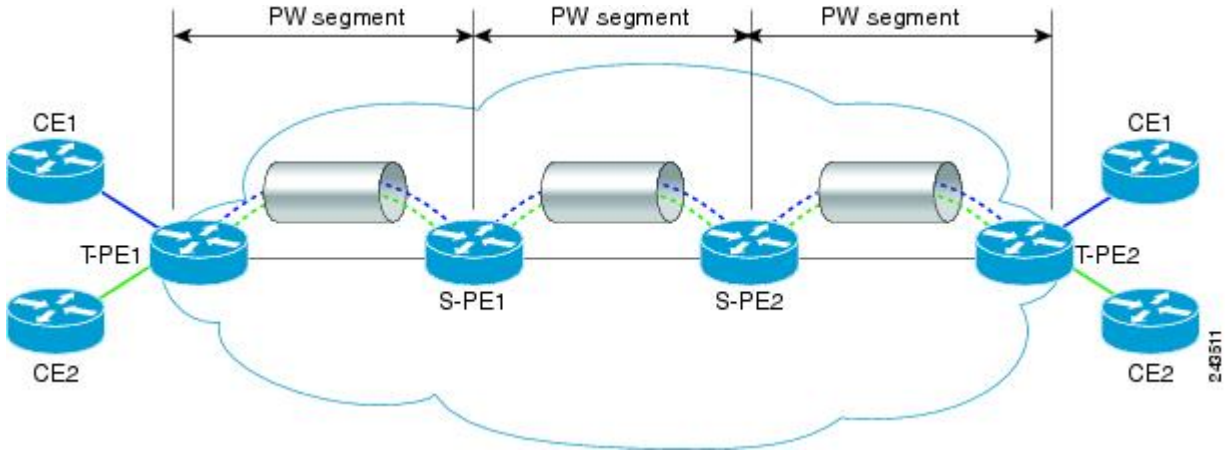
An L2VPN pseudowire (PW) is a tunnel established between two provider edge (PE) routers across the core carrying the Layer 2 payload encapsulated as MPLS data, as shown in the figure below. This helps carriers migrate from traditional Layer 2 networks such as Frame Relay and ATM to an MPLS core. In the L2VPN pseudowire shown in the figure, the PWs between two PE routers are located within the same autonomous system. Routers PE1 and PE2 are called terminating PE routers (T-PEs). Attachment circuits are bounded to the PW on these PE routers.



L2VPN Multisegment Pseudowire Defined

An L2VPN multisegment pseudowire (MS-PW) is a set of two or more PW segments that function as a single PW. It is also known as switched PW. MS-PWs span multiple cores or autonomous systems of the same or different carrier networks. A L2VPN MS-PW can include up to 254 PW segments.

The figure below is an example of a Multisegment Pseudowire topology.



The end routers are called terminating PE routers (T-PEs), and the switching routers are called S-PE routers. The S-PE router terminates the tunnels of the preceding and succeeding PW segments in an MS-PW. The S-PE router can switch the control and data planes of the preceding and succeeding PW segments of the MS-PW. An MS-PW is declared to be up when all the single-segment PWs are up. For more information, see the *L2VPN Pseudowire Switching* document.

How to Configure L2VPN Multisegment Pseudowires

Configuring L2VPN Multisegment Pseudowires

Perform the following steps on the S-PE routers to create L2VPN Multisegment Pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. mpls label protocol ldp
4. **mpls ldp router-id** *interface* **force**
5. **pseudowire-class** name
6. **encapsulation mpls**
7. **switching tlv**
8. **exit**
9. **l2 vfi** *name* **point-to-point**
10. **description** *string*
11. **neighbor** *ip-address vcid {* **encapsulation mpls** **pw-class** *pw-class-name}*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	mpls label protocol ldp **Example:** `Router(config)# mpls label protocol ldp`	Configures the use of Label Distribution Protocol (LDP) on all interfaces.
Step 4	**mpls ldp router-id** *interface* **force** **Example:**	Specifies the preferred interface for determining the LDP router ID.

	Command or Action	Purpose
	`Router(config)# mpls ldp router-id loopback0 force`	
Step 5	**pseudowire-class** name **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify, and enters pseudowire class configuration mode.
Step 6	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. • For MPLS L2VPNs, the encapsulation type is **mpls**.
Step 7	**switching tlv** **Example:** `Router(config-pw-class)# switching tlv`	(Optional) Enables the advertisement of the switching point type-length variable (TLV) in the label binding. • This command is enabled by default.
Step 8	**exit** **Example:** `Router(config-pw-class)# exit`	Exits pseudowire class configuration mode.
Step 9	**l2 vfi** *name* **point-to-point** **Example:** `Router(config)# l2 vfi atomtunnel point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 10	**description** *string* **Example:** `Router(config-vfi)# description segment1`	Provides a description of the switching provider edge router for a multisegment pseudowire.
Step 11	**neighbor** *ip-address vcid {* **encapsulation mpls pw-class** *pw-class-name}* **Example:** `Router(config-vfi)# neighbor 10.0.0.1 100 pw-class mpls`	Sets up an emulated VC. • Specify the IP address and the VC ID of the peer router. Also specify the pseudowire class to use for the emulated VC. **Note** Only two **neighbor**commands are allowed for each **l2 vfi point-to-point** command.

ConfiguringL2VPNMultisegmentPseudowiresusingthecommandsassociated with the L2VPN Protocol-Based CLIs feature

Perform this task on the S-PE routers to create L2VPN multisegment pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **mpls ldp router-id** *interface* **force**
5. **interface pseudowire** *number*
6. **encapsulation mpls**
7. **switching tlv**
8. **neighbor** *peer-address vcid-value*
9. **exit**
10. **l2vpn xconnect context** *context-name*
11. **description** *string*
12. **member** *ip-address vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Device(config)# mpls label protocol ldp`	Configures the use of Label Distribution Protocol (LDP) on all interfaces.
Step 4	**mpls ldp router-id** *interface* **force** **Example:** `Device(config)# mpls ldp router-id loopback0 force`	Specifies the preferred interface for determining the LDP router ID.
Step 5	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 1`	Establishes an interface pseudowire with a value that you specify, and enters pseudowire configuration mode.
Step 6	**encapsulation mpls** **Example:** `Device(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation. • For MPLS L2VPNs, the encapsulation type is **mpls**.

	Command or Action	Purpose
Step 7	**switching tlv** **Example:** `Device(config-pw)# switching tlv`	(Optional) Enables the advertisement of the switching point type-length variable (TLV) in the label binding. • This command is enabled by default.
Step 8	**neighbor** *peer-address vcid-value* **Example:** `Router(config-pw)#  neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 9	**exit** **Example:** `Device(config-pw)# exit`	Exits pseudowire configuration mode.
Step 10	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 11	**description** *string* **Example:** `Device(config-xconnect)# description segment1`	Provides a description of the switching provider edge router for a multisegment pseudowire.
Step 12	**member** *ip-address vcid* **encapsulation mpls** **Example:** `Device(config-xconnect)# member 10.10.10.10 1 encapsulation mpls`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection. **Note** Only two **member**commands are allowed for each **l2vpn xconnect context** command.

Displaying Information About the L2VPN Multisegment Pseudowires

SUMMARY STEPS

1. **show mpls l2transport binding**
2. **show mpls l2transport vc detail**

DETAILED STEPS

Step 1 **show mpls l2transport binding**

Use the **show mpls l2transport binding** command to display information about the pseudowire switching point, as shown in bold in the output. (In the following examples PE1 and PE4 are the T-PE routers.)

Example:

```
Router# show mpls l2transport binding

  Destination Address: 10.1.1.1,  VC ID: 102
    Local Label:  17
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2], TTL [3]
              CV Type: LSPV [2]
    Remote Label: 16
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2], TTL [3]
              CV Type: LSPV [2]
        PW Switching Point:
             Vcid   local IP addr      remote IP addr      Description
             101    10.11.11.11        10.20.20.20         PW Switching Point PE3
             100    10.20.20.20        10.11.11.11         PW Switching Point PE2
```

Step 2 **show mpls l2transport vc detail**

Use the **show mpls l2transport vc detail** command to display status of the pseudowire switching point. In the following example, the output (shown in bold) displays the segment that is the source of the fault of the multisegment pseudowire:

Example:

```
Router# show mpls l2transport vc detail
Local interface: Se3/0/0 up, line protocol up, HDLC up
  Destination address: 12.1.1.1, VC ID: 100, VC status: down
    Output interface: Se2/0, imposed label stack {23}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:03:02, last status change time: 00:01:41
  Signaling protocol: LDP, peer 10.1.1.1:0 up
    Targeted Hello: 10.1.1.4(LDP Id) -> 10.1.1.1, LDP is UP
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRrd
      Last local dataplane   status rcvd: No fault
      Last local SSS circuit status rcvd: No fault
      Last local SSS circuit status sent: DOWN(PW-tx-fault)
      Last local  LDP TLV    status sent: No fault
      Last remote LDP TLV    status rcvd: DOWN(PW-tx-fault)
       PW Switching Point:
       Fault type  Vcid   local IP addr   remote IP addr   Description
       PW-tx-fault  101   10.1.1.1        10.1.1.1         S-PE2
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 19, remote 23
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 16, send 27
    byte totals:   receive 2506, send 3098
    packet drops:  receive 0, seq error 0, send 0
```

Displaying Information About the L2VPN Multisegment Pseudowires using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **show l2vpn atom binding**
2. **show l2vpn atom vc detail**

DETAILED STEPS

Step 1 show l2vpn atom binding

Use the **show l2vpn atom binding** command to display information about the pseudowire switching point, as shown in bold in the output. (In the following examples PE1 and PE4 are the T-PE routers.)

Example:

```
Device# show l2vpn atom binding

  Destination Address: 10.1.1.1,  VC ID: 102
    Local Label:  17
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2], TTL [3]
              CV Type: LSPV [2]
    Remote Label: 16
        Cbit: 1,    VC Type: FastEthernet,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2], TTL [3]
              CV Type: LSPV [2]
        PW Switching Point:
             Vcid   local IP addr      remote IP addr      Description
             101    10.11.11.11        10.20.20.20         PW Switching Point PE3
             100    10.20.20.20        10.11.11.11         PW Switching Point PE2
```

Step 2 show l2vpn atom vc detail

Use the **show l2vpn atom vc detail** command to display status of the pseudowire switching point. In the following example, the output (shown in bold) displays the segment that is the source of the fault of the multisegment pseudowire:

Example:

```
Device# show l2vpn atom vc detail
Local interface: Se3/0/0 up, line protocol up, HDLC up
  Destination address: 12.1.1.1, VC ID: 100, VC status: down
    Output interface: Se2/0, imposed label stack {23}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:03:02, last status change time: 00:01:41
  Signaling protocol: LDP, peer 10.1.1.1:0 up
    Targeted Hello: 10.1.1.4(LDP Id) -> 10.1.1.1, LDP is UP
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRrd
      Last local dataplane   status rcvd: No fault
      Last local SSS circuit status rcvd: No fault
```

```
    Last local SSS circuit status sent: DOWN(PW-tx-fault)
    Last local  LDP TLV    status sent: No fault
    Last remote LDP TLV    status rcvd: DOWN(PW-tx-fault)
     PW Switching Point:
     Fault type  Vcid   local IP addr   remote IP addr   Description
     PW-tx-fault  101   10.1.1.1        10.1.1.1        S-PE2
    Last remote LDP ADJ    status rcvd: No fault
  MPLS VC labels: local 19, remote 23
  Group ID: local 0, remote 0
  MTU: local 1500, remote 1500
  Remote interface description:
Sequencing: receive disabled, send disabled
VC statistics:
  packet totals: receive 16, send 27
  byte totals:   receive 2506, send 3098
  packet drops:  receive 0, seq error 0, send 0
```

Performing ping mpls and trace mpls Operations on the L2VPN Multisegment Pseudowires

You can use the **ping mpls** and **trace mpls**commands to verify that all the segments of the MPLS multisegment pseudowire are operating.

You can use the **ping mpls** command to verify connectivity at the following pseudowire points:

- From one end of the pseudowire to the other
- From one of the pseudowires to a specific segment
- The segment between two adjacent S-PE routers

You can use the **trace mpls**command to verify connectivity at the following pseudowire points:

- From one end of the pseudowire to the other
- From one of the pseudowires to a specific segment
- The segment between two adjacent S-PE routers
- A range of segments

SUMMARY STEPS

1. **ping mpls pseudowire** *destination-address vc-id* [**segment** *segment-number*]
2. **trace mpls pseudowire** *destination-address vc-id* **segment** *segment-number segment-number*

DETAILED STEPS

Step 1 **ping mpls pseudowire** *destination-address vc-id* [**segment** *segment-number*]

Where:

- *destination-address* is the address of the S-PE router, which is the end of the segment from the direction of the source.

- *vc-id* is the VC ID of the segment from the source to the next PE router.
- **segment** *segment-number is optional and specifies the segment you want to ping.*

The following examples use the topology shown in the second figure above :

- To perform an end-to-end ping operation from T-PE1 to T-PE2, enter the following command:

ping mpls pseudowire *<addr-of-S-PE1> <vc-id between T-PE1 and S-PE1>*

- To perform a ping operation from T-PE1 to segment 2, enter the following command:

ping mpls pseudowire *<addr-of-S-PE1> <vc-id between T-PE1 and S-PE1>* **segment 2**

Example:

Step 2 **trace mpls pseudowire** *destination-address vc-id* **segment** *segment-number segment-number*

Where:

- *destination-address is the address of the next S-PE router from the original of the trace.*
- *vc-id* is the VC ID of the segment from which the **trace** command is issued.
- *segment-number* indicates the segment upon which the trace operation will act. If you enter two segment numbers, the traceroute operation will perform a trace on that range of routers.

The following examples use the topology shown in the second figure above :

- To perform a trace operation from T-PE1 to segment 2 of the multisegment pseudowire, enter the following command:

trace mpls pseudowire *<addr-of-S-PE1> <vc-id between T-PE1 and S-PE1>* **segment 2**

This example performs a trace from T-PE1 to S-PE2.

- To perform a trace operation on a range of segments, enter the following command. This example performs a trace from S-PE2 to T-PE2.

trace mpls pseudowire *<addr-of-S-PE1> <vc-id between T-PE1 and S-PE1>* **segment 2 4**

The following command performs a trace operation on S-PE router 10.10.10.9, on segment 1 and then on segment 2:

Example:

```
router# trace mpls pseudowire 10.10.10.9 220 segment 1
Tracing MS-PW segments within range [1-1] peer address 10.10.10.9 and timeout 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
L 1 10.10.9.9 0 ms [Labels: 18 Exp: 0]
    local 10.10.10.22 remote 10.10.10.9 vc id 220
router# trace mpls pseudowire 10.10.10.9 220 segment 2
Tracing MS-PW segments within range [1-2] peer address 10.10.10.9 and timeout 2 seconds
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
```

```
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
L 1 10.10.9.9 4 ms [Labels: 18 Exp: 0]
    local 10.10.10.22 remote 10.10.10.9 vc id 220
! 2 10.10.3.3 4 ms [Labels: 16 Exp: 0]
    local 10.10.10.9 remote 10.10.10.3 vc id 220
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
Layer 2 VPNS	• Any Transport over MPLS • *L2VPN Pseudowire Switching* • MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV

Standards

Standard	Title
RFC 4777	Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for L2VPN Multisegment Pseudowires

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 109: Feature Information for L2VPN Multisegment Pseudowires

Feature Name	Releases	Feature Information
MPLS OAM Support for Multisegment Pseudowires	Cisco IOS XE Release 2.3 Cisco IOS XE Release 3.5S	The L2VPN Multisegment Pseudowires feature enables you to configure two or more Layer 2 pseudowire segments that function as a single pseudowire. The L2VPN Multisegment Pseudowires feature span multiple cores or autonomous systems of the same or different carrier networks. In isco IOS XE Release 2.3, this feature was introduced and implemented on the Cisco ASR 1000 Series Routers. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **description** (l2 vfi), **ping mpls**, **show mpls l2transport binding**, **show mpls l2transport vc**, **switching tlv**, **trace mpls**.

CHAPTER 49

MPLS Quality of Service

The MPLS Quality of Service feature (formerly named as the MPLS CoS feature) enables you to provide differentiated services across an MPLS network. To satisfy a wide range of networking requirements, you can specify the class of service applicable to each transmitted IP packet. Different classes of service can be established for IP packets by setting the IP precedence bit in the header of each packet.

Prerequisites for MPLS Quality of Service

To use MPLS CoS to full advantage in your network, the following functionality must be supported:

- Multiprotocol Label Switching (MPLS)—MPLS is the standardized label switching protocol defined by the Internet Engineering Task Force (IETF).
- Cisco Express Forwarding—Cisco Express Forwarding is an advanced Layer 3 IP switching technology that optimizes performance and scalability in networks that handle large volumes of traffic and that exhibit dynamic traffic patterns.
- Asynchronous Transfer Mode (ATM)—ATM signaling support is required if you are using ATM interfaces in your network.

  If you are using only packet interfaces in your network, ATM functionality is not needed.
- QoS features:
  - Weighted fair queueing (WFQ)—Used on non-GSR platforms, WFQ is a dynamic scheduling method that allocates bandwidth fairly to all network traffic.

    WFQ applies priorities, or weights, to traffic to classify the traffic into flows and determine how much bandwidth to allow each flow. WFQ moves interactive traffic to the front of a queue to reduce response time and fairly shares the remaining bandwidth among high-bandwidth flows.
  - Weighted random early detection (WRED)—WRED is a congestion avoidance mechanism that extends RED functionality by allowing different RED parameters to be configured per IP precedence value.

IP precedence bits, contained in the type of service (ToS) octet in the IP packet header, are used to denote the relative importance or priority of an IP packet. WRED uses these IP precedence values to classify packets into different discard priorities or classes of service.

- Modified deficit round robin (MDRR)—Used only on GSR platforms, MDRR is a traffic class prioritization mechanism that incorporates emission priority as a facet of quality of service. MDRR is similar in function to WFQ on non-GSR platforms.

  In MDRR, IP traffic is mapped to different classes of service queues. A group of queues is assigned to each traffic destination. On the transmit side of the platform, a group of queues is defined on a per-interface basis; on the receive side of the platform, a group of queues is defined on a per-destination basis. IP packets are then mapped to these queues, based on their IP precedence value.

  These queues are serviced on a round-robin basis, except for a queue that has been defined to run in either of two ways: strict priority mode or alternate priority mode.

  In strict priority mode, the high priority queue is serviced whenever it is not empty; this ensures the lowest possible delay for high priority traffic. In this mode, however, the possibility exists that other traffic might not be serviced for long periods of time if the high priority queue is consuming most of the available bandwidth.

  In alternate priority mode, the traffic queues are serviced in turn, alternating between the high priority queue and the remaining queues.

- Committed access rate (CAR)—CAR is a QoS feature that limits the input or output transmission rate on an interface and classifies packets by setting the IP precedence value or the QoS group in the IP packet header.

Information About MPLS Quality of Service

MPLS Quality of Service Overview

MPLS CoS functionality enables network administrators to provide differentiated services across an MPLS network. Network administrators can satisfy a wide range of networking requirements by specifying the class of service applicable to each transmitted IP packet. Different classes of service can be established for IP packets by setting the IP precedence bit in the header of each packet.

MPLS CoS supports the following differentiated services in an MPLS network:

- Packet classification
- Congestion avoidance
- Congestion management

The table below describes the MPLS CoS services and functions.

Table 110: MPLS CoS Services and Functions

Service	CoS Function	Description
Packet classification	Committed access rate (CAR). Packets are classified at the edge of the network before labels are assigned.	CAR uses the type of service (ToS) bits in the IP header to classify packets according to input and output transmission rates. CAR is often configured on interfaces at the edge of a network in order to control traffic flowing into or out of the network. You can use CAR classification commands to classify or reclassify a packet.
Congestion avoidance	Weighted random early detection (WRED). Packet classes are differentiated based on drop probability.	WRED monitors network traffic to anticipate and prevent congestion at common network and internetwork bottlenecks. WRED can selectively discard lower priority traffic when an interface becomes congested; WRED can also provide differentiated performance characteristics for different classes of service.
Congestion management	Weighted fair queueing WFQ) for non-GSR platform. Packet classes are differentiated based on bandwidth requirements and finite delay characteristics. Modified deficit round robin (MDRR) for GSR platforms.	WFQ is an automated scheduling system that ensures fair bandwidth allocation to all network traffic. WFQ uses weights (priorities) to determine how much bandwidth each class of traffic is allocated. MDRR, similar in function to WFQ for non-GSR platforms, is a traffic prioritization scheme that maps IP traffic to different classes of service queues, based on the IP precedence value of each packet. The queues are then serviced on a round-robin basis.

MPLS CoS enables you to duplicate Cisco IP CoS (Layer 3) features as closely as possible in MPLS devices, including label edge switch routers (edge LSRs) and label switch routers (LSRs). MPLS CoS functions map nearly one-for-one to IP CoS functions on all types of interfaces.

Tag Switching and MPLS Terminology

The table below lists the existing legacy tag switching terms and the new, equivalent Multiprotocol Label Switching (MPLS) IETF terms used in this document and other related Cisco publications.

Table 111: Tag Switching Terms and Equivalent MPLS Terms

Old Designation	New Designation
Tag switching	Multiprotocol Label Switching
Tag (short for tag switching)	MPLS
Tag (item or packet)	Label
TDP (Tag Distribution Protocol)	LDP (Label Distribution Protocol). Cisco TDP and LDP (MPLS Label Distribution Protocol) closely parallel each other in function, but differ in detail, such as message formats and the commands required to configure the respective protocols and to monitor their operation
Tag switched	Label switched
TFIB (tag forwarding information base)	LFIB (label forwarding information base)
TSR (tag switching router)	LSR (label switching router)
TVC (tag VC, tag virtual circuit)	LVC (label VC, label virtual circuit)
TSP (tag switch path)	LSP (label switch path)

LSRs Used at the Edge of an MPLS Network

Label switching routers (LSRs) used at the edge of a Multiprotocol Label Switching (MPLS) network backbone are devices running MPLS software. The edge LSRs can be at the ingress or the egress of the network.

At the ingress of an MPLS network, devices process packets as follows:

1. IP packets enter the edge of the MPLS network at the edge LSR.
2. The edge LSR uses a classification mechanism such as the Modular Quality of Service Command-Line Interface (CLI) (MQC) to classify incoming IP packets and set the IP precedence value. Alternatively, IP packets can be received with the IP precedence value already set.
3. For each packet, the device performs a lookup on the IP address to determine the next-hop LSR.
4. The appropriate label is inserted into the packet, and the IP precedence bits are copied into the MPLS EXP bits in the label header.
5. The labeled packets are forwarded to the appropriate output interface for processing.
6. The packets are differentiated by class according to one of the following:
   - Drop probability—Weighted random early detection (WRED)
   - Bandwidth allocation and delay—Class-based weighted fair queueing (CBWFQ)

In either case, LSRs enforce the defined differentiation by continuing to employ WRED or CBWFQ on every ingress device.

At the egress of an MPLS network, devices process packets as follows:

1. MPLS-labeled packets enter the edge LSR from the MPLS network backbone.
2. The MPLS labels are removed and IP packets may be (re)classified.
3. For each packet, the device performs a lookup on the IP address to determine the packet's destination and forwards the packet to the destination interface for processing.
4. The packets are differentiated by the IP precedence values and treated appropriately, depending on the WRED or CBWFQ drop probability configuration.

LSRs Used at the Core of an MPLS Network

Label switching routers (LSRs) used at the core of a Multiprotocol Label Switching (MPLS) network are devices running MPLS software. These devices at the core of an MPLS network process packets as follows:

1. MPLS labeled packets coming from the edge devices or other core devices enter the core device.
2. A lookup is done at the core device to determine the next hop LSR.
3. An appropriate label is placed (swapped) on the packet and the MPLS EXP bits are copied.
4. The labeled packet is then forwarded to the output interface for processing.
5. The packets are differentiated by the MPLS EXP field marking and treated appropriately, depending on the weighted early random detection (WRED) and class-based weighted fair queueing (CBWFQ) configuration.

Benefits of MPLS CoS in IP Backbones

You realize the following benefits when you use MPLS CoS in a backbone consisting of IP devices running Multiprotocol Label Switching (MPLS):

- Efficient resource allocation—Weighted fair queueing (WFQ) is used to allocate bandwidth on a per-class and per-link basis, thereby guaranteeing a percentage of link bandwidth for network traffic.
- Packet differentiation—When IP packets traverse an MPLS network, packets are differentiated by mapping the IP precedence bits of the IP packets to the MPLS CoS bits in the MPLS EXP field. This mapping of bits enables the service provider to maintain end-to-end network guarantees and meet the provisions of customer service level agreements (SLAs).
- Future service enhancements—MPLS CoS provides building blocks for future service enhancements (such as virtual leased lines) by meeting bandwidth requirements.

How to Configure MPLS Quality of Service

Configuring WRED

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **random-detect**
5. **random-detect precedence** *min-threshold max-threshold mark-probability*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# gigabitethernet0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**random-detect** **Example:** `Device(config-if)# random-detect`	Configures the interface to use weighted random early detection/distributed weighted random early detection (WRED/DWRED).
Step 5	**random-detect precedence** *min-threshold max-threshold mark-probability* **Example:** `Device(config-if)# random-detect precedence 0 32 256 100`	Configures WRED/DWRED parameters per precedence value.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying WRED

To verify weighted random early detection (WRED), use a command of the form shown in the following table. This example is based on "Device2" in the network topology shown in the figure in the configuration examples section.

SUMMARY STEPS

1. **show queueing interface** *subinterface*

DETAILED STEPS

show queueing interface *subinterface*

Example:

```
Device2# show queueing interface gigabitethernet6/0/0
```

Verifies the WRED configuration on the specified interface.

```
Device2# show queueing interface gigabitethernet6/0/0

Interface Gige6/0/0 queueing strategy:random early detection (WRED)
     Exp-weight-constant:9 (1/512)
     Mean queue depth:0

     Class   Random       Tail    Minimum    Maximum     Mark
               drop       drop  threshold  threshold  probability
       0         85          0         20         40     1/10
       1         22          0         22         40     1/10
       2          0          0         24         40     1/10
       3          0          0         26         40     1/10
       4          0          0         28         40     1/10
       5          0          0         31         40     1/10
       6          0          0         33         40     1/10
       7          0          0         35         40     1/10
     rsvp         0          0         37         40     1/10
```

Configuring CAR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *name*
4. **rate-limit input** [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *conform-action* **exceed-action** *exceed-action*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *name* **Example:** `Device(config)# interface gigabitethernet`	Designates the input interface, and enters interface configuration mode.
Step 4	**rate-limit input** [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *conform-action* **exceed-action** *exceed-action* **Example:** `Device(config-if)# rate-limit input access-group 101 496000 32000 64000 conform-action set-prec-transmit 4`	Specifies the action to take on packets during label imposition.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the CAR Configuration

SUMMARY STEPS

1. **show interfaces** *slot/port* **rate-limit**

DETAILED STEPS

show interfaces *slot/port* **rate-limit**

Example:

```
Device2# show interfaces fe1/1/1 rate-limit
```

Verifies the CAR configuration, use a command of the following form.

```
Device2# show interfaces fe1/1/1 rate-limit

FastEthernet1/1/1
   Input
     matches:access-group 101
       params: 496000 bps, 32000 limit, 64000 extended limit
       conformed 2137 packets, 576990 bytes; action:set-prec-transmit 4
```

```
exceeded 363 packets, 98010 bytes; action:set-prec-transmit 0
last packet:11788ms ago, current burst:39056 bytes
last cleared 00:01:18 ago, conformed 58000 bps, exceeded 10000 bps
```

Configuring CBWFQ

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name*
4. **match** *type number*
5. **policy-map** *policy-map-name*
6. **class** *class-map-name*
7. **bandwidth** *number*
8. **interface** *type number*
9. **service-policy output** *policy-map-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* **Example:** `Device(config)# class-map class-map-1`	Creates a class map, and enters class-map configuration mode.
Step 4	**match** *type number* **Example:** `Device(config-cmap)# match ip precedence 0 1`	Specifies the traffic on which the class map is to match.
Step 5	**policy-map** *policy-map-name* **Example:** `Device(config-cmap)# policy-map outputmap`	Creates a policy map, and enters policy-map configuration mode.
Step 6	**class** *class-map-name* **Example:**	Associates the class map with the policy map.

	Command or Action	Purpose
	`Device(config-pmap)# class class-map-1`	
Step 7	**bandwidth** *number* **Example:** `Device(config-pmap-c)# bandwidth 10000`	Associates the bandwidth (CBWFQ) action to act on traffic matched by the class map, and enters policy-map class configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config-pmap-c)# interface gigabitethernet0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 9	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output outputmap`	Assigns the policy map to an interface.
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the CBWFQ Configuration

SUMMARY STEPS

1. **show policy-map interface** *type number*

DETAILED STEPS

show policy-map interface *type number*

Example:

```
Device5# show policy-map interface fe5/1/0
```

Verifies the class-based weighted fair queueing (CBWFQ) configuration, use a command of the following form. This example is based on “Device 5” in the network topology shown in the figure in the configuration examples section.

```
Device5# show policy-map interface fe5/1/0

  FastEthernet5/1/0
   service-policy output:outputmap
     class-map:prec_01 (match-all)
       522 packets, 322836 bytes
       5 minute rate 1000 bps
       match:ip precedence 0  1
       queue size 0, queue limit 1356
       packet output 522, packet drop 0
       tail/random drop 0, no buffer drop 0, other drop 0
       bandwidth:class-based wfq, weight 10
       random-detect:
         Exp-weight-constant:9 (1/512)
         Mean queue depth:0
```

```
Class Random       Tail    Minimum    Maximum     Mark      Output
        drop       drop  threshold  threshold  probability packets
0          0          0       3390       6780     1/10          522
1          0          0       3813       6780     1/10            0
2          0          0       4236       6780     1/10            0
3          0          0       4659       6780     1/10            0
4          0          0       5082       6780     1/10            0
5          0          0       5505       6780     1/10            0
6          0          0       5928       6780     1/10            0
7          0          0       6351       6780     1/10            0

  class-map:prec_23 (match-all)
    0 packets, 0 bytes
    5 minute rate 0 bps
    match:ip precedence 2  3
    queue size 0, queue limit 0
    packet output 0, packet drop 0
    tail/random drop 0, no buffer drop 0, other drop 0
    bandwidth:class-based wfq, weight 15
    random-detect:
      Exp-weight-constant:9 (1/512)
      Mean queue depth:0
Class Random       Tail    Minimum    Maximum     Mark      Output
        drop       drop  threshold  threshold  probability packets
0          0          0          0          0    1/10            0
1          0          0          0          0    1/10            0
2          0          0          0          0    1/10            0
3          0          0          0          0    1/10            0
4          0          0          0          0    1/10            0
5          0          0          0          0    1/10            0
6          0          0          0          0    1/10            0
7          0          0          0          0    1/10            0

  class-map:prec_45 (match-all)
    2137 packets, 576990 bytes
    5 minute rate 16000 bps
    match:ip precedence 4  5
    queue size 0, queue limit 2712
    packet output 2137, packet drop 0
    tail/random drop 0, no buffer drop 0, other drop 0
    bandwidth:class-based wfq, weight 20
    random-detect:
      Exp-weight-constant:9 (1/512)
      Mean queue depth:0
Class Random       Tail    Minimum    Maximum     Mark      Output
        drop       drop  threshold  threshold  probability packets
0          0          0       3390       6780    1/10            0
1          0          0       3813       6780    1/10            0
2          0          0       4236       6780    1/10            0
3          0          0       4659       6780    1/10            0
4          0          0       5082       6780    1/10         2137
5          0          0       5505       6780    1/10            0
6          0          0       5928       6780    1/10            0
7          0          0       6351       6780    1/10            0

  class-map:prec_67 (match-all)
    0 packets, 0 bytes
    5 minute rate 0 bps
    match:ip precedence 6  7
    queue size 0, queue limit 0
    packet output 0, packet drop 0
    tail/random drop 0, no buffer drop 0, other drop 0
    bandwidth:class-based wfq, weight 25
    random-detect:
```

```
        Exp-weight-constant:9 (1/512)
        Mean queue depth:0
   Class Random       Tail    Minimum    Maximum     Mark      Output
           drop       drop  threshold  threshold  probability packets
   0          0          0          0          0    1/10             0
   1          0          0          0          0    1/10             0
   2          0          0          0          0    1/10             0
   3          0          0          0          0    1/10             0
   4          0          0          0          0    1/10             0
   5          0          0          0          0    1/10             0
   6          0          0          0          0    1/10             0
   7          0          0          0          0    1/10             0

     class-map:class-default (match-any)
       0 packets, 0 bytes
       5 minute rate 0 bps
       match:any
         0 packets, 0 bytes
         5 minute rate 0 bps
       queue size 0, queue limit 4068
       packet output 90, packet drop 0
       tail/random drop 0, no buffer drop 0, other drop 0
Device5#
Device5# show queueing interface fa1/1/0

Interface FastEthernet1/1/0 queueing strategy:VIP-based fair queueing
  FastEthernet1/1/0 queue size 0
         pkts output 2756, wfq drops 0, nobuffer drops 0
  WFQ:aggregate queue limit 13561 max available buffers 13561

      Class 0:weight 30 limit 4068 qsize 0 pkts output 97 drops 0
      Class 2:weight 10 limit 1356 qsize 0 pkts output 522 drops 0
      Class 3:weight 15 limit 0 qsize 0 pkts output 0 drops 0
      Class 4:weight 20 limit 2712 qsize 0 pkts output 2137 drops 0
      Class 5:weight 25 limit 0 qsize 0 pkts output 0 drops 0 \
```

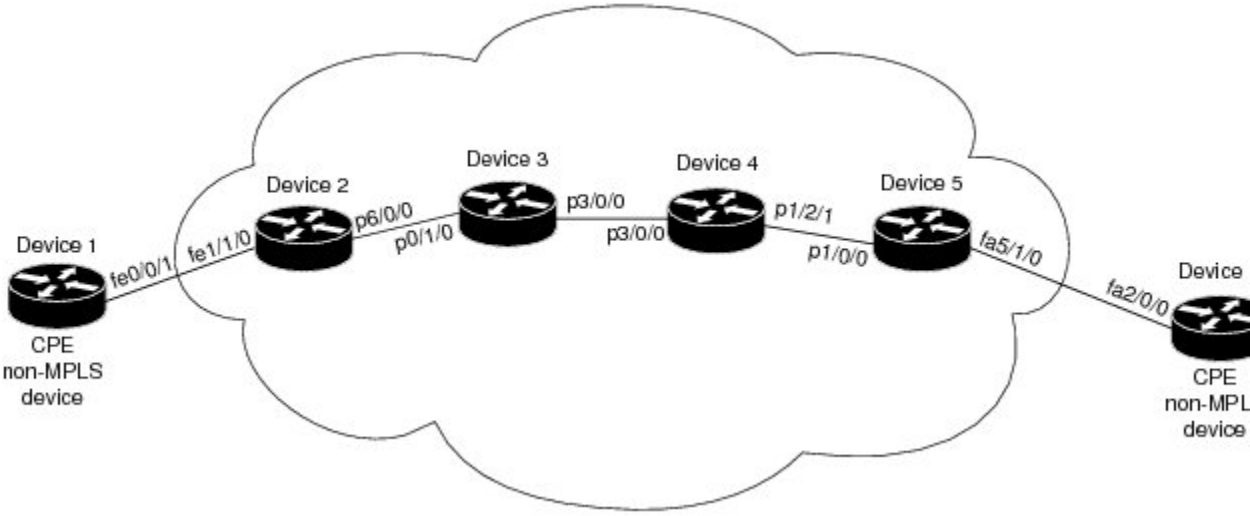
What to do next

-

Configuration Examples for MPLS Quality of Service

The configuration examples are based on the sample network topology shown in the figure below.

Figure 69: Sample Network Topology for Configuring MPLS CoS on Device Interfaces



Example: Configuring Cisco Express Forwarding

Cisco Express Forwarding must be running on all devices in the Multiprotocol Label Switching (MPLS) network for MPLS CoS to work. To enable Cisco Express Forwarding, use one of the following commands:

```
Device(config)# ip cef
```

or

```
Device(config)# ip cef distributed
```

Example: Running IP on Device 1

The following commands enable IP routing on Device 1. All devices in the figure must have IP enabled. Device 1 is not part of the Multiprotocol Label Switching (MPLS) network.

```
!
ip routing
!
hostname R1
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface FastEthernet0/0/1
 ip address 10.0.0.1 255.0.0.0
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.1 0.255.255.255 area 100
```

Example: Running MPLS on Device 2

Device 2 is a label edge router. Cisco Express Forwarding and Multiprotocol Label Switching (MPLS) must be enabled on this device. Committed access rate (CAR) is also configured on Device 2 and Fast Ethernet interface 1/1/3. The CAR policy used at Fast Ethernet interface 1/1/0 acts on incoming traffic matching access-list 101. If the traffic rate is less than the committed information rate (in this example, 496000), the traffic will be sent with IP precedence 4. Otherwise, this traffic will be sent with IP precedence 0.

```
!
ip routing
!
hostname R2
!
ip cef
mpls ip
tag-switching advertise-tags
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
!
interface FastEthernet1/1/0
 ip address 10.0.0.2 255.0.0.0
 rate-limit input access-group 101 496000 32000 64000 conform-action set-prec-transmit 4
exceed-action set-prec-transmit 0
!
interface POS6/0/0
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 random-detect
 clock source internal
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
 network 11.0.1.0 0.255.255.255 area 100
!
access-list 101 permit ip host 10.10.1.1 any
```

Example: Running MPLS on Device 3

Device 3 is running Multiprotocol Label Switching (MPLS). Cisco Express Forwarding and MPLS must be enabled on this device.

```
!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R3
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
!
interface POS0/1/0
 ip address 10.0.0.2 255.0.0.0
 mpls label protocol ldp
 mpls ip
```

```
 crc 16
!
interface POS3/0/0
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 crc 16
 clock source internal
 tx-cos stm16-rx
!
router ospf 100
 network 10.0.1.0 0.255.255.255 area 100
 network 10.0.0.1 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
!
cos-queue-group stm16-rx
 precedence 0 random-detect-label 0
 precedence 0 queue 0
 precedence 1 queue 1
 precedence 1 random-detect-label 1
 precedence 2 queue 2
 precedence 2 random-detect-label 2
 precedence 3 random-detect-label 2
 precedence 4 random-detect-label 2
 precedence 5 random-detect-label 2
 precedence 6 random-detect-label 2
 precedence 7 queue low-latency
 precedence 7 random-detect-label 2
 random-detect-label 0 250 1000 1
 random-detect-label 1 500 1250 1
 random-detect-label 2 750 1500 1
 queue 0 50
 queue 1 100
 queue 2 150
 queue low-latency alternate-priority 500
```

Example: Running MPLS on Device 4

Device 4 is running Multiprotocol Label Switching (MPLS). Cisco Express Forwarding and MPLS must be enabled on this device.

```
!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R4
!
interface Loopback0
 ip address 10.0.0.0 255.255.255.255
!
interface POS1/2/1
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 crc 16
 clock source internal
 tx-cos stm16-rx
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 100
```

```
 network 10.1.0.0 0.255.255.255 area 100
 network 10.0.1.0 0.255.255.255 area 100
!
cos-queue-group stm16-rx
 precedence 0 queue 0
 precedence 0 random-detect-label 0
 precedence 1 queue 1
 precedence 1 random-detect-label 1
 precedence 2 queue 2
 precedence 2 random-detect-label 2
 precedence 3 random-detect-label 2
 precedence 4 random-detect-label 2
 precedence 5 random-detect-label 2
 precedence 6 random-detect-label 2
 precedence 7 queue low-latency
 random-detect-label 0 250 1000 1
 random-detect-label 1 500 1250 1
 random-detect-label 2 750 1500 1
 queue 0 50
 queue 1 100
 queue 2 150
 queue low-latency alternate-priority 200
```

Example: Running MPLS on Device 5

Device 5 is running Multiprotocol Label Switching (MPLS). Cisco Express Forwarding and MPLS must be enabled on this device. Device 5 has class-based weighted fair queueing (CBWFQ) enabled on Fast Ethernet interface 5/1/0. In this example, class maps are created, matching packets with various IP precedence values. These class maps are then used in a policy map named “outputmap,” where CBWFQ is assigned to each class. Finally, the policy map is assigned to the outbound Fast Ethernet interface 5/1/0.

```
!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R5
!
!
class-map match-all prec_01
  match ip precedence 0  1
class-map match-all prec_23
  match ip precedence 2  3
class-map match-all prec_45
  match ip precedence 4  5
class-map match-all prec_67
  match ip precedence 6  7
!
!
policy-map outputmap
  class prec_01
    bandwidth 10000
    random-detect
  class prec_23
    bandwidth 15000
    random-detect
  class prec_45
    bandwidth 20000
    random-detect
```

```
  class prec_67
    bandwidth 25000
    random-detect
!
ip cef distributed
!
interface Loopback0
 ip address 10.0.0.0 255.255.255.255
 no ip directed-broadcast
!
interface POS1/1/0
 ip address 10.0.0.2 255.0.0.0
 ip route-cache distributed
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet5/1/0
 ip address 10.0.0.1 255.0.0.0
 ip route-cache distributed
 full-duplex
 service-policy output outputmap
!
router ospf 100
 network 10.1.0.0 0.255.255.255 area 100
 network 10.0.1.0 0.255.255.255 area 100
 network 10.0.0.1 0.255.255.255 area 100
```

Example: Running IP on Device 6

Device 6 is running IP. Cisco Express Forwarding must be enabled on this device. Device 6 is not part of the Multiprotocol Label Switching (MPLS) network.

```
!
ip routing
!
hostname R6
!
ip cef distributed
!
interface Loopback0
 ip address 10.0.0.0 255.255.255.255
!
interface FastEthernet2/0/0
 ip address 10.0.0.2 255.0.0.0
 ip route-cache distributed
 full-duplex
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
!
```

Additional References for MPLS Quality of Service

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS QoS commands	Cisco IOS Quality of Service Solutions Command Reference Cisco IOS Multiprotocol Label Switching Command Reference

MIBs

MIB	MIBs Link
• CISCO-WRED-MIB • CISCO-CAR-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for MPLS Quality of Service

Table 112: Feature Information for MPLS Quality of Service

Feature Name	Releases	Feature Information
MPLS Quality of Service	12.0(5)T 12.0(11)T 12.0(22)S 12.2(17b)SXA 12.2(8)T Cisco IOS XE Release 2.1	The MPLS Quality of Service feature (formerly named as the MPLS CoS feature) enables you to provide differentiated services across an MPLS network. To satisfy a wide range of networking requirements, you can specify the class of service applicable to each transmitted IP packet. Different classes of service can be established for IP packets by setting the IP precedence bit in the header of each packet No new or modified commands were introduced.

CHAPTER 50

QoS Policy Support on L2VPN ATM PVPs

This feature enables you to configure Quality of Service (QoS) service policies in ATM permanent virtual path (PVP) mode for Layer 2 Virtual Private Networks (L2VPNs).

Prerequisites for QoS Policy Support on L2VPN ATM PVPs

Before configuring QoS policies on L2VPN ATM PVPs, you should understand the concepts and configuration instructions in the following documents:

- Any Transport over MPLS
- Applying QoS Features Using the MQC

Restrictions for QoS Policy Support on L2VPN ATM PVPs

- Queueing-based policies are not supported in ATM PVP mode and virtual circuit (VC) mode at the same time under the same main interface. However, nonqueueing policies can be mixed. For example, you can configure a nonqueueing policy in PVP mode and configure queueing policies on in VC mode under the same main interface. Similarly, you can configure a queueing policy in PVP mode and configure nonqueueing policies in VC mode in the input or output direction.
- ATM PVP mode does not support sessions.
- When you enable a policy in PVP mode, do not configure ATM rates on the VCs that are part of the PVP. The VCs should be unspecified bit rate (UBR) VCs only.
- If VCs are part of a PVP that has a policy configured, you cannot configure ATM VC traffic shaping.
- You cannot configure a queueing policy on an ATM PVP with UBR.

- You cannot configure queueing-based policies with UBR traffic shaping.

Information About QoS Policy Support on L2VPN ATM PVPs

The MQC Structure

The MQC structure allows you to define a traffic class, create a traffic policy, and attach the traffic policy to an interface.

The MQC structure consists of the following three high-level steps.

SUMMARY STEPS

1. Define a traffic class by using the **class-map**command. A traffic class is used to classify traffic.
2. Create a traffic policy by using the **policy-map** command. (The terms traffic policy and policy map are often synonymous.) A traffic policy (policy map) contains a traffic class and one or more QoS features that will be applied to the traffic class. The QoS features in the traffic policy determine how to treat the classified traffic.
3. Attach the traffic policy (policy map) to the interface by using the **service-policy** command.

DETAILED STEPS

Step 1 Define a traffic class by using the **class-map**command. A traffic class is used to classify traffic.

Step 2 Create a traffic policy by using the **policy-map** command. (The terms traffic policy and policy map are often synonymous.) A traffic policy (policy map) contains a traffic class and one or more QoS features that will be applied to the traffic class. The QoS features in the traffic policy determine how to treat the classified traffic.

Step 3 Attach the traffic policy (policy map) to the interface by using the **service-policy** command.

Elements of a Traffic Class

A traffic class contains three major elements: a traffic class name, a series of match commands, and, if more than one match command is used in the traffic class, instructions on how to evaluate these match commands.

The match commands are used for classifying packets. Packets are checked to determine whether they meet the criteria specified in the match commands; if a packet meets the specified criteria, that packet is considered a member of the class. Packets that fail to meet the matching criteria are classified as members of the default traffic class.

Elements of a Traffic Policy

A traffic policy contains three elements: a traffic policy name, a traffic class (specified with the class command), and the command used to enable the QoS feature.

The traffic policy (policy map) applies the enabled QoS feature to the traffic class once you attach the policy map to the interface (by using the service-policy command).

Note A packet can match only one traffic class within a traffic policy. If a packet matches more than one traffic class in the traffic policy, the first traffic class defined in the policy will be used.

How to Configure QoS Policy Support on L2VPN ATM PVPs

Enabling a Service Policy in ATM PVP Mode

You can enable a service policy in ATM PVP mode. You can also enable a service policy on PVP on a multipoint subinterface.

Note The **show policy-map interface** command does not display service policy information for ATM interfaces.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **atm pvp** *vpi* **l2transport**
5. **service-policy** [**input** | **output**] *policy-map-name*
6. **xconnect** *peer-router-id vcid* encapsulation mpls
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Defines the interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells and enters l2transport PVP configuration mode. • The **l2transport**keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	**service-policy** [**input** | **output**] *policy-map-name* **Example:** `Router(config-if-atm-l2trans-pvp)# service policy input pol1`	Enables a service policy on the specified PVP.
Step 6	**xconnect** *peer-router-id vcid* encapsulation mpls **Example:** `Router(config-if-atm-l2trans-pvp)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. • The syntax for this command is the same as for all other Layer 2 transports.

Step 7	**end**
	Example:
	`Router(config-if-atm-l2trans-pvp)#`
	`end`
		Exits l2transport PVP configuration mode and returns to privileged EXEC mode.

Enabling a Service Policy in ATM PVP Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

You can enable a service policy in ATM PVP mode. You can also enable a service policy on PVP on a multipoint subinterface.

Note The **show policy-map interface** command does not display service policy information for ATM interfaces.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **atm pvp** *vpi* **l2transport**
5. **service-policy** [**input** | **output**] *policy-map-name*
6. **end**
7. **interface pseudowire** *number*

8. **encapsulation mpls**
9. **neighbor** *peer-address vcid-value*
10. **exit**
11. **l2vpn xconnect context** *context-name*
12. **member pseudowire** *interface-number*
13. **member gigabitethernet** *interface-number*
14. **end**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Defines the interface and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells and enters l2transport PVP configuration mode. • The **l2transport**keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	**service-policy** [**input** | **output**] *policy-map-name* **Example:** `Router(config-if-atm-l2trans-pvp)# service policy input pol1`	Enables a service policy on the specified PVP.
Step 6	**end** **Example:** `Router(config-if-atm-l2trans-pvp)# end`	Exits to privileged EXEC mode.
Step 7	**interface pseudowire** *number* **Example:**	Specifies the pseudowire interface and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface pseudowire 100`	
Step 8	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 9	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 11	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 12	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 13	**member gigabitethernet** *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet0/0/0.1`	Specifies the location of the Gigabit Ethernet member interface.
Step 14	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Step 15	**end**
	Example:
	`Router(config-xconnect)#`
	`end`
		Exits xconnecrt configuration mode and returns to privileged EXEC mode.

Enabling Traffic Shaping in ATM PVP Mode

Traffic shaping commands are supported in PVP mode. For egress VP shaping, one configuration command is supported for each ATM service category. The supported service categories are constant bit rate (CBR), UBR, variable bit rate-nonreal time (VBR-NRT), and variable bit rate real-time(VBR-RT).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **atm pvp** *vpi* **l2transport**
5. Do one of the following:
   - **ubr** *pcr*
   -
   - **cbr** *pcr*
   - or
   - **vbr-nrt** *pcr scr mbs*
   - or
   - **vbr-rt** *pcr scr mbs*
6. **xconnect** *peer-router-id vcid* encapsulation mpls

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Defines the interface and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells and enters l2transport PVP configuration mode. • The **l2transport**keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	Do one of the following:	Enables traffic shaping in ATM PVP mode.

	Command or Action	Purpose
	• **ubr** *pcr* • • **cbr** *pcr* • or • **vbr-nrt** *pcr scr mbs* • or • **vbr-rt** *pcr scr mbs* **Example:** `Router(config-if-atm-l2trans-pvp)# cbr 1000`	• *pcr* = peak cell rate • *scr* = sustain cell rate • *mbs* = maximum burst size
Step 6	**xconnect** *peer-router-id vcid* encapsulation mpls **Example:** `Router(config-if-atm-l2trans-pvp)# xconnect 10.0.0.1 123 encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. • The syntax for this command is the same as for all other Layer 2 transports.

Enabling Traffic Shaping in ATM PVP Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

Traffic shaping commands are supported in PVP mode. For egress VP shaping, one configuration command is supported for each ATM service category. The supported service categories are constant bit rate (CBR), UBR, variable bit rate-nonreal time (VBR-NRT), and variable bit rate real-time(VBR-RT).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot* / *subslot* / *port* [**.** *subinterface*]
4. **atm pvp** *vpi* **l2transport**
5. Do one of the following:
   - **ubr** *pcr*
   -
   - **cbr** *pcr*
   - or
   - **vbr-nrt** *pcr scr mbs*
   - or
   - **vbr-rt** *pcr scr mbs*
6. **end**
7. **interface pseudowire** *number*
8. **encapsulation mpls**
9. **neighbor** *peer-address vcid-value*
10. **exit**
11. **l2vpn xconnect context** *context-name*

12. **member pseudowire** *interface-number*
13. **member gigabitethernet** *interface-number*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot* / *subslot* / *port* [**.** *subinterface*] **Example:** `Router(config)# interface atm1/0/0`	Defines the interface and enters interface configuration mode.
Step 4	**atm pvp** *vpi* **l2transport** **Example:** `Router(config-if)# atm pvp 1 l2transport`	Specifies that the PVP is dedicated to transporting ATM cells and enters l2transport PVP configuration mode. • The **l2transport**keyword indicates that the PVP is for cell relay. This mode is for Layer 2 transport only; it is not for regular PVPs.
Step 5	Do one of the following: • **ubr** *pcr* • • **cbr** *pcr* • or • **vbr-nrt** *pcr scr mbs* • or • **vbr-rt** *pcr scr mbs* **Example:** `Router(config-if-atm-l2trans-pvp)# cbr 1000`	Enables traffic shaping in ATM PVP mode. • *pcr* = peak cell rate • *scr* = sustain cell rate • *mbs* = maximum burst size
Step 6	**end** **Example:** `Router(config-if-atm-l2trans-pvp)# end`	Exits to privileged EXEC mode.
Step 7	**interface pseudowire** *number* **Example:**	Specifies the pseudowire interface and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface pseudowire 100`	
Step 8	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 9	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 11	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 12	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 13	**member gigabitethernet** *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet0/0/0.1`	Specifies the location of the Gigabit Ethernet member interface.
Step 14	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Enabling Traffic Shaping in ATM PVP Mode Example using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example enables traffic shaping in ATM PMP mode.

```
interface atm 1/0
 atm pvp 100 l2transport
  ubr 1000
```

```
 xconnect 10.11.11.11 777 encapsulation mpls
atm pvp 101 l2transport
 cbr 1000
 xconnect 10.11.11.11 888 encapsulation mpls
atm pvp 102 l2transport
 vbr-nrt 1200 800 128
 xconnect 10.11.11.11 999 encapsulation mpls
```

Enabling Matching of ATM VCIs

You can match on an ATM VCI or range of VCIs, using the **match atm-vci** command in class-map configuration mode.

Note When you configure the **match atm-vci**command in class-map configuration mode, you can add this class map to a policy map that can be attached only to an ATM VP.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name* [**match-all** | **match-any**]
4. **match atm-vci** *vc-id* **[- vc-id]**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* [**match-all**	**match-any**] **Example:** `Router(config)# class-map class1`	Creates a class map to be used for matching traffic to a specified class, and enters class-map configuration mode.
Step 4	**match atm-vci** *vc-id* **[- vc-id]** **Example:** `Router(config-cmap)# match atm-vci 50`	Enables packet matching on an ATM VCI or range of VCIs. The range is 32 to 65535. **Note** You can use the **match not** command to remove the match criteria.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-cmap)# end`	(Optional) Returns to privileged EXEC mode.

Configuration Examples for QoS Policy Support on L2VPN ATM PVPs

Example Enabling Traffic Shaping in ATM PVP Mode

The following example enables traffic shaping in ATM PMP mode.

```
int atm 1/0/0
   atm pvp 100 l2transport
       ubr 1000
       xconnect 10.11.11.11 777 encapsulation mpls
   atm pvp 101 l2transport
       cbr 1000
       xconnect 10.11.11.11 888 encapsulation mpls
   atm pvp 102 l2transport
       vbr-nrt 1200 800 128
       xconnect 10.11.11.11 999 encapsulation mpls
```

Example Enabling Traffic Shaping in ATM PVP Mode using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example enables traffic shaping in ATM PMP mode.

```
int atm 1/0/0
   atm pvp 100 l2transport
       ubr 1000
       interface pseudowire 100
       encapsulation mpls
       neighbor 10.0.0.1 123
!
l2vpn xconnect context A
 member pseudowire 100
 member g0/0/0.1
   atm pvp 101 l2transport
       cbr 1000
       interface pseudowire 100
       encapsulation mpls
       neighbor 10.0.0.1 123
!
l2vpn xconnect context A
 member pseudowire 100
 member g0/0/0.1
   atm pvp 102 l2transport
       vbr-nrt 1200 800 128
```

```
       interface pseudowire 100
       encapsulation mpls
       neighbor 10.0.0.1 123
!
l2vpn xconnect context A
 member pseudowire 100
 member g0/0/0.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC)	Applying QoS Features Using the MQC
Any Transport over MPLS	Any Transport over MPLS

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

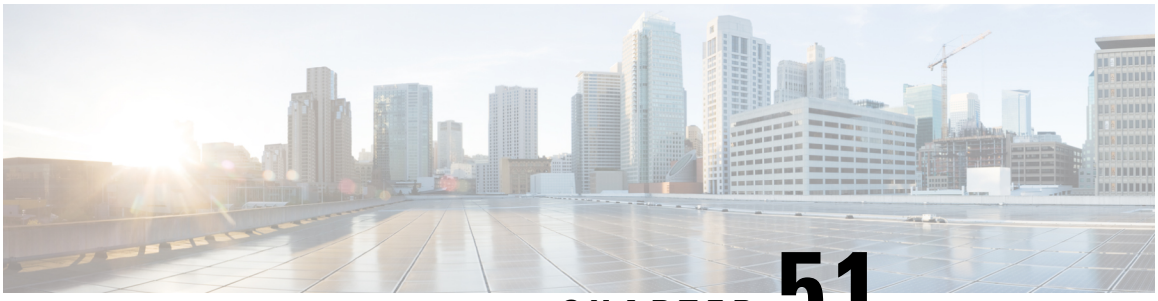
Feature Information for QoS Policy Support on L2VPN ATM PVPs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 113: Feature Information for QoS Policy Support on L2VPN ATM PVPs

Feature Name	Releases	Feature Information
QoS Policy Support on L2VPN ATM PVPs	Cisco IOS XE Release 2.3	This feature enables you to configure Quality of Service (QoS) service policies in ATM permanent virtual path (PVP) mode for Layer 2 Virtual Private Networks (L2VPNs). The following commands were introduced or modified: **cbr**, **match atm-vci**, **service-policy**, **ubr**, **vbr-nrt**, **vbr-rt**.
Cell-Based ATM Shaping per PVP	Cisco IOS XE Release 2.3	This feature was introduced for Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 51

MPLS Pseudowire Status Signaling

The MPLS Pseudowire Status Signaling feature enables you to configure the router so it can send pseudowire status to a peer router, even when the attachment circuit is down.

Prerequisites for MPLS Pseudowire Status Signaling

- Before configuring this feature, make sure that both peer routers are capable of sending and receiving pseudowire status messages.

Restrictions for MPLS Pseudowire Status Signaling

- Both peer routers must support the ability to send and receive pseudowire status messages in label advertisement and label notification messages. If both peer routers do not support pseudowire status messages, Cisco recommends that you disable the messages with the **no status** command.
- This feature is not integrated with Any Transport over MPLS (AToM) Virtual Circuit Connection Verification (VCCV).
- This feature is not integrated with Bidirectional Forwarding Detection (BFD).
- The standby and required switchover values from IETF draft-muley-pwe3-redundancy-02.txt are not supported.

Information About MPLS Pseudowire Status Signaling

How MPLS Pseudowire Status Switching Works

The pseudowire status messages are sent in label advertisement and label notification messages if the peer also supports the MPLS Pseudowire Status Signaling feature. You can issue the **show mpls l2transport vc detail** command to show that both the local and remote routers support pseudowire status messages. The following example shows the line of output to look for:

```
Router# show mpls l2transport vc detail

.

.

.

status TLV support (local/remote): enabled/supported
```

How MPLS Pseudowire Status Switching Works using the commands associated with the L2VPN Protocol-Based CLIs feature

The pseudowire status messages are sent in label advertisement and label notification messages if the peer also supports the MPLS Pseudowire Status Signaling feature. You can issue the **show l2vpn atom vc detail** command to show that both the local and remote routers support pseudowire status messages. The following example shows the line of output to look for:

```
Device# show l2vpn atom vc detail

.

.

.

status TLV support (local/remote): enabled/supported
```

When One Router Does Not Support MPLS Pseudowire Status Signaling

The peer routers must support the ability to send and receive pseudowire status messages in label advertisement and label notification messages. If one router does not support pseudowire status messages, Cisco recommends that you disable the messages with the **no status** command. This returns the router to label withdraw mode.

If the peer does not support the MPLS Pseudowire Status Signaling feature, the local router changes its mode of operation to label withdraw mode. You can issue the **show mpls l2transport vc detail** command to show

that the remote router does not support pseudowire status messages. The following example shows the line of output to look for:

```
Router# show mpls l2transport vc detail

.

.

.

status TLV support (local/remote): enabled/not supported
```

When you issue the following **debug mpls l2transport vc**commands, the messages show that the peer router does not supportthe MPLS Pseudowire Status Signaling feature and that the local router is changing to withdraw mode, as shown in bold in the following example:

Router# debug mpls l2transport vc event Router# **debug mpls l2transport vc status event** Router# **debug mpls l2transport vc status fsm** Router# **debug mpls l2transport vc ldp**

*Feb 26 13:41:40.707: AToM LDP [10.1.1.2]: Sending label withdraw msg *Feb 26 13:41:40.707: AToM LDP [10.1.1.2]: VC Type 5, mtu 1500 *Feb 26 13:41:40.707: AToM LDP [10.1.1.2]: VC ID 100, label 18 *Feb 26 13:41:40.707: AToM LDP [10.1.1.2]: Status 0x0000000A [PW Status NOT supported]

When One Router Does Not Support MPLS Pseudowire Status Signaling using the commands associated with the L2VPN Protocol-Based CLIs feature

The peer routers must support the ability to send and receive pseudowire status messages in label advertisement and label notification messages. If one router does not support pseudowire status messages, we recommend that you disable the messages with the **no status** command. This returns the router to label withdraw mode.

If the peer does not support the MPLS Pseudowire Status Signaling feature, the local router changes its mode of operation to label withdraw mode. You can issue the **show l2vpn atom vc detail** command to show that the remote router does not support pseudowire status messages. The following example shows the line of output to look for:

```
Device# show l2vpn atom vc detail

.

.

.

status TLV support (local/remote): enabled/not supported
```

When you issue the following **debug l2vpn atom vc** commands, the messages show that the peer router does not support the MPLS Pseudowire Status Signaling feature and that the local router is changing to withdraw mode, as shown in the following example:

```
Device# debug l2vpn atom vc event
Device# debug l2vpn atom vc status event
Device# debug l2vpn atom vc status fsm
Device# debug l2vpn atom vc ldp

*Feb 26 13:41:40.707: AToM LDP [110.1.1.2]: Sending label withdraw msg
*Feb 26 13:41:40.707: AToM LDP [110.1.1.2]:   VC Type   5, mtu 1500
*Feb 26 13:41:40.707: AToM LDP [110.1.1.2]:   VC ID     100, label 18
*Feb 26 13:41:40.707: AToM LDP [110.1.1.2]:   Status 0x0000000A [PW Status NOT supported]
```

Status Messages Indicating That the Attachment Circuit Is Down

When the attachment circuit is down between the two routers, the output of the **show mpls l2transport vc detail** command shows the following status:

```
Router# show mpls l2transport vc detail

.

.

.

Last remote LDP TLV    status rcvd: AC DOWN(rx,tx faults)
```

The debug messages also indicate that the attachment circuit is down, as shown in bold in the command output:

Router# debug mpls l2transport vc event Router# **debug mpls l2transport vc status event** Router# **debug mpls l2transport vc status fsm** Router# **debug mpls l2transport vc ldp**

```
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]: Received notif msg, id 88
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]:   Status    0x00000007 [PW Status]
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]:   PW Status 0x00000006 [AC DOWN(rx,tx faults)]
```

Other pseudowire status messages include not-forwarding, pw-tx-fault, and pw-rx-fault.

Status Messages Indicating That the Attachment Circuit Is Down using the commands associated with the L2VPN Protocol-Based CLIs feature

When the attachment circuit is down between the two routers, the output of the **show l2vpn atom vc detail** command shows the following status:

```
Device# show l2vpn atom vc detail

.

.

.
```

```
Last remote LDP TLV    status rcvd: AC DOWN(rx,tx faults)
```

The debug messages also indicate that the attachment circuit is down, as shown in bold in the command output:

```
Device# debug l2vpn atom  vc event
Device# debug l2vpn atom  vc status event
Device# debug l2vpn atom vc status fsm
Device# debug l2vpn atom vc ldp
```

```
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]: Received notif msg, id 88
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]:   Status    0x00000007 [PW Status]
*Feb 26 11:51:42.427: AToM LDP [10.1.1.1]:   PW Status 0x00000006 [AC DOWN(rx,tx faults)]
```

Other pseudowire status messages include not-forwarding, pw-tx-fault, and pw-rx-fault.

Message Codes in the Pseudowire Status Messages

The **debug mpls l2transport vc**and the **show mpls l2transport vc detail** commands show output that contains message codes. For example:

```
    Label/status state machine: established, LruRru

AToM MGR [10.9.9.9, 100]: S:Evt local up, LndRru->LnuRru
```

The message codes (LruRru, LndRru, and LnuRru) indicate the status of the local and remote routers. You can use the following key to interpret the message codes:

- L--local router
- R--remote router
- r or n--ready (r) or not ready (n)
- u or d--up (u) or down (d) status

The output also includes other values:

- D--Dataplane
- S--Local shutdown

Message Codes in the Pseudowire Status Messages using the commands associated with the L2VPN Protocol-Based CLIs feature

The **debug l2vpn atom vc** and the **show l2vpn atom vc detail** commands show output that contains message codes. For example:

```
    Label/status state machine: established, LruRru

AToM MGR [10.9.9.9, 100]: S:Evt local up, LndRru->LnuRru
```

The message codes (LruRru, LndRru, and LnuRru) indicate the status of the local and remote routers. You can use the following key to interpret the message codes:

L—local router

R—remote router

r or n—ready (r) or not ready (n)

u or d—up (u) or down (d) status

The output also includes other values:

D—Dataplane

S—Local shutdown

How to Configure MPLS Pseudowire Status Signaling

Enabling MPLS Pseudowire Status Signaling

Perform the following task to enable the router to send pseudowire status to a peer router even when the attachment circuit is down. If both routers do not support pseudowire status messages, then disable the messages with the **no status** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **status**
5. **encapsulation mpls**
6. **exit**
7. **exit**
8. **show mpls l2transport vc detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**pseudowire-class** *name* **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 4	**status** **Example:** `Router(config-pw)# status`	(Optional) Enables the router to send pseudowire status messages to the peer router through label advertisement and label notification messages. **Note** By default, status messages are enabled. This step is included only in case status messages have been disabled. If you need to disable status messages because both peer routers do not support this functionality, enter the **no status**command.
Step 5	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 6	**exit** **Example:** `Router(config-pw)# exit`	Exits pseudowire class configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 8	**show mpls l2transport vc detail** **Example:** `Router# show mpls l2transport vc detail`	Validates that pseudowire messages can be sent and received.

Enabling MPLS Pseudowire Status Signaling using the commands associated with the L2VPN Protocol-Based CLIs feature

Perform this task to enable the router to send pseudowire status to a peer router even when the attachment circuit is down. If both routers do not support pseudowire status messages, then disable the messages with the **no status** command.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface pseudowire** *number*
4. **status**
5. **encapsulation mpls**
6. **neighbor** *peer-address vcid-value*
7. **exit**
8. **exit**
9. **show l2vpn atom vc detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 1`	Establishes an interface pseudowire with a value that you specify and enters pseudowire configuration mode.
Step 4	**status** **Example:** `Device(config-pw)# status`	(Optional) Enables the router to send pseudowire status messages to the peer router through label advertisement and label notification messages. **Note** By default, status messages are enabled. This step is included only in case status messages have been disabled. If you need to disable status messages because both peer routers do not support this functionality, enter the **no status**command.
Step 5	**encapsulation mpls** **Example:** `Device(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 6	**neighbor** *peer-address vcid-value* **Example:** `Device(config-pw)#  neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-pw)# exit`	Exits pseudowire class configuration mode.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 9	**show l2vpn atom vc detail** **Example:** `Device# show l2vpn atom vc detail`	Validates that pseudowire messages can be sent and received.

Configuration Examples for MPLS Pseudowire Status Signaling

Example MPLS Pseudowire Status Signaling

The following example configures the MPLS Pseudowire Status Signaling feature on two PE routers. By default, status messages are enabled. The **status** command is included in this example in case status messages have been disabled.

PE1

```
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
pseudowire-class atomstatus
 encapsulation mpls
 status
!
interface GigabitEthernet0/0/1
 xconnect 10.1.1.2 123 pw-class atomstatus
```

PE2

```
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
!
pseudowire-class atomstatus
 encapsulation mpls
 status
!
interface GigabitEthernet3/3/0
 xconnect 10.1.1.1 123 pw-class atomstatus
```

Example MPLS Pseudowire Status Signaling using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example configures the MPLS Pseudowire Status Signaling feature on two PE routers. By default, status messages are enabled. The **status** command is included in this example in case status messages have been disabled.

PE1

```
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
template type pseudowire atomstatus
 encapsulation mpls
 status
!
interface pseudowire 100
 source template type pseudowire atomstatus
interface GigabitEthernet0/0/1
 service instance 300 ethernet
l2vpn xconnect context con1
 member GigabitEthernet2/1/1 service-instance 300
 member Pseudowire 100
```

PE2

```
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
!
template type pseudowire atomstatus
 encapsulation mpls
 status
!
interface Pseudowire 100
 source template type pseudowire atomstatus
interface GigabitEthernet3/3/0
 service instance 300 ethernet
l2vpn xconnect context con1
 member GigabitEthernet2/1/1 service-instance 300
 member Pseudowire 100
```

Example Verifying That Both Routers Support Pseudowire Status Messages

You can issue the **show mpls l2transport vc detail** command to show that both the local and remote routers support pseudowire status messages. The following example shows the line of output to look for:

```
Router# show mpls l2transport vc detail

.

.

.
```

```
status TLV support (local/remote): enabled/supported
```

Example Verifying That Both Routers Support Pseudowire Status Messages using the commands associated with the L2VPN Protocol-Based CLIs feature

You can issue the **show l2vpn atom vc detail** command to show that both the local and remote routers support pseudowire status messages. The following example shows the line of output to look for:

```
Device# show l2vpn atom vc detail

.

.

.

status TLV support (local/remote): enabled/supported
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
Any Transport over MPLS	Any Transport over MPLS

Standards

Standard	Title
draft-ietf-pwe3-control-protocol-15.txt	Pseudowire Setup and Maintenance Using LDP
draft-ietf-pwe3-iana-allocation-08.txt	IANA Allocations for Pseudo Wire Edge to Edge Emulation (PWE3)
draft-martini-pwe3-pw-switching-03.txt	Pseudo Wire Switching

MIBs

MIB	MIBs Link
Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

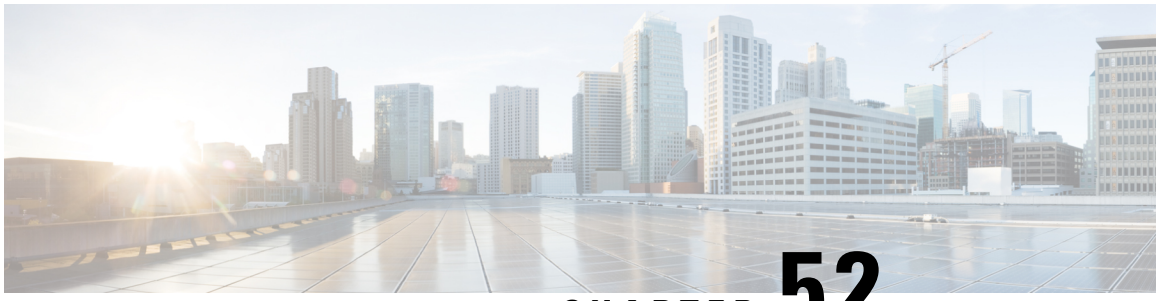
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Pseudowire Status Signaling

Table 114: Feature Information for MPLS Pseudowire Status Signaling

Feature Name	Releases	Feature Information
MPLS Pseudowire Status Signaling	Cisco IOS XE Release 2.3	The MPLS Pseudowire Status Signaling feature enables you to configure the router so it can send pseudowire status to a peer router, even when the attachment circuit is down. The following commands were introduced or modified: **debug mpls l2transport vc**, **show mpls l2transport vc**, **status** (pseudowire class).



CHAPTER 52

L2VPN VPLS Inter-AS Option B

The L2VPN VPLS Inter-AS Option B feature expands the existing features of VPLS autodiscovery to operate across multiple Border Gateway Protocol (BGP) autonomous systems. Using BGP-based autodiscovery as the underlying framework, the L2VPN VPLS Inter-AS Option B feature creates a dynamic multisegmented pseudowire (PW) configuration between neighboring Autonomous System Boundary Routers (ASBRs.)

Prerequisites for L2VPN VPLS Inter-AS Option B

The L2VPN VPLS Inter-AS Option B feature extends the functionality of the VPLS Autodiscovery: BGP Based feature. For example, as a result of L2VPN VPLS Inter-AS Option B feature, stateful switchover (SSO) and nonstop forwarding (NSF) are supported in a standard VPLS Autodiscovery configuration.

Before you configure the L2VPN VPLS Inter-AS Option B feature, enable the VPLS Autodiscovery: BGP Based feature.

For more information about the VPLS Autodiscovery: BGP Based feature, see the “VPLS Autodiscovery: BGP” module.

Restrictions for L2VPN VPLS Inter-AS Option B

Introduced in Cisco IOS Release 15.1(1)S, the L2VPN VPLS Inter-AS Option B feature is supported only on a Cisco 7600 series router that is equipped with a line card capable of running Virtual Private LAN Switching (VPLS).

Information About L2VPN VPLS Inter-AS Option B

VPLS Functionality and L2VPN VPLS Inter-AS Option B

VPLS is a multipoint Layer 2 VPN (L2VPN) that connects two or more customer devices using Ethernet over Multiprotocol Label Switching (EoMPLS) bridging techniques.

VPLS Inter-AS support exists in a number of variations or options (for example, Option A, B, C, and D). The L2VPN VPLS Inter-AS Option B feature supports Option B only and is in compliance with RFC 4364, BGP/MPLS IP Virtual Private Networks (VPNs) .

For more information about VPLS, see the "VPLS Overview" section in the Configuring Multiprotocol Label Switching on the Optical Services Modules document.

L2VPN VPLS Inter-AS Option B Description

The L2VPN VPLS Inter-AS Option B feature extends VPLS across multiple autonomous system boundaries by dynamically creating multisegment pseudowires across the ASBRs.

When a router with external BGP (eBGP) advertises routes to its BGP neighbors, the router uses the source IP address as the next hop of the advertised routes.

When a router with internal BGP (iBGP) advertises routes to its BGP neighbors, the router does not change the next hop designation of the route advertised. For the L2VPN VPLS Inter-AS Option B feature, enter the **neighbor next-hop-self** command at the ASBRs. This forces the pseudowires to be targeted to the ASBR and not targeted to the provider edge (PE) routers. The net result is that a pseudowire for the first autonomous system is stitched to a pseudowire for the second autonomous system by means of a third pseudowire between the ASBRs. This creates a multisegmented pseudowire. For more information about multisegmented pseudowires, see the "L2VPN Multisegment Pseudowires" module.

Note The L2VPN VPLS Inter-AS Option B feature supports Route Processors (RPs), SSO, and NSF.

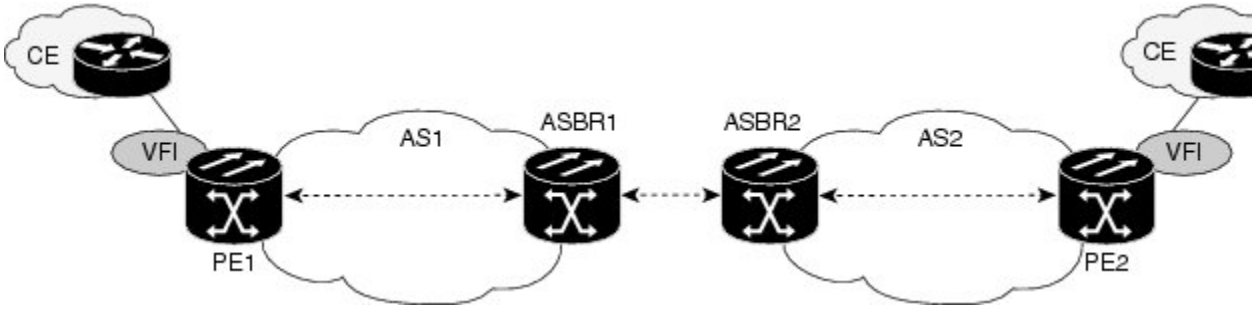
L2VPN VPLS Inter-AS Option B Sample Topology

The figure below illustrates a simplified L2VPN VPLS Inter-AS Option B topology. In this topology, AS1 and AS2 are the autonomous systems. ASBR1 and ASBR2 are ASBRs. A customer edge (CE) router is attached to both AS1 and AS2.

Each autonomous system consists of an ASBR and a PE router. PE1 belongs to a virtual forwarding instance (VFI) in AS1. PE2 belongs to a VFI in AS2. PE1 and PE2 are terminating PEs (TPEs).

Multisegmented pseudowires are created to establish dual connections between the TPE in the local ASBR to the TPE in the neighboring ASBR. The first segment establishes a path between the TPE in AS1 to ASBR1. The next segment establishes a path between the ASBR1 and ASBR2, and the final segment establishes a path between ASBR2 to the TPE in AS2.

Figure 70: Sample L2VPN VPLS Inter-AS Option B Topology



Active and Passive PEs in an L2VPN VPLS Inter-AS Option B Configuration

A TPE terminates a multisegment pseudowire. By default, the TPEs on both ends of a multisegmented pseudowire are in active mode. The L2VPN VPLS Inter-AS Option B feature requires that one of the TPEs be in passive mode. The system determines which PE is the passive TPE based on a comparison of the Target Attachment Individual Identifier (TAII) received from BGP and the Source Attachment Individual Identifier (SAII) of the local router. The TPE with the numerically higher identifier assumes the active role.

When you are configuring the PEs for the L2VPN VPLS Inter-AS Option B feature, use the **terminating-pe tie-breaker**command to negotiate the mode of the TPE. Then use the **mpls ldp discovery targeted-hello accept**command to ensure that a passive TPE can accept Label Distribution Protocol (LDP) sessions from the LDP peers.

For more information about configuring the PEs, see the Enabling L2VPN VPLS Inter-AS Option B on the Provider Edge (PE) Router, on page 1070.

Benefits of L2VPN VPLS Inter-AS Option B

Private IP Addresses

While a large number of pseudowires are required, IPv4 reachability is maintained within the ASBR and, therefore, IP addresses are private.

One Targeted LDP Session

With the L2VPN VPLS Inter-AS Option B feature, only one targeted Label Distribution Protocol (LDP) session is created between the autonomous systems. Since only one targeted LDP session between autonomous systems is created, service providers can apply tighter security policies for control plane traffic going across the autonomous system.

How to Configure L2VPN VPLS Inter-AS Option B

Modifying the VPLS Autodiscovery Settings for Use with L2VPN VPLS Inter-AS Option B

Note Before you configure the L2VPN VPLS Inter-AS Option B feature, you must enable the VPLS Autodiscovery: BGP Based feature. Make sure you have enabled the VPLS Autodiscovery: BGP Based feature before proceeding with this task.

For the L2VPN VPLS Inter-AS Option B feature to function properly, you must configure the VPLS ID value and the route-target value for each PE router in the virtual forwarding instance (VFI). To modify these values, complete the following steps at each PE router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *vfi-name* **autodiscovery**
4. **vpn id** *vpn-id*
5. **vpls-id** {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*}
6. **route-target** [**import** | **export** | **both**] {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*}
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *vfi-name* **autodiscovery** **Example:** `Router(config)# l2 vfi vpls1 autodiscovery`	Enables the VPLS Autodiscovery: BGP Based feature on the PE router and enters L2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:**	Configures a VPN ID for the VPLS domain. • Enter a VPN ID value.

	Command or Action	Purpose
	`Router(config-vfi)# vpn id 10`	
Step 5	**vpls-id** {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*} **Example:** `Router(config-vfi)# vpls-id 5:300`	Specifies the VPLS ID. • The VPLS Autodiscovery: BGP Based feature automatically generates a VPLS ID using the BGP autonomous system number and the configured VFI VPN ID. Use this command to change the automatically generated VPLS ID for the PE in the VFI. • There are two formats for configuring the VPLS ID argument. It can be configured in the *autonomous-system-number* **:** *network number* (*ASN* **:** *nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address* **:** *nn)*.
Step 6	**route-target** [**import** | **export** | **both**] {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*} **Example:** `Router(config-vfi)# route-target 600:2222`	Specifies the route target (RT). • The VPLS Autodiscovery feature automatically generates a route target using the lower 6 bytes of the RD and VPN ID. Use this command to change the automatically generated route target for the PE in the VFI. • There are two formats for configuring the route target argument. It can be configured in the *autonomous-system-number* **:** *network number* (*ASN* **:** *nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address* **:** *nn)*.
Step 7	**exit** **Example:** `Router(config-vfi)# exit`	Exits L2 VFI configuration mode. • Commands take effect after the router exits L2 VFI configuration mode.

What to Do Next

Repeat the steps at each PE in the autonomous system.

Modifying the VPLS Autodiscovery Settings for Use with L2VPN VPLS Inter-AS Option B using the commands associated with the L2VPN Protocol-Based CLIs feature

Note Before you configure the L2VPN VPLS Inter-AS Option B feature, you must enable the VPLS Autodiscovery: BGP Based feature. Make sure you have enabled the VPLS Autodiscovery: BGP Based feature before proceeding with this task.

For the L2VPN VPLS Inter-AS Option B feature to function properly, you must configure the VPLS ID value and the route-target value for each PE router in the virtual forwarding instance (VFI). To modify these values, complete the following steps at each PE router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *vfi-name*
4. **vpn id** *vpn-id*
5. **autodiscovery bgp signaling ldp**
6. **vpls-id** {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*}
7. **route-target** [**import** | **export** | **both**] {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*}
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *vfi-name* **Example:** `Device(config)# l2vpn vfi context vpls1`	Establishes an L2VPN VFI context and enters L2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain. • Enter a VPN ID value.

	Command or Action	Purpose
Step 5	**autodiscovery bgp signaling ldp** **Example:** `Device(config-vfi)# autodiscovery bgp signaling ldp`	Enables the VPLS Autodiscovery: BGP Based feature on the PE router.
Step 6	**vpls-id** {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*} **Example:** `Device(config-vfi)# vpls-id 5:300`	Specifies the VPLS ID. • The VPLS Autodiscovery: BGP Based feature automatically generates a VPLS ID using the BGP autonomous system number and the configured VFI VPN ID. Use this command to change the automatically generated VPLS ID for the PE in the VFI. • There are two formats for configuring the VPLS ID argument. It can be configured in the *autonomous-system-number* **:** *network number* (*ASN* **:** *nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address* **:** *nn)*.
Step 7	**route-target** [**import** | **export** | **both**] {*autonomous-system-number* **:** *nn* | *ip-address* **:** *nn*} **Example:** `Device(config-vfi)# route-target 600:2222`	Specifies the route target (RT). • The VPLS Autodiscovery feature automatically generates a route target using the lower 6 bytes of the RD and VPN ID. Use this command to change the automatically generated route target for the PE in the VFI. • There are two formats for configuring the route target argument. It can be configured in the *autonomous-system-number* **:** *network number* (*ASN* **:** *nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address* **:** *nn)*.
Step 8	**exit** **Example:** `Device(config-vfi)# exit`	Exits L2 VFI configuration mode. • Commands take effect after the router exits L2 VFI configuration mode.

What to Do Next

Repeat the steps at each PE in the autonomous system.

Enabling L2VPN VPLS Inter-AS Option B on the ASBR

To enable the L2VPN VPLS Inter-AS Option B feature on the ASBR, complete the following steps on *each* ASBR in the autonomous system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**
5. **address-family l2vpn vpls**
6. **no bgp default route-target filter**
7. **exit**
8. **exit**
9. **mpls ldp discovery targeted-hello accept**
10. Complete Steps 11 through 13, only if you are changing the range of VC IDs reserved for switching pseudowires. Otherwise, advance to Step 14.
11. **l2 pseudowire routing**
12. **switching-point vcid** *minimum-vcid-value maximum-vcid-value*
13. **exit**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 1`	Configures the BGP routing process and enters router configuration mode. • Enter the number of the autonomous system.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Router(config-router)# neighbor 10.10.0.1 next-hop-self`	Configures the ASBR as the next hop for a BGP-speaking neighbor or peer group. • Enter the IP address or the peer group name. **Note** Use this command to identify each PE in the autonomous system.

	Command or Action	Purpose
Step 5	**address-family l2vpn vpls** **Example:** `Router(config-router)# address-family l2vpn vpls`	Configures a routing session using L2VPN endpoint provisioning address information and enters address family configuration mode.
Step 6	**no bgp default route-target filter** **Example:** `Router(config-router-af)# no bgp default route-target filter`	Enables pseudowire switching at the ASBR.
Step 7	**exit** **Example:** `Router(config-router-af) exit`	Exits address family configuration mode.
Step 8	**exit** **Example:** `Router(config-router) exit`	Exits router configuration mode.
Step 9	**mpls ldp discovery targeted-hello accept** **Example:** `Router(config)# mpls ldp discovery targeted-hello accept`	Configures the routers from which LDP sessions will be accepted. • With the **targeted-hello accept** keywords, LDP sessions from *any* router will be accepted. • For the other keyword choices available for this command, see the *Cisco IOS Multiprotocol Label Switching Command Reference*.
Step 10	Complete Steps 11 through 13, only if you are changing the range of VC IDs reserved for switching pseudowires. Otherwise, advance to Step 14.	
Step 11	**l2 pseudowire routing** **Example:** `Router(config))# l2 pseudowire routing`	(Optional) Enters Layer 2 pseudowire routing configuration mode.
Step 12	**switching-point vcid** *minimum-vcid-value maximum-vcid-value* **Example:** `Router(config-l2_pw_rtg)# switching-point vcid 200 3500`	(Optional) Configures a switching point and specifies a virtual circuit (VC) ID range. **Note** With the L2VPN VPLS Inter-AS Option B feature, VC IDs in the VC ID range of 1001 to 2147483647 are reserved for switching pseudowires. This command allows you to change this range if, for example, an existing xconnect VC is using one of the reserved VC IDs.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Router(config-l2_pw_rtg)#) exit`	Exits Layer 2 pseudowire routing configuration mode.
Step 14	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

What to Do Next

Repeat the steps at each ASBR in the autonomous system.

Enabling L2VPN VPLS Inter-AS Option B on the ASBR using the commands associated with the L2VPN Protocol-Based CLIs feature

To enable the layer 2 virtual private network virtual private LAN services (L2VPN VPLS) Inter-AS Option B feature on the autonomous system boundary router (ASBR), perform this task on each ASBR in the autonomous system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**
5. **address-family l2vpn vpls**
6. **no bgp default route-target filter**
7. **exit**
8. **exit**
9. **mpls ldp discovery targeted-hello accept**
10. Complete Steps 11 through 13, only if you are changing the range of VC IDs reserved for switching pseudowires. Otherwise, advance to Step 14.
11. **l2vpn**
12. **pseudowire routing**
13. **switching-point vcid** *minimum-vcid-value maximum-vcid-value*
14. **exit**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 1`	Configures the BGP routing process and enters router configuration mode. • Enter the number of the autonomous system.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Device(config-router)# neighbor 10.10.0.1 next-hop-self`	Configures the ASBR as the next hop for a BGP-speaking neighbor or peer group. • Enter the IP address or the peer group name. **Note** Use this command to identify each PE in the autonomous system.
Step 5	**address-family l2vpn vpls** **Example:** `Device(config-router)# address-family l2vpn vpls`	Configures a routing session using L2VPN endpoint provisioning address information and enters address family configuration mode.
Step 6	**no bgp default route-target filter** **Example:** `Device(config-router-af)# no bgp default route-target filter`	Enables pseudowire switching at the ASBR.
Step 7	**exit** **Example:** `Device(config-router-af) exit`	Exits address family configuration mode.
Step 8	**exit** **Example:** `Device(config-router) exit`	Exits router configuration mode.
Step 9	**mpls ldp discovery targeted-hello accept** **Example:** `Device(config)# mpls ldp discovery targeted-hello accept`	Configures the routers from which LDP sessions will be accepted. • With the **targeted-hello accept** keywords, LDP sessions from *any* router will be accepted.

	Command or Action	Purpose
		• For the other keyword choices available for this command, see the *Cisco IOS Multiprotocol Label Switching Command Reference*.
Step 10	Complete Steps 11 through 13, only if you are changing the range of VC IDs reserved for switching pseudowires. Otherwise, advance to Step 14.	
Step 11	**l2vpn** **Example:** `Device(config)# l2vpn`	(Optional) Enters Layer 2 VPN configuration mode.
Step 12	**pseudowire routing** **Example:** `Device(l2vpn-config)# pseudowire routing`	(Optional) Enters Layer 2 pseudowire routing configuration mode.
Step 13	**switching-point vcid** *minimum-vcid-value maximum-vcid-value* **Example:** `Device(config-l2_pw_rtg)# switching-point vcid 200 3500`	(Optional) Configures a switching point and specifies a virtual circuit (VC) ID range. **Note** With the L2VPN VPLS Inter-AS Option B feature, VC IDs in the VC ID range of 1001 to 2147483647 are reserved for switching pseudowires. This command allows you to change this range if, for example, an existing xconnect VC is using one of the reserved VC IDs.
Step 14	**exit** **Example:** `Device(config-l2_pw_rtg)# exit`	Exits Layer 2 pseudowire routing configuration mode.
Step 15	**end** **Example:** `Device(config)# end`	Exits global configuration mode.

What to Do Next

Repeat the steps at each ASBR in the autonomous system.

Enabling L2VPN VPLS Inter-AS Option B on the Provider Edge (PE) Router

To enable the L2VPN VPLS Inter-AS Option B on the PE router, complete the following steps on each PE in the autonomous system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 pseudowire routing**
4. **terminating-pe tie-breaker**
5. exit
6. **mpls ldp discovery targeted-hello accept**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2 pseudowire routing** **Example:** `Router(config))# l2 pseudowire routing`	Enters Layer 2 pseudowire routing configuration mode.
Step 4	**terminating-pe tie-breaker** **Example:** `Router(config-l2_pw_rtg)# terminating-pe tie-breaker`	Negotiates the behavior mode (either active or passive) for a terminating provider edge (TPE) route.
Step 5	exit **Example:** `Router(config-l2_pw_rtg)# exit`	Returns to global configuration mode.
Step 6	**mpls ldp discovery targeted-hello accept** **Example:** `Router(config)# mpls ldp discovery targeted-hello accept`	Configures the routers from which LDP sessions will be accepted. • With the **targeted-hello accept** keywords, LDP sessions from *any* router will be accepted. • For the other keyword choices available for this command, see the *Cisco IOS Multiprotocol Label Switching Command Reference* .

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

What to Do Next

Repeat the steps at each PE in the autonomous system.

Enabling L2VPN VPLS Inter-AS Option B on the Provider Edge (PE) Router using the commands associated with the L2VPN Protocol-Based CLIs feature

To enable the L2VPN VPLS Inter-AS Option B on the PE router, perform this task on each PE in the autonomous system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn**
4. **pseudowire routing**
5. **terminating-pe tie-breaker**
6. **end**
7. **mpls ldp discovery targeted-hello accept**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn** **Example:** `Device(config)# l2vpn`	(Optional) Enters Layer 2 VPN configuration mode.

	Command or Action	Purpose
Step 4	**pseudowire routing** **Example:** `Device(l2vpn-config)# pseudowire routing`	(Optional) Enters Layer 2 pseudowire routing configuration mode.
Step 5	**terminating-pe tie-breaker** **Example:** `Device(config-l2_pw_rtg)# terminating-pe tie-breaker`	Negotiates the behavior mode (either active or passive) for a terminating provider edge (TPE) route.
Step 6	**end** **Example:** `Device(config-l2_pw_rtg)# exit`	Returns to global configuration mode.
Step 7	**mpls ldp discovery targeted-hello accept** **Example:** `Device(config)# mpls ldp discovery targeted-hello accept`	Configures the routers from which LDP sessions will be accepted. • With the **targeted-hello accept** keywords, LDP sessions from *any* router will be accepted. • For the other keyword choices available for this command, see the *Cisco IOS Multiprotocol Label Switching Command Reference*.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode.

What to Do Next

Repeat the steps at each PE in the autonomous system.

Verifying the L2VPN VPLS Inter-AS Option B Configuration

To verify the L2VPN VPLS Inter-AS Option B configuration, use one or more of the following commands at any router.

SUMMARY STEPS

1. **enable**
2. **show xconnect rib detail**
3. **show mpls l2transport vc** [**detail**] [**pwid** *pw-identifier*] [**vpls-id** *vpls-identifier*] [**stitch** *endpoint endpoint*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show xconnect rib detail** **Example:** `Router# show xconnect rib detail`	(Optional) Displays the information about the pseudowire Routing Information Base (RIB).
Step 3	**show mpls l2transport vc** [**detail**] [**pwid** *pw-identifier*] [**vpls-id** *vpls-identifier*] [**stitch** *endpoint endpoint*] **Example:** `Router# show mpls l2transport vc`	(Optional) Displays the information about Multiprotocol Label Switching (MPLS) Any Transport over ATM (AToM) VCs and static pseudowires that have been enabled to route Layer 2 packets on a router. • Use the optional keywords and arguments, as applicable.
Step 4	**end** **Example:** `Router# end`	Exits privileged EXEC mode.

Verifying the L2VPN VPLS Inter-AS Option B Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

To verify the L2VPN VPLS Inter-AS Option B configuration, use one or more of the following commands on any router.

SUMMARY STEPS

1. **enable**
2. **show l2vpn rib detail**
3. **show l2vpn atom vc** [**pwid** *pw-identifier*] [**vpls-id** *vpls-identifier*] [**stitch** *endpoint endpoint*][**detail**]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show l2vpn rib detail** **Example:** `Device# show l2vpn rib detail`	(Optional) Displays the information about the pseudowire Routing Information Base (RIB).
Step 3	**show l2vpn atom vc** [**pwid** *pw-identifier*] [**vpls-id** *vpls-identifier*] [**stitch** *endpoint endpoint*][**detail**] **Example:** `Device# show l2vpn atom vc`	(Optional) Displays the information about Multiprotocol Label Switching (MPLS) Any Transport over ATM (AToM) VCs and static pseudowires that have been enabled to route Layer 2 packets on a router. • Use the optional keywords and arguments, as applicable.
Step 4	**end** **Example:** `Device# end`	Exits privileged EXEC mode.

Configuration Examples for L2VPN VPLS Inter-AS Option B

Example Modifying the VPLS Autodiscovery Settings for Use with L2VPN VPLS Inter-AS Option B

In the following example, the VPLS Autodiscovery: BGP Based feature is modified for use with the L2VPN VPLS Inter-AS Option B feature:

```
Router> enable

Router# configure terminal

Router(config)# l2 vfi vpls1 autodiscovery

Router(config-vfi)# vpn id 10

Router(config-vfi)# vpls-id 5:300

Router(config-vfi)# route-target 600:2222

Router(config-vfi)# exit
```

Example: Modifying the VPLS Autodiscovery Settings for Use with L2VPN VPLS Inter-AS Option B using the commands associated with the L2VPN Protocol-Based CLIs feature

In the following example, the VPLS Autodiscovery: BGP Based feature is modified for use with the L2VPN VPLS Inter-AS Option B feature:

```
Device# enable
Device# configure terminal
Device(config)# l2vpn vfi context vpls1
Device(config-vfi)# vpn id id
Device(config-vfi)# autodiscovery bgp signaling ldp
Device(config-vfi)# vpls-id 5:300
Device(config-vfi)# route-target 600:2222
Device(config-vfi)# exit
```

Example Enabling L2VPN VPLS Inter-AS Option B on the ASBR

In the following example, the L2VPN VPLS Inter-AS Option B feature has been configured on one ASBR:

```
Router> enable

Router# configure terminal

Router(config)# router bgp 1

Router(config-router)# neighbor 10.10.0.1 next-hop-self

Router(config-router)# address-family l2vpn vpls

Router(config-router-af)# no bgp default route-target filter

Router(config-router-af)# exit

Router(config-router)# exit

Router(config)# mpls ldp discovery targeted-hello accept

Router(config)# end
```

Example Enabling L2VPN VPLS Inter-AS Option B on the PE Router

In the following example, the L2VPN VPLS Inter-AS Option B feature is configured on a PE router. The PE is also a TPE.

```
Router> enable
```

```
Router# configure terminal

Router(config))# l2 pseudowire routing

Router(config-l2_pw_rtg)# terminating-pe tie-breaker

Router(config-l2_pw_rtg)# exit

Router(config)# mpls ldp discovery targeted-hello accept

Router(config)# end
```

Example Enabling L2VPN VPLS Inter-AS Option B on the PE Device using the commands associated with the L2VPN Protocol-Based CLIs feature

In the following example, the L2VPN VPLS Inter-AS Option B feature is configured on a provider edge (PE) router. The PE is also a terminating provider edge (TPE).

```
Device> enable
Device# configure terminal
Device(config)# l2vpn
Device(l2vpn-config)#  pseudowire routing
Device(config-l2_pw_rtg)# terminating-pe tie-breaker
Device(config-l2_pw_rtg)# exit
Device(config)# mpls ldp discovery targeted-hello accept
Device(config)# end
```

Example Verifying the L2VPN VPLS Inter-AS Option B Configuration

The output of the **show xconnect rib detail**command can be used to verify the L2VPN VPLS Inter-AS Option B configuration.

The following is sample output from the **show xconnect rib detail** command when used in an ASBR configuration. On an ASBR, the **show xconnect rib detail** command displays the Layer 2 VPN BGP Network Layer Reachability Information (NLRI) received from the BGP peers. The display also shows the signaling messages received from the targeted LDP sessions for a given TAII.

```
Router# show xconnect rib detail
Local Router ID: 10.1.1.3
VPLS-ID: 1:1, Target ID: 10.1.1.1
  Next-Hop: 10.1.1.1
  Hello-Source: 10.1.1.3
  Route-Target: 2:2
  Incoming RD: 10.0.0.0:1
  Forwarder:
  Origin: BGP
  Provisioned: Yes
  SAII: 10.0.0.1, LDP Peer Id: 10.255.255.255, VC Id: 1001 ***
  SAII: 10.1.0.1, LDP Peer Id: 10.255.255.255, VC Id: 1002 ***
```

After the passive TPE router receives the BGP information (and before the passive TPE router receives the LDP label), the peer information will be displayed in the output of the **show xconnect rib** command. The peer information will not be displayed in the **show mpls l2transport vc**command because the VFI AToM xconnect has not yet been provisioned.

Therefore, for passive TPEs, the entry "Passive : Yes" is added to the output of the **show xconnect rib detail** command. In addition, the entry "Provisioned: Yes" is displayed after the neighbor xconnect is successfully created (without any retry attempts).

In the sample output, the two lines beginning with "SAII" show that this ASBR is stitching two provider PE routers (10.0.0.1 and 10.1.0.1) to the TAII 10.1.1.1.

Example Verifying the L2VPN VPLS Inter-AS Option B Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

The output of the **show l2vpn rib detail** command can be used to verify the L2VPN VPLS Inter-AS Option B configuration.

The following is sample output from the **show l2vpn rib detail** command when used in an autonomous system boundary router (ASBR) configuration. On an ASBR, the **show l2vpn rib detail** command displays the Layer 2 VPN BGP Network Layer Reachability Information (NLRI) received from the BGP peers. The display also shows the signaling messages received from the targeted label distribution protocol (LDP) sessions for a given TAII.

```
Device# show l2vpn rib detail
Local Router ID: 10.1.1.3
VPLS-ID: 1:1, Target ID: 10.1.1.1
  Next-Hop: 10.1.1.1
  Hello-Source: 10.1.1.3
  Route-Target: 2:2
  Incoming RD: 10.0.0.0:1
  Forwarder:
  Origin: BGP
  Provisioned: Yes
  SAII: 10.0.0.1, LDP Peer Id: 10.255.255.255, VC Id: 1001 ***
  SAII: 10.1.0.1, LDP Peer Id: 10.255.255.255, VC Id: 1002 ***
```

After the passive terminating provider edge (TPE) router receives the BGP information (and before the passive TPE router receives the LDP label), the peer information will be displayed in the output of the **show l2vpn rib** command. The peer information will not be displayed in the **show l2vpn atom vc** command because the VFI AToM xconnect has not yet been provisioned.

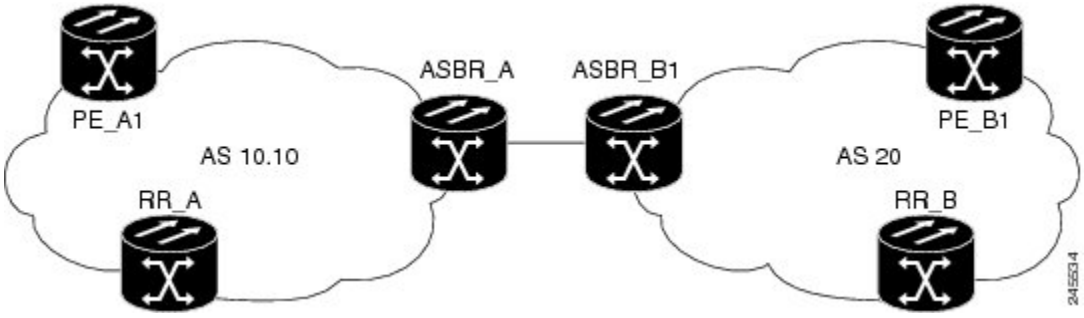
Therefore, for passive TPEs, the entry "Passive : Yes" is added to the output of the **show l2vpn rib detail** command. In addition, the entry "Provisioned: Yes" is displayed after the neighbor xconnect is successfully created (without any retry attempts).

In the sample output, the two lines beginning with "SAII" show that this ASBR is stitching two provider PE routers (10.0.0.1 and 10.1.0.1) to the TAII 10.1.1.1.

Example Sample L2VPN VPLS Inter-AS Option B Configuration

The following is a sample L2VPN VPLS Inter-AS Option B configuration based on the topology shown in the figure below.

Figure 71: L2VPN VPLS Inter-AS Option B Topology Used for Configuration Example



The topology shown in the figure above consists of two PE routers connected across an autonomous system boundary using two ASBRs. Routes are shared within each autonomous system using BGP route reflectors (RRs). (The RRs are included only for the purpose of showing a complete configuration. RRs are not a requirement for the L2VPN Inter-AS Option B configuration.)

The specific configurations for each of the elements in this topology are shown below. The text in bold indicates the additions needed to the standard VPLS Autodiscovery: BGP Based configuration.

PE_A1 Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
l2 router-id 10.1.1.1
!
l2 pseudowire routing
 terminating-pe tie-breaker
!
l2 vfi vfiA autodiscovery
 vpn id 111
 vpls-id 111:111
 rd 111:111
 route-target 111:111
 no auto-route-target
!
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
!
interface GigabitEthernet2/0/9
 description AS-10.10-Backbone-LAN
 ip address 10.100.100.1 255.255.255.0
 mpls ip
!
router ospf 10
 network 10.1.1.1 0.0.0.0 area 0
 network 10.100.100.1 0.0.0.0 area 0
!
router bgp 10.10
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.3.3.3 remote-as 10.10
 neighbor 10.3.3.3 description RR-AS-10.10
 neighbor 10.3.3.3 update-source Loopback0
```

```
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 send-community extended
 exit-address-family
!
mpls ldp router-id Loopback0
!
```

ASBR_A Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
!
interface Loopback0
 ip address 10.4.4.4 255.255.255.255
!
interface GigabitEthernet1/10
 description AS-10.10-backbone-Lan
 ip address 10.100.100.4 255.255.255.0
 mpls ip
!
interface GigabitEthernet2/0/1
 description B2B-AS-20-ASBR-B1
 ip address 10.12.1.4 255.255.255.0
 mpls ip
!
router ospf 10
 passive-interface GigabitEthernet1/12
 passive-interface GigabitEthernet2/0/1
 passive-interface GigabitEthernet2/0/2
 network 10.4.4.4 0.0.0.0 area 0
 network 10.100.100.4 0.0.0.0 area 0
 network 10.12.0.0 0.0.255.255 area 0
!
router bgp 10.10
 bgp router-id 10.4.4.4
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default route-target filter
 no bgp default ipv4-unicast
 timers bgp 10 30
 neighbor AS20 peer-group
 neighbor AS20 remote-as 20
 neighbor 10.3.3.3 remote-as 10.10
 neighbor 10.3.3.3 update-source Loopback0
 neighbor 10.12.1.6 peer-group AS20
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor AS20 send-community extended
  neighbor AS20 next-hop-self
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 send-community extended
  neighbor 10.3.3.3 next-hop-self
```

```
  neighbor 12.12.1.6 activate
 exit-address-family
!
ip route 10.6.6.6 255.255.255.255 10.12.1.6
ip route 10.9.9.9 255.255.255.255 10.12.3.9
!
mpls ldp router-id Loopback0
!
```

RR_A Router

```
interface Loopback0
 ip address 10.3.3.3 255.255.255.255
!
interface Ethernet2/0
 ip address 10.100.100.3 255.255.255.0
 duplex half
!
router ospf 10
 network 10.3.3.3 0.0.0.0 area 0
 network 10.100.100.3 0.0.0.0 area 0
!
router bgp 10.10
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor rr-client peer-group
 neighbor rr-client remote-as 10.10
 neighbor rr-client update-source Loopback0
 neighbor 10.1.1.1 peer-group rr-client
 neighbor 10.4.4.4 peer-group rr-client
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor rr-client send-community extended
  neighbor rr-client route-reflector-client
  neighbor 10.1.1.1 activate
  neighbor 10.4.4.4 activate
 exit-address-family
!
```

PE_B1 Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
l2 router-id 10.5.5.5
l2 pseudowire routing
 terminating-pe tie-breaker
l2 vfi vfiA autodiscovery
 vpn id 111
 vpls-id 111:111
 rd 111:111
 route-target 111:111
 no auto-route-target
!
interface Loopback0
 ip address 10.5.5.5 255.255.255.255
```

```
!
interface GigabitEthernet2/0/7
 description AS20-Backbone-LAN
 ip address 10.100.100.5 255.255.255.0
 mpls ip
!
router ospf 20
 network 10.5.5.5 0.0.0.0 area 0
 network 10.100.100.5 0.0.0.0 area 0
!
router bgp 20
 bgp router-id 10.5.5.5
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.8.8.8 remote-as 20
 neighbor 10.8.8.8 update-source Loopback0
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.8.8.8 activate
  neighbor 10.8.8.8 send-community extended
 exit-address-family
!
mpls ldp router-id Loopback0
!
```

ASBR_B1 Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
l2 router-id 10.6.6.6
l2 pseudowire routing
 terminating-pe tie-breaker
!
interface Loopback0
 ip address 10.6.6.6 255.255.255.255
!
interface Ethernet1/3
 description B2B-AS-10.10-ASBR-A
 ip address 10.12.1.6 255.255.255.0
 duplex half
 mpls ip
!
interface Ethernet2/1
 description AS-20-backbone-Lan
 ip address 10.100.100.6 255.255.255.0
 duplex half
 mpls ip
!
router ospf 20
 passive-interface Ethernet1/3
 network 10.12.1.6 0.0.0.0 area 0
 network 10.6.6.6 0.0.0.0 area 0
 network 10.100.100.6 0.0.0.0 area 0
!
router bgp 20
 bgp router-id 10.6.6.6
 bgp asnotation dot
```

```
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 timers bgp 10 30
 neighbor 10.12.1.4 remote-as 10.10
 neighbor 10.12.1.4 ebgp-multihop 255
 neighbor 10.8.8.8 remote-as 20
 neighbor 10.8.8.8 update-source Loopback0
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  no bgp default route-target filter
  neighbor 10.12.1.4 activate
  neighbor 10.12.1.4 send-community extended
  neighbor 10.12.1.4 next-hop-self
  neighbor 10.8.8.8 activate
  neighbor 10.8.8.8 send-community extended
  neighbor 10.8.8.8 next-hop-self
 exit-address-family
!
```

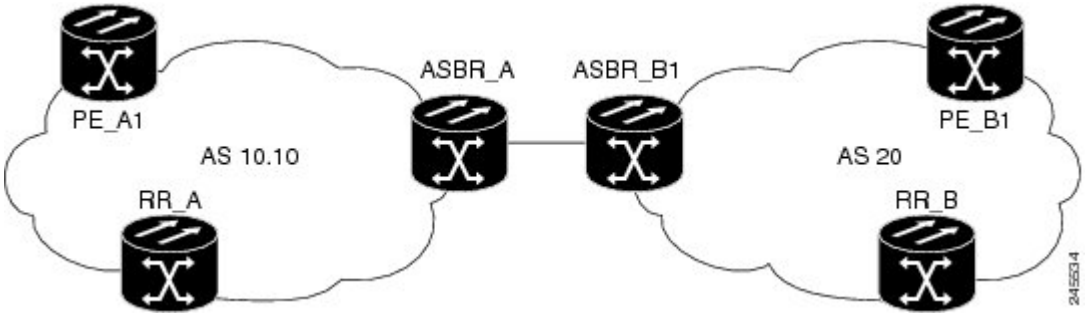
RR_B Router

```
interface Loopback0
 ip address 10.8.8.8 255.255.255.255
!
interface Ethernet2/1
 ip address 10.100.100.8 255.255.255.0
 duplex half
!
router ospf 20
 network 10.8.8.8 0.0.0.0 area 0
 network 10.100.100.8 0.0.0.0 area 0
!
router bgp 20
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor rrc peer-group
 neighbor rrc remote-as 20
 neighbor rrc update-source Loopback0
 neighbor 10.5.5.5 peer-group rrc
 neighbor 10.6.6.6 peer-group rrc
 neighbor 10.9.9.9 peer-group rrc
 neighbor 10.9.9.9 shutdown
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor rrc send-community extended
  neighbor rrc route-reflector-client
  neighbor 10.5.5.5 activate
  neighbor 10.6.6.6 activate
  neighbor 10.9.9.9 activate
 exit-address-family
!
```

Example Sample L2VPN VPLS Inter-AS Option B Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

The example below is a sample L2VPN VPLS Inter-AS Option B configuration based on the topology shown in the following figure.

Figure 72: L2VPN VPLS Inter-AS Option B Topology Used for Configuration Example



The topology shown in the figure above consists of two provider edge (PE) routers connected across an autonomous system boundary using two ASBRs. Routes are shared within each autonomous system using BGP route reflectors (RRs). (The RRs are included only for the purpose of showing a complete configuration. RRs are not a requirement for the L2VPN Inter-AS Option B configuration.)

The specific configurations for each of the elements in this topology are shown below. The commands highlighted in bold indicate the additions needed to the standard VPLS Autodiscovery: BGP Based configuration.

PE_A1 Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
l2vpn
 router-id 10.1.1.1
 pseudowire routing
   terminating-pe tie-breaker
!
l2vpn vfi context vfiA
 vpn id 111
 autodiscovery bgp signaling ldp
 vpls-id 111:111
 rd 111:111
 route-target 111:111
 no auto-route-target
!
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
!
interface GigabitEthernet2/0/9
 description AS-10.10-Backbone-LAN
 ip address 10.100.100.1 255.255.255.0
 mpls ip
!
router ospf 10
```

```
 network 10.1.1.1 0.0.0.0 area 0
 network 10.100.100.1 0.0.0.0 area 0
!
router bgp 10.10
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.3.3.3 remote-as 10.10
 neighbor 10.3.3.3 description RR-AS-10.10
 neighbor 10.3.3.3 update-source Loopback0
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 send-community extended
 exit-address-family
!
mpls ldp router-id Loopback0
!
```

ASBR_A Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
!
interface Loopback0
 ip address 10.4.4.4 255.255.255.255
!
interface GigabitEthernet1/10
 description AS-10.10-backbone-Lan
 ip address 10.100.100.4 255.255.255.0
 mpls ip
!
interface GigabitEthernet2/0/1
 description B2B-AS-20-ASBR-B1
 ip address 10.12.1.4 255.255.255.0
 mpls ip
!
router ospf 10
 passive-interface GigabitEthernet1/12
 passive-interface GigabitEthernet2/0/1
 passive-interface GigabitEthernet2/0/2
 network 10.4.4.4 0.0.0.0 area 0
 network 10.100.100.4 0.0.0.0 area 0
 network 10.12.0.0 0.0.255.255 area 0
!
router bgp 10.10
 bgp router-id 10.4.4.4
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default route-target filter
 no bgp default ipv4-unicast
 timers bgp 10 30
 neighbor AS20 peer-group
 neighbor AS20 remote-as 20
 neighbor 10.3.3.3 remote-as 10.10
 neighbor 10.3.3.3 update-source Loopback0
 neighbor 10.12.1.6 peer-group AS20
 !
```

```
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor AS20 send-community extended
  neighbor AS20 next-hop-self
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 send-community extended
  neighbor 10.3.3.3 next-hop-self
  neighbor 12.12.1.6 activate
 exit-address-family
!
ip route 10.6.6.6 255.255.255.255 10.12.1.6
ip route 10.9.9.9 255.255.255.255 10.12.3.9
!
mpls ldp router-id Loopback0
!
```

RR_A Router

```
interface Loopback0
 ip address 10.3.3.3 255.255.255.255
!
interface Ethernet2/0
 ip address 10.100.100.3 255.255.255.0
 duplex half
!
router ospf 10
 network 10.3.3.3 0.0.0.0 area 0
 network 10.100.100.3 0.0.0.0 area 0
!
router bgp 10.10
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor rr-client peer-group
 neighbor rr-client remote-as 10.10
 neighbor rr-client update-source Loopback0
 neighbor 10.1.1.1 peer-group rr-client
 neighbor 10.4.4.4 peer-group rr-client
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor rr-client send-community extended
  neighbor rr-client route-reflector-client
  neighbor 10.1.1.1 activate
  neighbor 10.4.4.4 activate
 exit-address-family
!
```

PE_B1 Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
l2vpn
 router-id 10.5.5.5
```

```
 pseudowire routing
  terminating-pe tie-breaker
l2vpn vfi context vfiA
 vpn id 111
 autodiscovery bgp signaling ldp
 vpls-id 111:111
 rd 111:111
 route-target 111:111
 no auto-route-target
!
interface Loopback0
 ip address 10.5.5.5 255.255.255.255
!
interface GigabitEthernet2/0/7
 description AS20-Backbone-LAN
 ip address 10.100.100.5 255.255.255.0
 mpls ip
!
router ospf 20
 network 10.5.5.5 0.0.0.0 area 0
 network 10.100.100.5 0.0.0.0 area 0
!
router bgp 20
 bgp router-id 10.5.5.5
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.8.8.8 remote-as 20
 neighbor 10.8.8.8 update-source Loopback0
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 10.8.8.8 activate
  neighbor 10.8.8.8 send-community extended
 exit-address-family
!
mpls ldp router-id Loopback0
!
```

ASBR_B1 Router

```
mpls ldp discovery targeted-hello accept
mpls label protocol ldp
!
l2vpn
 router-id 10.6.6.6
 pseudowire routing
  terminating-pe tie-breaker
!
interface Loopback0
 ip address 10.6.6.6 255.255.255.255
!
interface Ethernet1/3
 description B2B-AS-10.10-ASBR-A
 ip address 10.12.1.6 255.255.255.0
 duplex half
 mpls ip
!
interface Ethernet2/1
 description AS-20-backbone-Lan
```

```
 ip address 10.100.100.6 255.255.255.0
 duplex half
 mpls ip
!
router ospf 20
 passive-interface Ethernet1/3
 network 10.12.1.6 0.0.0.0 area 0
 network 10.6.6.6 0.0.0.0 area 0
 network 10.100.100.6 0.0.0.0 area 0
!
router bgp 20
 bgp router-id 10.6.6.6
 bgp asnotation dot
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 timers bgp 10 30
 neighbor 10.12.1.4 remote-as 10.10
 neighbor 10.12.1.4 ebgp-multihop 255
 neighbor 10.8.8.8 remote-as 20
 neighbor 10.8.8.8 update-source Loopback0
 !
 address-family ipv4
  no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
  no bgp default route-target filter
  neighbor 10.12.1.4 activate
  neighbor 10.12.1.4 send-community extended
  neighbor 10.12.1.4 next-hop-self
  neighbor 10.8.8.8 activate
  neighbor 10.8.8.8 send-community extended
  neighbor 10.8.8.8 next-hop-self
 exit-address-family
!
```

RR_B Router

```
interface Loopback0
 ip address 10.8.8.8 255.255.255.255
!
interface Ethernet2/1
 ip address 10.100.100.8 255.255.255.0
 duplex half
!
router ospf 20
 network 10.8.8.8 0.0.0.0 area 0
 network 10.100.100.8 0.0.0.0 area 0
!
router bgp 20
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor rrc peer-group
 neighbor rrc remote-as 20
 neighbor rrc update-source Loopback0
 neighbor 10.5.5.5 peer-group rrc
 neighbor 10.6.6.6 peer-group rrc
 neighbor 10.9.9.9 peer-group rrc
 neighbor 10.9.9.9 shutdown
 !
 address-family ipv4
  no auto-summary
 exit-address-family
```

```
 !
 address-family l2vpn vpls
  neighbor rrc send-community extended
  neighbor rrc route-reflector-client
  neighbor 10.5.5.5 activate
  neighbor 10.6.6.6 activate
  neighbor 10.9.9.9 activate
 exit-address-family
!
```

Additional References for L2VPN VPLS Inter-AS Option B

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
IP Routing (BGP) commands	Cisco IOS IP Routing: BGP Command Reference
Concepts and tasks related to configuring the VPLS Autodiscovery: BGP Based feature.	*VPLS Autodiscovery BGP Based*
BGP support for the L2VPN address family	*BGP Support for the L2VPN Address Family*
VPLS	"VPLS Overview" section in the *Configuring Multiprotocol Label Switching on the Optical Services Modules* document
L2VPN multisegment pseudowires, MPLS OAM support for L2VPN multisegment pseudowires, MPLS OAM support for L2VPN inter-AS option B	*L2VPN Multisegment Pseudowires*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing standards has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4360	*BGP Extended Communities Attribute*
RFC 4364	*BGP/MPLS IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for L2VPN VPLS Inter-AS Option B

Table 115: Feature Information for L2VPN VPLS Inter-AS Option B

Feature Name	Releases	Feature Information
L2VPN VPLS Inter-AS Option B	15.1(1)S Cisco IOS XE Release 3.8S	The L2VPN VPLS Inter-AS Option B feature expands the existing features of VPLS autodiscovery to operate across multiple BGP autonomous systems. Using BGP-based autodiscovery as the underlying framework, the L2VPN VPLS Inter-AS Option B features creates a dynamic multisegmented pseudowire configuration between neighboring ASBRs. The following commands were introduced or modified: **bgp default route-target filter**, **debug xconnect**, **l2 pseudowire routing**, **show ip bgp neighbors**, **show mpls forwarding-table**, **show mpls l2transport vc**, **show xconnect**, **switching-point vcid**, and **terminating-pe tie-breaker**.

Glossary

AGI —Attachment Group Identifier. An identifier common to a group of pseudowires that may be connected.

AII —Attachment individual identifier.

ASBR —Autonomous System Boundary Router.

PE —provider edge router.

NLRI —Network Layer Reachability Information.

SAII —Source Attachment Individual Identifier.

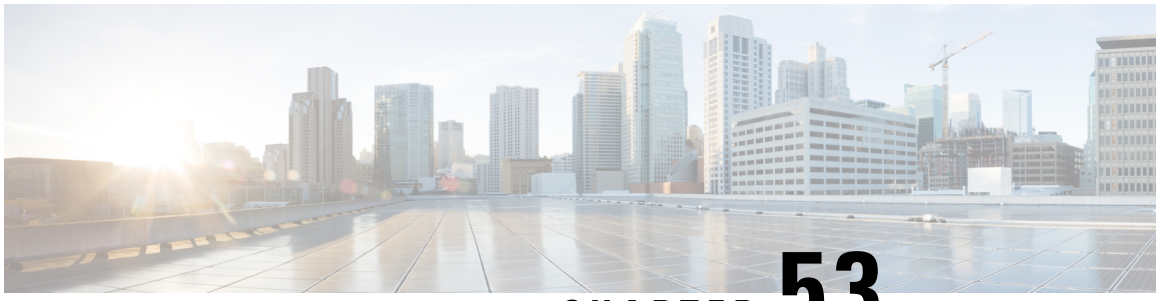
SPE —switching PE.

TAII —Target Attachment Individual Identifier.

TPE —terminating PE.

VFI —virtual forwarding instance. This identifies a group of pseudowires that are associated with a VSI.

VSI —virtual switching instance. This identifies the bridge domain within a single PE. In a single VPLS network, each participating PE has a VSI.



CHAPTER 53

IEEE 802.1Q Tunneling (QinQ) for AToM

This feature allows you to configure IEEE 802.1Q Tunneling (QinQ) for AToM. It also permits the rewriting of QinQ tags for Multiple Protocol Label Switching (MPLS) Layer 2 VPNs (L2VPNs).

Prerequisites for IEEE 802.1Q Tunneling (QinQ) for AToM

The QinQ (short for 802.1Q-in-802.1Q) tunneling and tag rewrite feature is supported on the following line cards:

- 8-port Fast Ethernet line card (ESR-HH-8FE-TX)
- 2-port half-height Gigabit Ethernet line card (ESR-HH-1GE)
- 1-port full-height Gigabit Ethernet line card (ESR-1GE)

Restrictions for IEEE 802.1Q Tunneling (QinQ) for AToM

- Up to a maximum of 447 outer-VLAN IDs and up to 4095 inner VLAN IDs can be supported by this feature.
- Only Unambiguous VLAN tagged Ethernet QinQ interfaces are supported in this release. That is, the Ethernet VLAN QinQ rewrite of both VLAN Tags capability is supported only on Ethernet subinterfaces with a QinQ encapsulation and explicit pair of VLAN IDs defined.

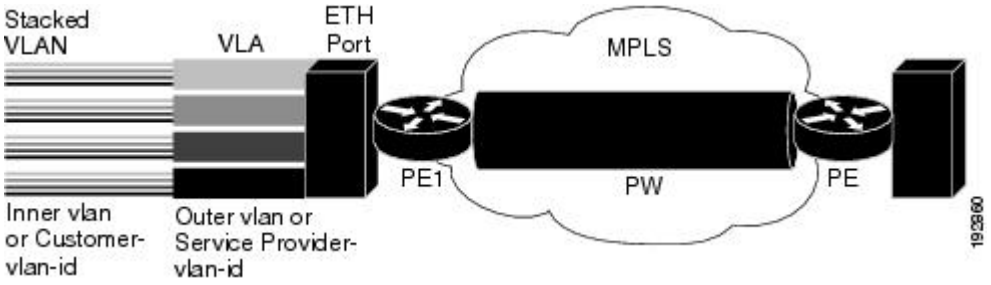
Note Ambiguous inner VLAN IDs are not supported in this release.

Information About IEEE 802.1Q Tunneling (QinQ) for AToM

Ethernet VLAN QinQ AToM

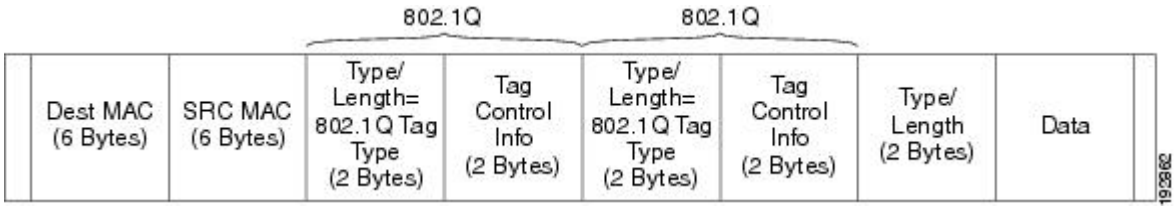
In Metro Ethernet deployment, in which CE routers and PE routers are connected through an Ethernet switched access network, packets that arrive at PE routers can contain up to two IEEE 802.1q VLAN tags (one inner VLAN tag which identifies the customer; and another outer VLAN tag which denotes the customer's service provider). This technique of allowing multiple VLAN tagging on the same Ethernet packet and creating a stack of VLAN IDs is known as QinQ (short for 802.1Q-in-802.1Q). The figure below shows how different edge devices can do L2 switching on the different levels of the VLAN stack.

Figure 73: Ethernet VLAN QinQ



When the outer VLAN tag is the service-delimiting VLAN tag, QinQ packets are processed similar to the ones with one VLAN tag (case previously named Ethernet VLAN Q-in-Q modified, which is already supported in the 12.2(31) SB release). However, when a customer must use a combination of the outer and inner VLAN tags to delimit service for customers, the edge device should be able to choose a unique pseudowire based on a combination of the inner and outer VLAN IDs on the packet shown in the figure below. The customer may want to be able to rewrite both the inner and the outer VLAN IDs on the traffic egress side.

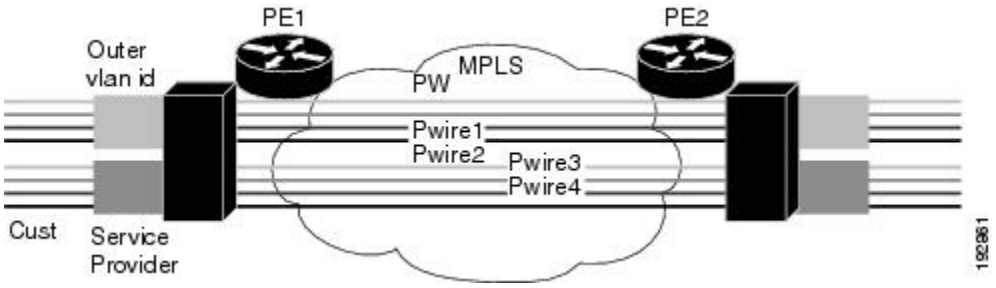
Figure 74: Ethernet VLAN QinQ Header



QinQ Tunneling Based on Inner and Outer VLAN Tags

When handling incoming QinQ Ethernet traffic, the edge router allows a customer to choose a unique pseudowire endpoint to switch the traffic based on the combination of inner and outer VLAN IDs. For example, the figure below shows how a unique pseudowire is selected depending upon the combination of inner (customer edge) and outer (service provider) VLAN IDs. Thus, traffic for different customers can be kept separate.

Figure 75: QinQ Connection



Rewritten Inner and Outer VLAN Tags on QinQ Frames

When managing incoming AToM Ethernet QinQ traffic, the edge router does the following tasks:

1. Strips off the MPLS labels.
2. Allows the customer to rewrite both the inner and outer VLAN IDs before sending the packets to the egress QinQ interface. Note this capability is provided only for AToM like-to-like Ethernet QinQ traffic.

The QinQ AToM feature is a like-to-like interworking case over AToM. This feature requires changes to the microcode to allow it to overwrite two layers of VLAN tags on Ethernet QinQ traffic, transported across AToM pseudowires.

- On the ingress side--The packets preserve their L2 header with the two VLAN tags, and it is sent across the pseudowire with VC type of 4.
- On the egress side--The MPLS label is stripped, and up to two levels of VLAN tags are rewritten per the configuration.

Only Unambiguous VLAN tagged Ethernet QinQ interfaces are supported in this release. The Ethernet VLAN Q-in-Q rewrite of both VLAN Tags capability is supported only on Ethernet subinterfaces with a QinQ encapsulation and explicit pair of VLAN IDs defined.

How to Configure IEEE 802.1Q Tunneling (QinQ) for AToM

This section explains how to configure IEEE 802.1Q Tunneling (QinQ) for AToM and includes the following procedures. While all of the procedures are listed as optional, you must choose one of the first two listed.

Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for AToM

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
5. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.100`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 100 second-dot1q 200`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 5	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.0.0.16 410 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for AToM using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
5. **interface pseudowire** *number*
6. **encapsulation mpls**
7. **neighbor** *peer-address vcid-value*
8. **exit**
9. **l2vpn xconnect context** *context-name*
10. **member pseudowire** *interface-number*

11. **member gigabitethernet** *interface-number*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.100`	Specifies the Gigabit Ethernet interface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 100 second-dot1q 200`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 5	**interface pseudowire** *number* **Example:** `Router(config-if)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 6	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 7	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 9	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 10	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 11	**member gigabitethernet** *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet1/0/0.100`	Specifies the location of the Gigabit Ethernet member interface.
Step 12	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for AToM

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
5. **xconnect** *peer-router-id vcid encapsulation mpls*
6. **exit**
7. **interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*]
8. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
9. **xconnect** *peer-router-id vcid encapsulation mpls*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.200`	Specifies the Gigabit Ethernet subinterface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[*-vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 5	**xconnect** *peer-router-id vcid encapsulation mpls* **Example:** `Router(config-if)# xconnect 10.0.0.16 420 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**interface gigabitethernet** *slot* **/** *subslot* **/** *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.201`	Specifies the next Gigabit Ethernet interface and enters interface configuration mode.
Step 8	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[*-vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 201 second-dot1q any`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 9	**xconnect** *peer-router-id vcid encapsulation mpls* **Example:** `Router(config-if)# xconnect 10.0.0.16 430 encapsulation mpls`	Creates the VC to transport the Layer 2 packets.

Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for AToM using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*]
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
5. **interface pseudowire** *number*
6. **encapsulation mpls**
7. **neighbor** *peer-address vcid-value*
8. **exit**
9. **interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*]
10. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]}
11. **interface pseudowire** *number*
12. **encapsulation mpls**
13. **neighbor** *peer-address vcid-value*
14. **exit**
15. **l2vpn xconnect context** *context-name*
16. **member pseudowire** *interface-number*
17. **member gigabitethernet** *interface-number*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.200`	Specifies the Gigabit Ethernet subinterface and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:**	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.

	Command or Action	Purpose
	`Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000`	
Step 5	**interface pseudowire** *number* **Example:** `Router(config-if)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 6	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 7	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**interface gigabitethernet** *slot* / *subslot* / *port* **.** [*subinterface*] **Example:** `Router(config)# interface GigabitEthernet1/0/0.201`	Specifies the next Gigabit Ethernet interface and enters interface configuration mode.
Step 10	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*[,*vlan-id*[-*vlan-id*]]} **Example:** `Router(config-if)# encapsulation dot1q 201 second-dot1q any`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance.
Step 11	**interface pseudowire** *number* **Example:** `Router(config-if)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 12	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.

	Command or Action	Purpose
Step 13	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 15	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 16	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 17	**member gigabitethernet** *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet1/0/0.201`	Specifies the location of the Gigabit Ethernet member interface.
Step 18	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.

Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration

SUMMARY STEPS

1. **enable**
2. **show mpls l2transport vc**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show mpls l2transport vc** **Example:** `Router# show mpls l2transport vc`	Displays information about Any Transport over MPLS (AToM) virtual circuits (VCs) and static pseudowires that have been enabled to route Layer 2 packets on a router.

Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **show l2vpn atom vc**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show l2vpn atom vc** **Example:** `Device# show l2vpn atom vc`	Displays information about Any Transport over MPLS (AToM) virtual circuits (VCs) and static pseudowires that have been enabled to route Layer 2 packets on a router.

Configuration Examples for IEEE 801.2 Tunneling (QinQ) for ATM

Example Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for ATM

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet1/0/0.100
Router(config-if)# encapsulation dot1q 100 second-dot1q 200
Router(config-if)# xconnect 10.0.0.16 410 encapsulation mpls
```

Example Configuring Unambiguous IEEE 802.1Q Tunneling (QinQ) for ATM using the commands associated with the L2VPN Protocol-Based CLIs feature

```
Router> enable
Router# configure terminal
```

```
Router(config)# interface GigabitEthernet1/0/0.100
Router(config-if)# encapsulation dot1q 100 second-dot1q 200
Router(config-if)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.0.0.1 123
Router(config-if)# exit
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100
Router(config-xconnect)# member GigabitEthernet1/0/0.100
```

Example Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM

The following is an example of an ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM configuration.

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet1/0/0.200
Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000
Router(config-if)# xconnect 10.0.0.16 420 encapsulation mpls
Router(config-if)# exit
Router(config)# interface GigabitEthernet1/0/0.201
Router(config-if) encapsulation dot1q 201 second-dot1q any
Router(config-if) xconnect 10.0.0.16 430 encapsulation mpls
```

Example Configuring Ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM using the commands associated with the L2VPN Protocol-Based CLIs feature

The following is an example of an ambiguous IEEE 802.1Q Tunneling (QinQ) for ATM configuration.

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet1/0/0.200
Router(config-if)# encapsulation dot1q 200 second-dot1q 1000-2000,3000,3500-4000
Router(config-if)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.0.0.1 123
Router(config-if)# exit
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100
Router(config-xconnect)# member GigabitEthernet1/0/0.200
Router(config-xconnect)# exit
Router(config)# interface GigabitEthernet1/0/0.201
Router(config-if) encapsulation dot1q 201 second-dot1q any
Router(config-if)# interface pseudowire 100
Router(config-if)# encapsulation mpls
Router(config-if)# neighbor 10.0.0.1 123
Router(config-if)# exit
Router(config)# l2vpn xconnect context A
Router(config-xconnect)# member pseudowire 100
Router(config-xconnect)# member GigabitEthernet1/0/0.201
```

Example Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration

The following is sample output of the **show mpls l2transport vc** command, which is usedto verify the VC set up in EoMPLS QinQ mode.

```
router# show mpls l2transport vc
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi1/0/0.1      Eth VLAN:100/200           10.1.1.2        1          UP
```

Example Verifying the IEEE 802.1Q Tunneling (QinQ) for ATM Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

The following is sample output of the **show l2vpn atom vc** command, which is used to verify the virtual circuit (VC) set up in EoMPLS QinQ mode.

```
Device# show l2vpn atom vc
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Gi1/0/0.1      Eth VLAN:100/200           10.1.1.2        1          UP
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
AToM and MPLS	Any Transport over MPLS

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

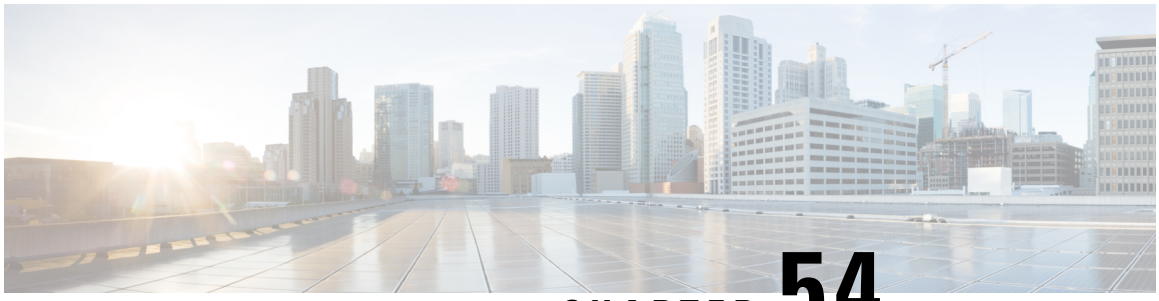
Feature Information for IEEE 802.1Q Tunneling (QinQ) for AToM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 116: Feature Information for IEEE 802.1Q Tunneling (QinQ) for AToM

Feature Name	Releases	Feature Information
IEEE 802.1Q Tunneling (QinQ) for AToM	Cisco IOS XE Release 2.4	This feature allows you to configure IEEE 802.1Q Tunneling (QinQ) for AToM. It also permits the rewriting of QinQ tags for Multiple Protocol Label Switching (MPLS) layer 2 VPNs (L2VPNs). In Cisco IOS XE Release 2.4, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **interface , encapsulation dot1q second-dot1q , xconnect .**



CHAPTER 54

Configuring the Managed IPv6 Layer 2 Tunnel Protocol Network Server

This document describes how to enable the Managed IPv6 Layer 2 Tunnel Protocol Network Server feature.

Prerequisites for Configuring the Managed IPv6 LNS

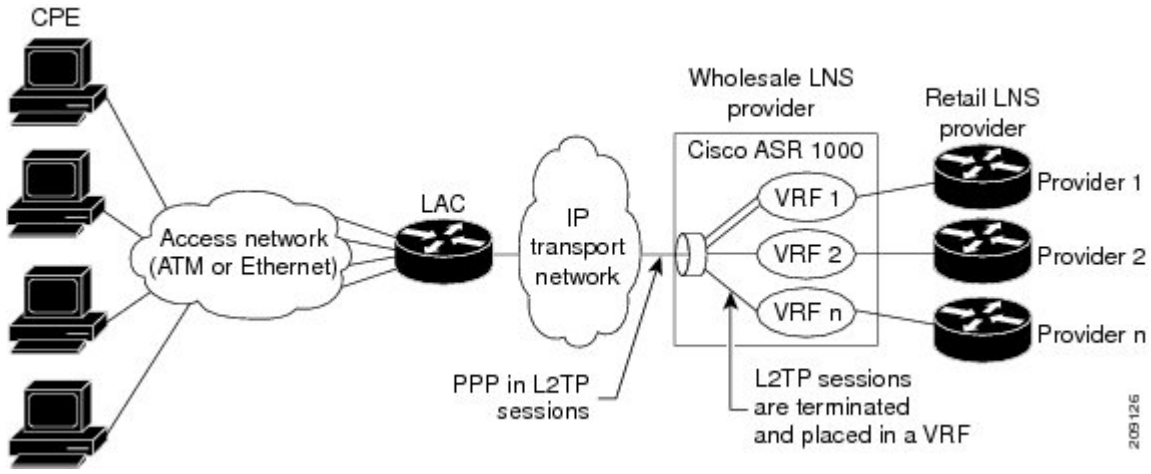
For the router to function as an LNS, you must enable Authentication, Authorization, and Accounting (AAA) on the Layer 2 Tunnel Protocol Network Server (LNS) and the Layer 2 Access Concentrator (LAC), by entering the **aaanew-model** global configuration command. For more information, see the " Authentication, Authorization, and Accounting " chapter in the *Cisco IOS XE Security: Securing User Services Configuration Guide* .

Information About Configuring the Managed IPv6 LNS

L2TP Network Server

The router can function as an LNS. The LNS is a peer to the LAC and sits on one side of an L2TP tunnel. The LNS routes packets to and from the LAC and a destination network. When the router functions as an LNS, you can configure the router to terminate the PPP sessions and route the client IP packets onto the ISP or corporate network toward their final destination (see the figure below). The router can use the Managed IPv6 LNS feature to terminate L2TP sessions from the LAC and place each session into the appropriate IPv6 VRF instance based on the VRF applied to the virtual template interface or alternatively, based on the VRF received for the user through AAA. The router then routes each session within the VRF to the destination network.

Figure 76: Terminating and Forwarding Sessions from the LAC



Tunnel Accounting

The tunnel accounting feature enhances AAA accounting by adding the ability to include tunnel-related statistics in the RADIUS information. Before you can collect tunnel usage information, you must configure the following attributes on the RADIUS server:

- Acct-Tunnel-Connection—Specifies the identifier assigned to the tunnel session. This attribute and the Tunnel-Client-Endpoint and Tunnel-Server-Endpoint attributes provide a way to uniquely identify a tunnel session for auditing purposes.
- Acct-Tunnel-Packets-Lost—Specifies the number of packets lost on a given link.

The table below describes the values for the Acct-Status-Type attribute that support tunnel accounting on the RADIUS server.

Table 117: Acct-Status-Type Values for RADIUS Tunnel Accounting

Acct-Status-Type Values	Value	Description
Tunnel-Link-Reject	14	Marks the rejection of the establishment of a new link in an existing tunnel.
Tunnel-Link-Start	12	Marks the creation of a tunnel link within an L2TP tunnel that carries multiple links.
Tunnel-Link-Stop	13	Marks the destruction of a tunnel link within an L2TP tunnel that carries multiple links.
Tunnel-Reject	11	Marks the rejection of the establishment of a tunnel with another device.
Tunnel-Start	9	Marks the establishment of a tunnel with another device.
Tunnel-Stop	10	Marks the destruction of a tunnel to or from another device.

For more information about the RADIUS tunnel accounting attributes or the Acct-Status-Type values that support RADIUS tunnel accounting, see RFC 2867, RADIUS Accounting Modifications for Tunnel Protocol Support.

For information about RADIUS accounting attributes supported on the Cisco ASR 1000 Series Aggregation Services Routers, see the "RADIUS Attributes" chapter in the Cisco IOS XE Security Configuration Guide: Securing User Services.

For more information on configuring RADIUS, see your RADIUS user documentation.

How to Configure the Managed LNS

Configuring a VRF on the LNS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4**|**ipv6**}
6. **route-target** {*import*|*export*|*both*} *route-target-ext-community*
7. **exit-address-family**
8. **address-family** {**ipv4**|**ipv6**}
9. **route-target** {*import*|*export*|*both*} *route-target-ext-community*
10. **end**
11. **show ipv6 route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf1`	Configures a VRF routing table and enters VRF configuration mode. • The *vrf-name*argument is the name of the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.

	Command or Action	Purpose
		You can enter a route distinguisher in either of these formats: • 16-bit autonomous system number (ASN): your 32-bit number For example, 101:3. • 32-bit IP address: your 16-bit number For example, 192.168.122.15:1.
Step 5	**address-family** {**ipv4**|**ipv6**} **Example:** `Router(config-vrf) address-family ipv6`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF. • The **ipv6** keyword specifies an IPv6 address family for a VRF.
Step 6	**route-target** {*import*|*export*|*both*} *route-target-ext-community* **Example:** `Router(config-vrf-af) route-target both 100:2`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF list of import, export, or both (import and export) route-target extended communities.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 8	**address-family** {**ipv4**|**ipv6**} **Example:** `Router(config-vrf) address-family ipv6`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF. • The **ipv6** keyword specifies an IPv6 address family for a VRF.
Step 9	**route-target** {*import*|*export*|*both*} *route-target-ext-community* **Example:**	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community.

	Command or Action	Purpose
	`Router(config-vrf-af)# route-target both 100:3`	• The **export** keyword specifies to export routing information to the target VPN extended community. • The **both** keyword specifies to import both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF list of import, export, or both (import and export) route-target extended communities. • Enter the **route-target** command one time for each target community.
Step 10	**end** **Example:** `Router(config-vrf-af)# end`	Exits VRF address family configuration mode and returns to privileged EXEC mode.
Step 11	**show ipv6 route vrf** *vrf-name* **Example:** `Router# show ipv6 route vrf vrf1`	Displays the IPv6 routing table associated with a VRF.

Configuring a Virtual Template Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **vrf forwarding** *name*
5. **ppp authentication chap**
6. **end**
7. **show interfaces virtual-access** *number* **[configuration]**
8. **debug ppp chap**
9. **debug ppp negotiation**
10. **debug ppp negotiation chap**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enters privileged EXEC mode.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template interface and enters interface configuration mode.
Step 4	**vrf forwarding** *name* **Example:** `Router(config-if)# vrf forwarding vpn-1`	(Optional) Maps the virtual template interface to a VRF routing table. **Note** If the VRF assignment is received via the RADIUS server, then this step is not required.
Step 5	**ppp authentication chap** **Example:** `Router(config-if)# ppp authentication chap`	Enables CHAP authentication on the virtual template interface, which is applied to virtual access interfaces (VAI).
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show interfaces virtual-access** *number* **[configuration]** **Example:** `Router# show interfaces virtual-access number [configuration]`	Displays status, traffic data, and configuration information about the VAI you specify.
Step 8	**debug ppp chap** **Example:** `Router# debug ppp chap`	Displays authentication protocol messages for Challenge Authentication Protocol (CHAP) packet exchanges. • This command is useful when a CHAP authentication failure occurs due to a configuration mismatch between devices. Verifying and correcting any username and password mismatch resolves the problem.
Step 9	**debug ppp negotiation** **Example:**	Displays information on traffic and exchanges in an internetwork implementing PPP.

	Command or Action	Purpose
	`Router# debug ppp negotiation`	
Step 10	**debug ppp negotiation chap** **Example:** `Router# debug ppp negotiation chap`	Deciphers a CHAP negotiation problem due to a connectivity problem between a Cisco and non-Cisco device.

Assigning a VRF via the RADIUS Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization configuration** *method-name* **group** *group-name*
4. **ipv6 dhcp pool** *pool-name*
5. **prefix-delegation aaa** [**method-list***method-list*]
6. **dns-server** *ipv6-address*
7. **exit**
8. **interface virtual-template** *number*
9. **ipv6 nd prefix framed-ipv6-prefix**
10. **ipv6 dhcp server** *pool-name* **rapid-commit**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authorization configuration** *method-name* **group** *group-name* **Example:** `Router(config)# aaa authorization configuration DHCPv6-PD group DHCPv6-PD-RADIUS`	Downloads configuration information from the AAA server using RADIUS.

	Command or Action	Purpose
Step 4	**ipv6 dhcp pool** *pool-name* **Example:** `Router(config)# ipv6 dhcp pool DHCPv6-PD`	Configures a DHCP for IPv6 configuration information pool and enters DHCP for IPv6 pool configuration mode.
Step 5	**prefix-delegation aaa** [**method-list***method-list*] **Example:** `Router(config-dhcpv6)# prefix-delegation aaa method-list DHCPv6-PD`	Specifies that prefixes are to be acquired from AAA servers.
Step 6	**dns-server** *ipv6-address* **Example:** `Router(config-dhcpv6)# dns-server 2001:0DB8:3000:3000::42`	Specifies the Domain Name System (DNS) IPv6 servers available to a DHCP for IPv6 client.
Step 7	**exit** **Example:** `Router(config-dhcpv6)# exit`	Exits DHCP for IPv6 pool configuration mode and enters global configuration mode.
Step 8	**interface virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating VAIs, and enters interface configuration mode.
Step 9	**ipv6 nd prefix framed-ipv6-prefix** **Example:** `Router(config-if)# ipv6 nd prefix framed-ipv6-prefix`	Adds the prefix in a received RADIUS framed IPv6 prefix attribute to the interface's neighbor discovery prefix queue.
Step 10	**ipv6 dhcp server** *pool-name* **rapid-commit** **Example:** `Router(config-if)# ipv6 dhcp server DHCPv6-PD rapid-commit`	Enables DHCPv6 on an interface.
Step 11	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the LNS to Initiate and Receive L2TP Traffic

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn enable**
4. **vpdn-group** *group-name*
5. **accept-dialin**
6. **protocol 12tp**
7. **virtual-template** *template-number*
8. **exit**
9. **terminate-from hostname** *hostname*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vpdn enable** **Example:** `Router(config)# vpdn enable`	Enables VPDN networking on the router and informs the router to look for tunnel definitions in a local database and on a remote authorization server (home gateway) if one is present.
Step 4	**vpdn-group** *group-name* **Example:** `Router(config)# vpdn-group group1`	Defines a local group name for which you can assign other VPDN variables. • Enters VPDN group configuration mode.
Step 5	**accept-dialin** **Example:** `Router(config-vpdn)# accept-dialin`	Configures the LNS to accept tunneled PPP connections from the LAC and creates an accept-dialin VPDN subgroup. • Enters accept dial-in VPDN subgroup configuration mode.
Step 6	**protocol 12tp** **Example:**	Specifies the Layer 2 Tunnel Protocol.

	Command or Action	Purpose
	`Router(config-vpdn-acc-in)# protocol l2tp`	
Step 7	**virtual-template** *template-number* **Example:** `Router(config-vpdn-acc-in)# virtual-template 1`	Specifies the virtual template to be used to clone VAIs.
Step 8	**exit** **Example:** `Router(config-vpdn-acc-in)# exit`	Returns to VPDN group configuration mode.
Step 9	**terminate-from hostname** *hostname* **Example:** `Router(config-vpdn)# terminate-from hostname lac1-vpn1`	Specifies the hostname of the remote LAC that is required when accepting a VPDN tunnel.
Step 10	**end** **Example:** `Router(config-vpdn)# end`	Exits VPDN configuration mode and returns to privileged EXEC mode.

Limiting the Number of Sessions per Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn-group** *group-name*
4. **accept-dialin**
5. **protocol 12tp**
6. **virtual-template** *template-number*
7. **exit**
8. **terminate-from hostname** *host-name*
9. **session-limit** *limit-number*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vpdn-group** *group-name* **Example:** `Router(config)# vpdn-group group1`	Defines a local group name for which you can assign other VPDN variables. • Enters VPDN group configuration mode.
Step 4	**accept-dialin** **Example:** `Router(config-vpdn)# accept-dialin`	Configures the LNS to accept tunneled PPP connections from the LAC and creates an accept-dialin VPDN subgroup. • Enters accept dial-in VPDN subgroup configuration mode.
Step 5	**protocol 12tp** **Example:** `Router(config-vpdn-acc-in)# protocol 12tp`	Specifies the Layer 2 Tunnel Protocol.
Step 6	**virtual-template** *template-number* **Example:** `Router(config-vpdn-acc-in)# virtual-template 1`	Specifies the virtual template to be used to clone VAIs.
Step 7	**exit** **Example:** `Router(config-vpdn-acc-in)# exit`	Returns to VPDN group configuration mode.
Step 8	**terminate-from hostname** *host-name* **Example:** `Router(config-vpdn)# terminate-from hostname test_LAC`	Specifies the hostname of the remote LAC that is required when accepting a VPDN tunnel.
Step 9	**session-limit** *limit-number* **Example:** `Router(config-vpdn)# session-limit 100`	Specifies the maximum number of sessions per tunnel.
Step 10	**exit** **Example:** `Router(config-vpdn)# exit`	Exits VPDN configuration mode and returns to privileged EXEC mode.

Configuring RADIUS Attribute Accept or Reject Lists

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication ppp default group** *group-name*
4. **aaa authorization network group group** *group-name*
5. **aaa group server radius** *group-name*
6. **server-private** *ip-address* [**acct-port***port-number*][**timeout***seconds*] [**retransmit***retries*] [**key***string*]
7. **authorization** [**accept**|**reject**] *list-name*
8. **exit**
9. **radius-server attribute list** *listname*
10. **attribute** *value1* [*value2* [*value3...*]]
11. **end**
12. **show accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication ppp default group** *group-name* **Example:** `Router(config)# aaa authentication ppp default group radius_authen1`	Specifies one or more AAA authentication methods for use on serial interfaces running PPP.
Step 4	**aaa authorization network group group** *group-name* **Example:** `Router(config)# aaa authorization network group group radius_authen1`	Sets the parameters that restrict network access to the user.
Step 5	**aaa group server radius** *group-name* **Example:** `Router(config)# aaa group server radius VPDN-Group`	Groups different RADIUS server hosts into distinct lists and distinct methods and enters server group RADIUS configuration mode.

	Command or Action	Purpose
Step 6	**server-private** *ip-address* [**acct-port***port-number*][**timeout***seconds*] [**retransmit***retries*] [**key***string*] **Example:** `Router(config-sg-radius)# server-private 10.1.1.2 acct-port 0 timeout 7 retransmit 3 key cisco1`	Configures the IP address of the private RADIUS server for the group server. • The *ip-address*argument specifies the IP address of the private RADIUS server host. • (Optional) The *port-number*argument specifies the UDP destination port for accounting requests. • (Optional) The *seconds* argument specifies the timeout value (1 to 1000). • (Optional) The *retries*argument specifies the number of times a RADIUS request is re-sent to a server, if that server is not responding or responding slowly. • The *string* argument specifies the authentication and encryption key for all RADIUS communications between the router and the RADIUS server.
Step 7	**authorization** [**accept**|**reject**] *list-name* **Example:** `Router(config-sg-radius)# authorization accept vpn1-autho-list`	Specifies a filter for the attributes that are returned in an Access-Accept packet from the RADIUS server. • The **accept** keyword indicates that all attributes will be rejected except the attributes specified in the *listname* argument. • The **reject** keyword indicates that all attributes will be accepted except for the attributes specified in the *listname* argument and all standard attributes.
Step 8	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits server group RADIUS configuration mode and enters global configuration mode.
Step 9	**radius-server attribute list** *listname* **Example:** `Router(config)# radius-server attribute list vpn1-autho-list`	Defines the list name given to the set of attributes defined using the **attribute** command and enters RADIUS attribute list configuration mode. • Define the *listname* argument to be the same as you defined it in step 7.
Step 10	**attribute** *value1* [*value2* [*value3...*]] **Example:** `Router(config-radius-attrl)# attribute 26,200`	Adds attributes to the configured accept or reject list. • You can use this command multiple times to add attributes to an accept or reject list.
Step 11	**end** **Example:**	Exits RADIUS attribute list configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-radius-attrl)# end`	
Step 12	**show accounting** **Example:** `Router# show accounting`	Displays accounting records for users currently logged in. • Displays active accountable events on the network and helps collect information in the event of a data loss on the accounting server.

Configuring AAA Accounting Using Named Method Lists

Note System accounting does not use named method lists. For system accounting you can define only the default method list. For more information, see the "Configuring Accounting " chapter in the Cisco IOS XE Security Configuration Guide: Securing User Services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting network** *list-name* **start-stop group radius**
4. **line** [**aux** | **console**| **vty**] [*line-number*]
5. **accounting** {**arap**|**commands***level*|**connection**|**exec**|**resource**} [**default** | *list-name*]
6. **end**
7. **debug aaa accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa accounting network** *list-name* **start-stop group radius** **Example:** `Router(config)# aaa accounting network methodlist start-stop group radius`	Creates an accounting method list and enables accounting.

	Command or Action	Purpose
Step 4	**line** [**aux** | **console**| **vty**] [*line-number*] **Example:** `Router(config)# line console 0`	Enters line configuration mode for the line to which you want to apply the accounting method list.
Step 5	**accounting** {**arap**|**commands***level*|**connection**|**exec**|**resource**} [**default** | *list-name*] **Example:** `Router(config-line)# accounting commands 15 list1`	Applies the accounting method list to a line or a set of lines.
Step 6	**end** **Example:** `Router(config-line)# end`	Exits line configuration mode and returns to privileged EXEC mode.
Step 7	**debug aaa accounting** **Example:** `Router# debug aaa accounting`	Displays information on accountable events as they occur.

Configuring RADIUS Tunnel Authentication Method Lists on the LNS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization network** *list-name method1* [*method2...*]
4. **vpdn tunnel authorization network** *lmethod-ist-name method1* [*method2...*]
5. **vpdn tunnel authorization virtual-template** *vtemplate-number*
6. **vpdn tunnel authorization password** *dummy-password*
7. **debug aaa authorization**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**aaa authorization network** *list-name method1* [*method2...*] **Example:** `Router(config)# aaa authorization network mymethodlist group VPDN-Group`	Sets parameters that restrict user access to a network. • The *list-name* argument is a character string used to name the list of authentication methods tried when a user logs in. • **group radius**: Uses the list of all RADIUS servers for authentication. • **group** *group-name*: Uses a subset of RADIUS servers for authentication as defined by the **aaa group server radius** command. • **if-authenticated**: Succeeds if user has been successfully authenticated. • **local**: Uses the local username database for authentication. • **none**: Uses no authentication. **Note** The method list is only for VPDN tunnel authorization and termination, not for domain and Digital Number Identification Service (DNIS) authorization. Therefore, the method list applies only on the tunnel terminator device - the LAC for dialout sessions and the LNS for dialin sessions.
Step 4	**vpdn tunnel authorization network** *lmethod-ist-name method1* [*method2...*] **Example:** `Router(config)# vpdn tunnel authorization network mymethodlist`	Specifies the AAA method list to use for VPDN remote tunnel hostname-based authorization. • If you do not specify a method list (including a default method list) by using the **vpdn tunnel authorization network** command, local authorization occurs by using the local VPDN group configuration.
Step 5	**vpdn tunnel authorization virtual-template** *vtemplate-number* **Example:** `Router(config)# vpdn tunnel authorization virtual-template 10`	Specifies the default virtual template interface used to clone a VAI. • If you do not specify a virtual template interface in the local VPDN group configuration or in a remote RADIUS configuration, then the default virtual template interface is used.
Step 6	**vpdn tunnel authorization password** *dummy-password* **Example:**	Specifies the password to use for the RADIUS authorization request to retrieve the tunnel configuration based on the remote tunnel hostname.

	Command or Action	Purpose
	`Router(config)# vpdn tunnel authorization password mypassword`	
Step 7	**debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays information on AAA authorization.

Configuring the LNS for RADIUS Tunnel Authentication

Perform the following tasks to configure LNS for RADIUS Tunnel Authentication:

Note Cisco ASR 1000 Series Aggregation Services Routers supports L2TP tunnel authorization. However, RADIUS does not provide attributes for such parameter values as L2TP tunnel timeouts, L2TP tunnel hello intervals, and L2TP tunnel receive window size. When the Cisco ASR 1000 Series Aggregation Services Router does not receive a RADIUS attribute for a parameter, the router uses the default value.

Configuring RADIUS Tunnel Authentication Method Lists on the LNS

To configure method lists on the LNS for RADIUS tunnel authentication, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization network** *list-name method1* [*method2...*]
4. **vpdn tunnel authorization network** *method- list-name*
5. **vpdn tunnel authorization virtual-template** *vtemplate-number*
6. **vpdn tunnel authorization password** *dummy-password*
7. **end**
8. **debug aaa authorization**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**aaa authorization network** *list-name method1* [*method2...*] **Example:** `Router(config)# aaa authorization network mymethodlist group VPDN-Group`	Sets parameters that restrict user access to a network • The*list-name*argument is a character string used to name the list of authentication methods tried when a user logs in. • **group**radius—Uses the list of all RADIUS servers for authentication. • **group***group-name*—Uses a subset of RADIUS servers for authentication as defined by the **aaagroupserverradius** command. • **if-authenticated**—Succeeds if user has been successfully authenticated. • **local**—Uses the local username database for authentication. • **none**—Uses no authentication. **Note** The method list is only for VPDN tunnel authorization and termination, not for domain and Digital Number Identification Service (DNIS) authorization. Therefore, the method list applies only on the tunnel terminator device—the LAC for dialout sessions and the LNS for dialin sessions.
Step 4	**vpdn tunnel authorization network** *method- list-name* **Example:** `Router(config)# vpdn tunnel authorization network mymethodlist`	Specifies the AAA method list to use for VPDN remote tunnel hostname-based authorization. • If you do not specify a method list (including a default method list) by using the **vpdntunnelauthorizationnetwork** command, local authorization occurs by using the local VPDN group configuration.
Step 5	**vpdn tunnel authorization virtual-template** *vtemplate-number* **Example:** `Router(config)# vpdn tunnel authorization virtual-template 10`	Specifies the default virtual template interface used to clone a VAI. • If you do not specify a virtual template interface in the local VPDN group configuration or in a remote RADIUS configuration, then the default virtual template interface is used. **Note** The **vpdntunnelauthorizationvirtual-template** command is applicable only on the LNS.

	Command or Action	Purpose
Step 6	**vpdn tunnel authorization password** *dummy-password* **Example:** `Router(config)# vpdn tunnel authorization password mypassword`	Specifies the password to use for the RADIUS authorization request to retrieve the tunnel configuration based on the remote tunnel hostname. • By default, the password is cisco, but you can configure a different password. **Note** The **vpdntunnelauthorizationpassword** command is applicable on both the LAC and LNS.
Step 7	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays information on AAA authorization.

Configuring AAA Authentication Methods

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. Configure RADIUS security protocol parameters. For more information about RADIUS, see the "Configuring RADIUS" chapter in the Cisco IOS XE Security Configuration Guide: Securing User Services .
5. **aaa authentication**
6. Apply the authentication method lists to an interface, a line, or a set of lines as required. For more information about authentication method lists, see the "Configuring Authentication" chapter in the Cisco IOS XE Security Configuration Guide: Securing User Services .
7. **end**

DETAILED STEPS

Step 1 **enable**

Step 2 **configure terminal**

Step 3 **aaa new-model**

Enter this command in global configuration mode to enable AAA.

Step 4 Configure RADIUS security protocol parameters. For more information about RADIUS, see the "Configuring RADIUS" chapter in the Cisco IOS XE Security Configuration Guide: Securing User Services .

Step 5 **aaa authentication**

Enter this command to define the authentication method lists.

Step 6 Apply the authentication method lists to an interface, a line, or a set of lines as required. For more information about authentication method lists, see the "Configuring Authentication" chapter in the Cisco IOS XE Security Configuration Guide: Securing User Services .

Step 7 **end**

Configuration Examples for the Managed IPv6 Layer 2 Tunnel Protocol Network Server

Example Managed IPv6 LNS Configuration

The following example shows how to configure Managed IPv6 LNS features on the router. In this example, the router terminates the tunnel from the LAC and associates the VRFs with the interfaces and the virtual template interfaces. This configuration also shows how to configure RADIUS attribute screening and AAA accounting for the VRFs.

```
!
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
vrf definition user_vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv6
 exit-address-family
!
logging buffered 10000000
enable password lab
!
aaa new-model
!
!
aaa group server radius radius_authen1
 server-private 10.1.1.2 acct-port 0 timeout 7 retransmit 3 key cisco1
 ip radius source-interface Loopback20000
!
aaa authentication login default none
aaa authentication ppp default group radius_authen1
aaa authorization network default group radius_authen1
```

```
aaa authorization configuration DHCPv6-PD group radius_authen1
!
!
!
!
!
aaa session-id common
aaa policy interface-config allow-subinterface
ppp hold-queue 80000
clock timezone EST -5 0
ip source-route
no ip gratuitous-arps
!
!
!
!
!
!
!
no ip domain lookup
ip host mcp-matrix 10.0.0.2
ip host mcp-sun-2 10.0.0.2
!
!
ipv6 unicast-routing
ipv6 dhcp binding track ppp
ipv6 dhcp pool ipv6_dhcp_pool1
 prefix-delegation aaa method-list DHCPv6-PD
!
!
!
!
multilink bundle-name authenticated
vpdn enable
!
vpdn-group VPDN_LNS1
 accept-dialin
  protocol l2tp
  virtual-template 1
 terminate-from hostname test_LAC1
 source-ip 10.0.0.2
 local name test_LNS1
 l2tp tunnel password 0 tunnel1
 l2tp tunnel receive-window 100
 l2tp tunnel timeout no-session 30
 l2tp tunnel retransmit retries 7
 l2tp tunnel retransmit timeout min 2
!
!
no virtual-template snmp
!
!
!
!
!
!
!
!
!
username asifp1@test1 password 0 hello1
!
redundancy
 notification-timer 30000
 mode none
```

```
!
!
!
!
!
ip tftp source-interface GigabitEthernet 0
!
!
!
!
!
!
!
!
interface Loopback1
 no ip address
!
interface Loopback20000
 ip address 209.165.202.131 255.255.255.224
!
interface GigabitEthernet1/1/0
 mac-address 8888.8888.8888
 no ip address
 load-interval 30
 negotiation auto
 hold-queue 4096 in
 hold-queue 4096 out
!
interface GigabitEthernet1/1/0.1
 encapsulation dot1Q 3
 ip address 209.165.202.132 255.255.255.224
!
interface GigabitEthernet1/1/1
 mac-address 4444.4444.4444
 no ip address
 load-interval 30
 no negotiation auto
 hold-queue 4096 in
 hold-queue 4096 out
!
interface GigabitEthernet1/1/1.1
 vrf forwarding user_vrf1
 encapsulation dot1Q 2
 ipv6 address 12::1/72
!
interface GigabitEthernet1/1/2
 no ip address
 negotiation auto
!
interface GigabitEthernet1/1/3
 no ip address
 negotiation auto
!
interface GigabitEthernet1/1/4
 no ip address
 negotiation auto
!
interface GigabitEthernet1/1/5
 no ip address
 negotiation auto
!
interface GigabitEthernet1/1/6
 no ip address
 negotiation auto
```

```
!
interface GigabitEthernet1/1/7
 description Connected to RADIUS
 ip address 209.165.201.1 255.255.255.224
 negotiation auto
!
interface GigabitEthernet1/3/0
 no ip address
 media-type sfp
 negotiation auto
!
interface GigabitEthernet1/3/1
 no ip address
 media-type sfp
 negotiation auto
!
interface GigabitEthernet 0
 vrf forwarding Mgmt-intf
 ip address 209.165.201.1 255.255.255.224
 negotiation auto
!
interface Virtual-Template 1
 no ip address
 no logging event link-status
 ipv6 dhcp server ipv6_dhcp_pool1 rapid-commit
 keepalive 30
 ppp mtu adaptive
 ppp authentication pap
!
ip default-gateway 10.1.0.5
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip route vrf Mgmt-intf 209.165.201.1 255.255.255.254 172.16.1.1
ip route vrf Mgmt-intf 209.165.201.29 255.255.255.224 172.16.0.1
!
ip radius source-interface GigabitEthernet1/1/7
logging esm config
cdp run
ipv6 route vrf user_vrf1 ::/0 12::2
!
ipv6 neighbor 12::2 GigabitEthernet1/1/1.1 2222.2222.2222
!
!
!
control-plane
!
call admission limit 90
!
!
!
alias exec call show caller summ
alias exec caller show caller summ
alias exec palt show plat
alias exec plat show platform
alias exec evsi sho plat hard cpp act feat ess stat
!
line con 0
 exec-timeout 0 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 password password1
```

```
!
exception data-corruption buffer truncate
end
```

Example LNS Tunnel Accounting Configuration

The following example shows how to configure the LNS to send tunnel accounting records to the RADIUS server:

```
aaa new-model
!
!
aaa accounting network m1 start-stop group radius
aaa accounting network m2 stop-only group radius
aaa session-id common
enable secret 5 $1$ftf.$wE6Q5Yv6hmQiwL9pizPCg1
!
username ENT_LNS password 0 tunnelpass
username user1@example.com password 0 lab
username user2@example.com password 0 lab
spe 1/0 1/7
firmware location system:/ucode/mica_port_firmware
spe 2/0 2/9
firmware location system:/ucode/mica_port_firmware
!
!
resource-pool disable
clock timezone est 2
!
ip subnet-zero
no ip domain-lookup
ip host CALLGEN-SECURITY-V2 10.24.80.28 10.47.0.0
ip host dirt 172.16.1.129
!
vpdn enable
vpdn tunnel accounting network m1
vpdn session accounting network m1
!
vpdn-group 1
accept-dialin
protocol l2tp
virtual-template 1
terminate-from hostname ISP_LAC
local name ENT_LNS
!
isdn switch-type primary-5ess
!
!
fax interface-type modem
mta receive maximum-recipients 0
!
interface Loopback 0
ip address 172.16.0.101 255.255.255.0
!
interface Loopback 1
ip address 192.168.0.101 255.255.255.0
!
interface Ethernet 0
ip address 10.1.26.71 255.255.255.0
no ip mroute-cache
no cdp enable
!
```

```
interface virtual-template 1
ip unnumbered Loopback 0
peer default ip address pool vpdn-pool1
ppp authentication chap
!
interface virtual-template 2
ip unnumbered Loopback1
peer default ip address pool vpdn-pool2
ppp authentication chap
!
interface fastethernet 0
no ip address
no ip mroute-cache
shutdown
duplex auto
speed auto
no cdp enable
!
ip local pool vpdn-pool1 172.16.5.1 172.16.128.100
ip local pool vpdn-pool2 10.0.0.1 10.0.0.100
ip default-gateway 10.1.26.254
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.26.254
ip route 192.168.1.2 255.255.255.255 10.1.26.254
no ip http server
ip pim bidir-enable
!
!
dialer-list 1 protocol ip permit
no cdp run
!
!
radius-server host 172.16.192.80 auth-port 1645 acct-port 1646 key rad123
radius-server retransmit 3
call rsvp-sync
end
```

Note For additional accounting examples, see the “Configuring Accounting” chapter in the Cisco IOS XE Security: Secure Services Configuration Guide .

Example Verifying the User Profile on the RADIUS Server

The following is an example user profile on the RADIUS server. The Cisco ASR 1000 Series Aggregation Services Routers retrieves the information in the user profile from the RADIUS server.

```
Radius Profile "user1"
Auth-Type = Local, User-Password = "pwd"
User-Service-Type = Framed-User
Framed-Protocol = PPP
cisco-avpair = "lcp:interface-config=vrf forwarding VRF01"
cisco-avpair = "lcp:interface-config=ipv6 unnumbered loopback1"
Framed-IPv6-Prefix = "2001:DB8:4567:1234::/64"
Delegated-IPv6-Prefix = "2001:DB8:AAAA::/48"
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS XE MPLS commands	*Cisco IOS MPLS Command Reference*
Authentication, authorization and accounting	Authentication, Authorization, and Accounting (AAA)
Configuring RADIUS	Configuring RADIUS
Configuring accounting	Configuring Accounting
RADIUS attributes	"RADIUS Attributes Overview and RADIUS IETF Attributes" module in the *Cisco IOS XE Security Configuration Guide: Securing User Services*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2867	RADIUS Accounting Modifications for Tunnel Protocol Support

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Managed IPv6 Layer 2 Tunnel Protocol Network Server

Table 118: Feature Information for Managed IPv6 Layer 2 Tunnel Protocol Network Server

Feature Name	Releases	Feature Information
Managed IPv6 Layer 2 Tunnel Protocol Network Server	Cisco IOS XE Release 3.3S	The Managed IPv6 LNS feature allows the service provider to offer a scalable end-to-end VPN of both IPv4 and IPv6 service to remote users. This feature integrates the Multiprotocol Label Switching (MPLS)-enabled backbone with broadband access capabilities. The following commands were introduced or modified: atm pppatm passive, radius-server attribute list, radius-server key, radius-server retransmit, radius-server vsa send.
Managed IPv6 Layer 2 Tunnel Protocol Network Server - VRF-Lite only	Cisco IOS XE Release 3.3S	The Managed IPv6 LNS feature allows the service provider to offer a scalable end-to-end VPN of both IPv4 and IPv6 service to remote users. This feature integrates the VRF-Lite enabled backbone with broadband access capabilities.
Managed IPv6 Layer 2 Tunnel Protocol Network Server - MPLS VPN	Cisco IOS XE Release 3.7S	The Managed IPv6 LNS feature allows the service provider to offer a scalable end-to-end VPN of both IPv4 and IPv6 service to remote users. This feature integrates the MPLS enabled backbone with broadband access capabilities.

CHAPTER 55

L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature lets you configure your network to detect a failure in the network and reroute the Layer 2 (L2) service to another endpoint that can continue to provide service. This feature provides the ability to recover from a failure either of the remote provider edge (PE) router or of the link between the PE and customer edge (CE) routers.

Prerequisites for L2VPN Pseudowire Redundancy

- This feature module requires that you understand how to configure basic L2 virtual private networks (VPNs).
  - Any Transport over MPLS
  - L2 VPN Interworking
  - Layer 2 Tunneling Protocol Version 3 (L2TPv3)
- The L2VPN Pseudowire Redundancy feature requires that the following mechanisms be in place to enable you to detect a failure in the network:
  - Label-switched paths (LSP) Ping/Traceroute and Any Transport over MPLS Virtual Circuit Connection Verification (AToM VCCV)
  - Local Management Interface (LMI)
  - Operation, Administration, and Maintenance (OAM)

Restrictions for L2VPN Pseudowire Redundancy

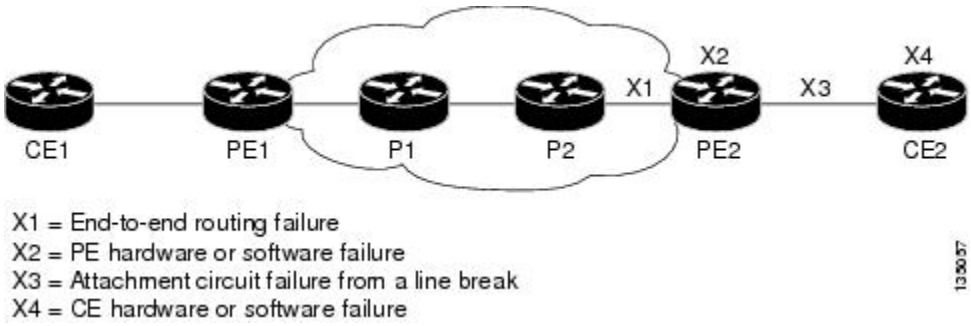
- The default Label Distribution Protocol (LDP) session hold-down timer will enable the software to detect failures in about 180 seconds. That time can be configured so that the software can detect failures more quickly. See the **mpls ldp holdtime** command for more information.
- L2VPN Pseudowire Redundancy does not support pseudowire interworking mode with L2TPv3. The connectivity between CEs may be impacted if you have interworking IP configured in the pseudowire class.
- The primary and backup pseudowires must run the same type of transport service. The primary and backup pseudowires must be configured with AToM or L2TPv3.
- The backup peer can only be configured for nonstatic L2TPv3 sessions. The backup L2TPv3 session cannot be static L2TPv3 session. The encapsulation type of primary and backup pseudowire must be the same.
- If you use L2VPN Pseudowire Redundancy with L2VPN Interworking, the interworking method must be the same for the primary and backup pseudowires.
- L2VPN Pseudowire Redundancy does support setting the experimental (EXP) bit on the Multiprotocol Label Switching (MPLS) pseudowire.
- L2VPN Pseudowire Redundancy does not support different pseudowire encapsulation types on the MPLS pseudowire.
- The **mpls l2transport route** command is not supported. Use the **xconnect** command instead.
- The ability to have the backup pseudowire fully operational at the same time that the primary pseudowire is operational is not supported. The backup pseudowire becomes active only after the primary pseudowire fails.
- The AToM VCCV feature is supported only on the active pseudowire.
- More than one backup pseudowire is not supported.

Information About L2VPN Pseudowire Redundancy

Introduction to L2VPN Pseudowire Redundancy

L2VPNs can provide pseudowire resiliency through their routing protocols. When connectivity between end-to-end PE routers fails, an alternative path to the directed LDP session and the user data can take over. However, there are some parts of the network where this rerouting mechanism does not protect against interruptions in service. The figure below shows those parts of the network that are vulnerable to an interruption in service.

Figure 77: Points of Potential Failure in an L2VPN Network

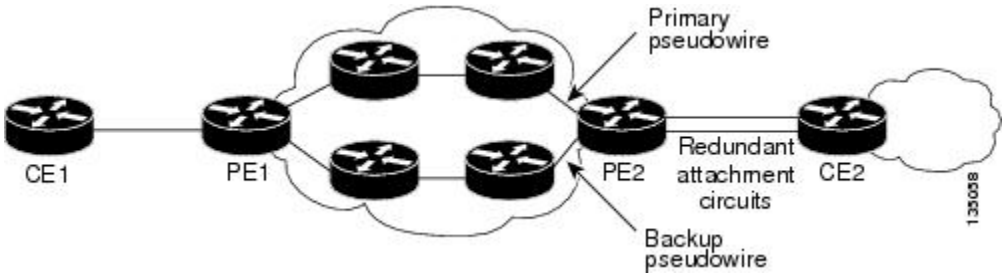


The L2VPN Pseudowire Redundancy feature provides the ability to ensure that the CE2 router in the figure above can always maintain network connectivity, even if one or all the failures in the figure occur.

The L2VPN Pseudowire Redundancy feature enables you to set up backup pseudowires. You can configure the network with redundant pseudowires and redundant network elements, which are shown in the three figures below.

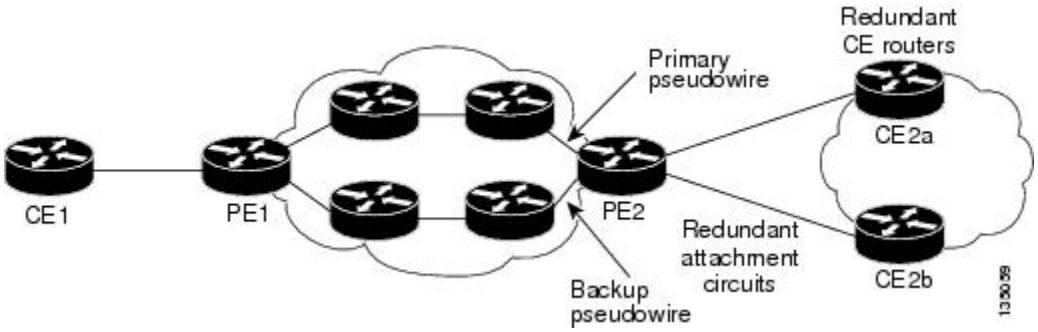
The figure below shows a network with redundant pseudowires and redundant attachment circuits.

Figure 78: L2VPN Network with Redundant PWs and Attachment Circuits



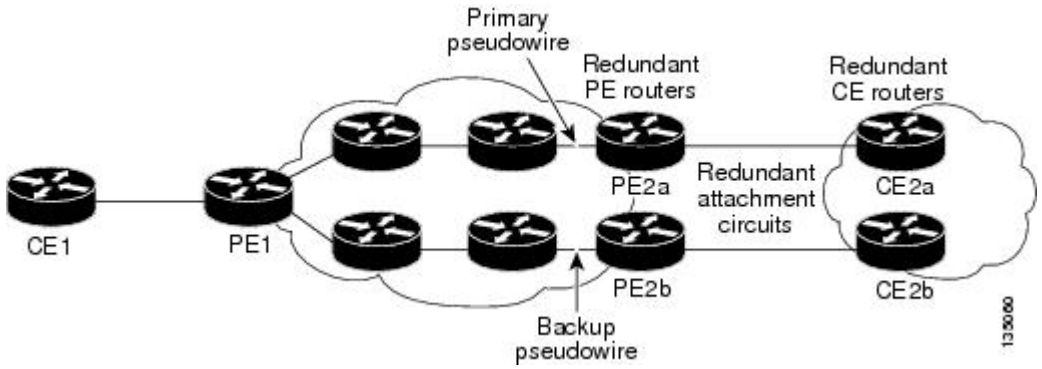
The figure below shows a network with redundant pseudowires, attachment circuits, and CE routers.

Figure 79: L2VPN Network with Redundant PWs, Attachment Circuits, and CE Routers



The figure below shows a network with redundant pseudowires, attachment circuits, CE routers, and PE routers.

Figure 80: L2VPN Network with Redundant PWs, Attachment Circuits, CE Routers, and PE Routers



How to Configure L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature enables you to configure a backup pseudowire in case the primary pseudowire fails. When the primary pseudowire fails, the PE router can switch to the backup pseudowire. You can have the primary pseudowire resume operation after it comes back up.

Configuring the Pseudowire

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a pseudowire, between the routers.

The pseudowire-class configuration group specifies the characteristics of the tunneling mechanism, which are:

- Encapsulation type
- Control protocol
- Payload-specific options

You must specify the **encapsulation mpls**command as part of the pseudowire class for the AToM VCs to work properly. If you omit the **encapsulation mpls**command as part of the **xconnect**command, you receive the following error:

```
% Incomplete command.
```

Perform this task to configure a pseudowire class.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** name
4. **encapsulation mpls**
5. **interworking** {**ethernet** | **ip**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** name **Example:** `Router(config)# pseudowire-class atom`	Establishes a pseudowire class with a name that you specify. Enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation. For AToM, the encapsulation type is **mpls**.
Step 5	**interworking** {**ethernet** | **ip**} **Example:** `Router(config-pw-class)# interworking ip`	(Optional) Enables the translation between the different Layer 2 encapsulations.

Configuring the Pseudowire using the commands associated with the L2VPN Protocol-Based CLIs feature

The successful transmission of the Layer 2 frames between PE routers is due to the configuration of the PE routers. You set up the connection, called a pseudowire, between the routers.

The pseudowire-class configuration group specifies the characteristics of the tunneling mechanism, which are:

- Encapsulation type
- Control protocol
- Payload-specific options

You must specify the **encapsulation mpls**command as part of the pseudowire class for the AToM VCs to work properly. If you omit the **encapsulation mpls**command as part of the **l2vpn xconnectcontext** command, you receive the following error:

```
% Incomplete command.
```

Perform this task to configure a pseudowire class.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pseudowire** *number*
4. **encapsulation mpls**
5. **neighbor** *peer-address vcid-value*
6. **interworking** {**ethernet** | **ip**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 1`	Establishes an interface pseudowire with a value that you specify. Enters pseudowire configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the tunneling encapsulation. For AToM, the encapsulation type is **mpls**.
Step 5	**neighbor** *peer-address vcid-value* **Example:** `Router(config-pw)#  neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 6	**interworking** {**ethernet**	**ip**} **Example:** `Router(config-pw)# interworking ip`	(Optional) Enables the translation between the different Layer 2 encapsulations.

Configuring L2VPN Pseudowire Redundancy

Perform this task to configure the L2VPN Pseudowire Redundancy feature.

Before you begin

For each transport type, the **xconnect**command is configured slightly differently. The following configuration steps use Ethernet VLAN over MPLS, which is configured in subinterface configuration mode. See *Any Transport over MPLS* to determine how to configure the **xconnect** command for other transport types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* **/** *subslot* **/** *interface* **.** *subinterface*
4. **encapsulation dot1q** *vlan-id*
5. **xconnect** *peer-router-id vcid* {**encapsulation mpls**| **pw-class** *pw-class-name}*
6. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*]
7. **backup delay** *e nable-delay* {*disable-delay* | **never**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* **/** *subslot* **/** *interface* **.** *subinterface* **Example:** `Router(config)# interface gigabitethernet0/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. **Note** Make sure that the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 4	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. **Note** The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet.
Step 5	**xconnect** *peer-router-id vcid* {**encapsulation mpls**| **pw-class** *pw-class-name}* **Example:** `Router(config-subif)# xconnect 10.0.0.1 123 pw-class atom`	Binds the attachment circuit to a pseudowire VC and enters xconnect configuration mode. • The syntax for this command is the same as for all other Layer 2 transports.

	Command or Action	Purpose
Step 6	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] **Example:** `Router(config-if-xconn)# backup peer 10.0.0.3 125 pw-class atom`	Specifies a redundant peer for the pseudowire VC. The pseudowire class name must match the name that you specified when you created the pseudowire class, but you can use a different pw-class in the **backup peer** command than the name that you used in the primary **xconnect** command.
Step 7	**backup delay** *e nable-delay* {*disable-delay* | **never**} **Example:** `Router(config-if-xconn)# backup delay 5 never`	Specifies how long (in seconds) the backup pseudowire VC should wait to take over after the primary pseudowire VC goes down. The range is from 0 to 180. Specifies how long the primary pseudowire should wait after it becomes active to take over for the backup pseudowire VC. The range is from 0 to 180 seconds. If you specify the **never** keyword, the primary pseudowire VC never takes over for the backup.

Configuring L2VPN Pseudowire Redundancy using the commands associated with the L2VPN Protocol-Based CLIs feature

Perform this task to configure the L2VPN Pseudowire Redundancy feature.

Before you begin

For each transport type, the **l2vpn xconnect context** command is configured slightly differently. The following configuration steps use Ethernet VLAN over MPLS, which is configured in subinterface configuration mode. See *Any Transport over MPLS* to determine how to configure the **l2vpn xconnect context** command for other transport types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *interface* **.** *subinterface*
4. **encapsulation dot1q** *vlan-id*
5. **end**
6. **interface pseudowire** *number*
7. **source template type pseudowire** *template-name*
8. **neighbor** *peer-address vcid-value*
9. **exit**
10. **l2vpn xconnect context** *context-name*
11. **member pseudowire** *interface-number*
12. **member pseudowire** *interface-number*
13. **member gigabitethernet** *interface-number*
14. **redundancy delay** *enable-delay*{*disable-delay* | **never**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot* / *subslot* / *interface* **.** *subinterface* **Example:** `Device(config)# interface gigabitethernet0/0/0.1`	Specifies the Gigabit Ethernet subinterface and enters subinterface configuration mode. Make sure that the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 4	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-subif)# encapsulation dot1q 100`	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers do not.
Step 5	**end** **Example:** `Router(config-subif)# end`	Exits to privileged EXEC mode.
Step 6	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 7	**source template type pseudowire** *template-name* **Example:** `Router(config-if)# source template type pseudowire atom`	Configures the source template of type pseudowire named atom
Step 8	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)#  neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 10	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 11	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 100 group GR_1 priority 2`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 12	**member pseudowire** *interface-number* **Example:** `Device(config-xconnect)# member pseudowire 1001 group GR_1 priority 2`	Specifies a second member pseudowire for redundancy.
Step 13	**member gigabitethernet** *interface-number* **Example:** `Device(config-xconnect)# member GigabitEthernet0/0/0.1 service instance 1`	Specifies the location of the Gigabit Ethernet member interface.
Step 14	**redundancy delay** *enable-delay*{*disable-delay* | **never**} **Example:** `Device(config-xconnect)# redundancy delay 0 0 group GR_1`	Specifies how long (in seconds) the backup pseudowire VC should wait to take over after the primary pseudowire VC goes down. The range is 0 to 180. Specifies how long the primary pseudowire should wait after it becomes active to take over for the backup pseudowire VC. The range is 0 to 180 seconds. If you specify the **never keyword**, the primary pseudowire VC never takes over for the backup.

Forcing a Manual Switchover to the Backup Pseudowire VC

To force the router switch over to the backup or primary pseudowire, you can enter the **xconnect backup force switchover** command in privileged EXEC mode. You can specify either the interface of the primary attachment circuit (AC) to switch to or the IP address and VC ID of the peer router.

A manual switchover can be made only if the interface or peer specified in the command is actually available and the xconnect moves to the fully active state when executing the command.

SUMMARY STEPS

1. **enable**
2. **xconnect backup force-switchover {** **interface** *interface-info* | **peer** *ip-address vcid}*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**xconnect backup force-switchover { interface** *interface-info* | **peer** *ip-address vcid}* **Example:** `Router# xconnect backup force-switchover peer 10.10.10.1 123`	Specifies that the router should switch to the backup or to the primary pseudowire.

Verifying the L2VPN Pseudowire Redundancy Configuration

Perform this task to verify that the L2VPN Pseudowire Redundancy feature is correctly configured.

SUMMARY STEPS

1. **show mpls l2transport vc**
2. **show xconnect all**
3. **xconnect logging redundancy**

DETAILED STEPS

Step 1 **show mpls l2transport vc**

The following is sample output from the **show mpls l2transport vc** command. In this example, the primary attachment circuit is up. The backup attachment circuit is available, but not currently selected.

Example:

```
Router# show mpls l2transport vc
Local intf     Local circuit           Dest address    VC ID      Status
-------------  ----------------------- --------------- ---------- ----------
Et0/0.1        Eth VLAN 101            10.0.0.2        101        UP
Et0/0.1        Eth VLAN 101            10.0.0.3        201        DOWN
Router# show mpls l2transport vc detail
Local interface: Et0/0.1 up, line protocol up, Eth VLAN 101 up
   Destination address 10.0.0.2 VC ID: 101, VC status UP
   .
   .
   .
Local interface: Et0/0.1 down, line protocol down, Eth VLAN 101 down
   Destination address 10.0.0.3 VC ID: 201, VC status down
   .
   .
   .
```

Step 2 **show xconnect all**

In this example, the topology is Attachment Circuit 1 to Pseudowire 1 with a Pseudowire 2 as a backup:

Example:

```
Router# show xconnect all
Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State
UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                          S2
------+---------------------------------+--+---------------------------------+--
UP pri ac   Et0/0(Ethernet)              UP mpls 10.55.55.2:1000               UP
IA sec ac   Et0/0(Ethernet)              UP mpls 10.55.55.3:1001               DN
```

In this example, the topology is Attachment Circuit 1 to Attachment Circuit 2 with a pseudowire backup for Attachment Circuit 2:

Example:

```
Router# show xconnect all
Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State
UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                          S2
------+---------------------------------+--+---------------------------------+--
UP pri ac   Se6/0:150(FR DLCI)           UP ac   Se8/0:150(FR DLCI)            UP
IA sec ac   Se6/0:150(FR DLCI)           UP mpls 10.55.55.3:7151               DN
```

Step 3 **xconnect logging redundancy**

In addition to the **show mpls l2transport vc** command and the **show xconnect** command, you can use the **xconnect logging redundancy** command to track the status of the xconnect redundancy group:

Example:

```
Router(config)# xconnect logging redundancy
```

When this command is configured, the following messages are displayed during switchover events:

Activating the primary member:

Example:

```
00:01:07: %XCONNECT-5-REDUNDANCY: Activating primary member 10.55.55.2:1000
```

Activating the backup member:

Example:

```
00:01:05: %XCONNECT-5-REDUNDANCY: Activating secondary member 10.55.55.3:1001
```

Verifying the L2VPN Pseudowire Redundancy Configuration using the commands associated with the L2VPN Protocol-Based CLIs feature

Use the following commands to verify that the L2VPN Pseudowire Redundancy feature is correctly configured.

SUMMARY STEPS

1. **show l2vpn atom vc**

2. **show l2vpn service all**
3. **logging redundancy**
4. **logging pseudowire status**

DETAILED STEPS

Step 1 show l2vpn atom vc

In this example, the primary attachment circuit is up. The backup attachment circuit is available, but not currently selected. The **show** output displays as follows:

Example:

```
Device# show l2vpn atom vc
Local intf     Local circuit           Dest address    VC ID      Status
-------------  ----------------------- --------------- ---------- ----------
Et0/0.1        Eth VLAN 101            10.0.0.2        101        UP
Et0/0.1        Eth VLAN 101            10.0.0.3        201        DOWN
Router# show l2vpn atom vc detail
Local interface: Et0/0.1 up, line protocol up, Eth VLAN 101 up
   Destination address 10.0.0.2 VC ID: 101, VC status UP
   .
   .
   .
Local interface: Et0/0.1 down, line protocol down, Eth VLAN 101 down
   Destination address 10.0.0.3 VC ID: 201, VC status down
   .
   .
   .
```

Step 2 show l2vpn service all

In this example, the topology is attachment circuit 1 to pseudowire 1 with apPseudowire 2 as a backup:

Example:

```
Device# show l2vpn service all
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down     IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering     NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                   Prio  St  XC St
  ---------          -----       -------------                   ----  --  -----
VPWS name: foo, State: UP
  Eth1/1.1                       Eth1/1.1:100(Eth VLAN)          0     UP  UP
  pw101              blue        102.1.1.1:100(MPLS)             2     UP  UP
  pw102              blue        103.1.1.1:100(MPLS)             5     SB  IA
  pw103              blue        104.1.1.1:100(MPLS)             8     SB  IA
  pw104              blue        105.1.1.1:100(MPLS)             11    SB  IA
```

In this example, the topology is attachment circuit 1 to attachment circuit 2 with a pseudowire backup for attachment circuit 2:

Example:

```
Device# show l2vpn service all
Legend: XC ST=Xconnect State, S1=Segment1 State, S2=Segment2 State
UP=Up, DN=Down, AD=Admin Down, IA=Inactive, NH=No Hardware
```

```
XC ST  Segment 1                        S1 Segment 2                        S2
------+---------------------------------+--+---------------------------------+--
UP pri ac   Se6/0:150(FR DLCI)          UP ac   Se8/0:150(FR DLCI)          UP
IA sec ac   Se6/0:150(FR DLCI)          UP mpls 10.55.55.3:7151             DN
```

Step 3 **logging redundancy**

In addition to the **show l2vpn atom vc** command and the **show l2vpn service** command, you can use the **logging redundancy** command to enable system message log (syslog) reporting of xconnect redundancy status events:

Example:

```
Device(config)# l2vpn
Device(config-l2vpn)# logging redundancy
```

When this command is configured, the messages below will be generated during switchover events:

Activating the primary member:

Example:

```
Device(config)# l2vpn
Device(config-l2vpn)# logging pseudowire status
```

When this command is configured, this is configured the status of the pseudowire can be monitored:

Activating the primary member:

Example:

```
00:01:07: %XCONNECT-5-REDUNDANCY: Activating primary member 10.55.55.2:1000
```

Activating the backup member:

Example:

```
00:01:05: %XCONNECT-5-REDUNDANCY: Activating secondary member 10.55.55.3:1001
```

Step 4 **logging pseudowire status**

you can use the **logging pseudowire status** command to monitor the status of the pseudowire.

Example:

```
Device(config)# l2vpn
Device(config-l2vpn)# logging pseudowire status
```

Configuration Examples for L2VPN Pseudowire Redundancy

Each of the configuration examples refers to one of the following pseudowire classes:

- AToM (like-to-like) pseudowire class:

```
pseudowire-class mpls
 encapsulation mpls
```

- L2VPN IP interworking:

```
pseudowire-class mpls-ip
 encapsulation mpls
 interworking ip
```

Example L2VPN Pseudowire Redundancy and AToM (Like to Like)

The following example shows a High-Level Data Link Control (HDLC) attachment circuit xconnect with a backup pseudowire:

```
interface Serial4/0
 xconnect 10.55.55.2 4000 pw-class mpls
 backup peer 10.55.55.3 4001 pw-class mpls
```

The following example shows a Frame Relay attachment circuit xconnect with a backup pseudowire:

```
connect fr-fr-pw Serial6/0 225 l2transport
 xconnect 10.55.55.2 5225 pw-class mpls
 backup peer 10.55.55.3 5226 pw-class mpls
```

Example L2VPN Pseudowire Redundancy and L2VPN Interworking

The following example shows an Ethernet attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet0/0
 xconnect 10.55.55.2 1000 pw-class mpls-ip
 backup peer 10.55.55.3 1001 pw-class mpls-ip
```

The following example shows an Ethernet VLAN attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet1/0.1
 encapsulation dot1Q 200
 no ip directed-broadcast
 xconnect 10.55.55.2 5200 pw-class mpls-ip
 backup peer 10.55.55.3 5201 pw-class mpls-ip
```

The following example shows a Frame Relay attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
connect fr-ppp-pw Serial6/0 250 l2transport
 xconnect 10.55.55.2 8250 pw-class mpls-ip
 backup peer 10.55.55.3 8251 pw-class mpls-ip
```

The following example shows a PPP attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Serial7/0
 encapsulation ppp
 xconnect 10.55.55.2 2175 pw-class mpls-ip
 backup peer 10.55.55.3 2176 pw-class mpls-ip
```

Example L2VPN Pseudowire Redundancy with Layer 2 Local Switching

The following example shows an Ethernet VLAN-VLAN local switching xconnect with a pseudowire backup for Ethernet segment E2/0.2. If the subinterface associated with E2/0.2 goes down, the backup pseudowire is activated:

```
connect vlan-vlan Ethernet1/0.2 Ethernet2/0.2
 backup peer 10.55.55.3 1101 pw-class mpls
```

The following example shows a Frame Relay-to-Frame Relay local switching connect with a pseudowire backup for Frame Relay segment S8/0 150. If data-link connection identifier (DLCI) 150 on S8/0 goes down, the backup pseudowire is activated:

```
connect fr-fr-ls Serial6/0 150 Serial8/0 150
 backup peer 10.55.55.3 7151 pw-class mpls
```

Example L2VPN Pseudowire Redundancy and Layer 2 Tunneling Protocol Version 3

The following example shows how to configure a backup peer for an xconnect session:

```
pseudowire-class 773
 encapsulation l2tpv3
 ip local interface GigabitEthernet0/0/0.773
!
pseudowire-class 774
 encapsulation l2tpv3
 ip local interface GigabitEthernet0/0/1.774
!
interface GigabitEthernet0/0/0.780
 encapsulation dot1Q 780
 xconnect 10.22.73.14 100 pw-class 773
  backup peer 10.22.74.14 101 pw-class 774
  backup delay 0 0
```

The following example shows how to configure a Gigabit Ethernet port with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/2
 xconnect 10.22.70.83 50 pw-class pe1-pw-primary
 backup peer 20.22.70.85 51 pw-class pe1-pw-secondary
```

The following example shows how to configure a Gigabit Ethernet VLAN with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.100
 encapsulation dot1q 100
 xconnect 10.22.70.83 60 pw-class pe1-pw-primary
 backup peer 10.22.70.85 61 pw-class pe1-pw-secondary
```

The following example shows how to configure a Gigabit Ethernet Q-in-Q with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.200
 encapsulation dot1q 200 second-dot1q 400
```

```
 xconnect 10.22.70.83 70 pw-class pe1-pw-primary
 backup peer 10.22.70.85 71 pw-class pe1-pw-secondary
```

The following example shows how to configure a Gigabit Ethernet Q-in-any with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.300
 encapsulation dot1q 300 second-dot1q any
 xconnect 10.22.70.83 80 pw-class pe1-pw-primary
 backup peer 10.22.70.85 81 pw-class pe1-pw-secondary
```

The following example shows how to configure an HDLC with L2VPN pseudowire redundancy and L2TPv3

```
interface Serial0/2/0:0
 no ip address
 xconnect 10.22.71.83 40 pw-class pe1-pw-hdlc
 backup peer 10.22.70.85 41 pw-class pe1-pw-hdlc-2
```

Configuration Examples for L2VPN Pseudowire Redundancy using the commands associated with the L2VPN Protocol-Based CLIs feature

Each of the configuration examples refers to one of the following interface pseudowires:

- AToM (like-to-like) interface pseudowire:

```
interface pseudowire 1
 encapsulation mpls
 neighbor 33.33.33.33 1
```

- L2VPN IP interworking:

```
interface pseudowire 1
 encapsulation mpls
 neighbor 33.33.33.33 1
 interworking ip
```

Example L2VPN Pseudowire Redundancy and AToM (Like to Like) using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows a High-Level Data Link Control (HDLC) attachment circuit xconnect with a backup pseudowire:

```
interface Serial4/0
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.55.55.3 4001
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
```

```
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
```

The following example shows a Frame Relay attachment circuit xconnect with a backup pseudowire:

```
connect fr-fr-pw Serial6/0 225 l2transport
interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.55.55.3 5226
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
```

Example L2VPN Pseudowire Redundancy and L2VPN Interworking using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows an Ethernet attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet0/0
interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowrire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows an Ethernet VLAN attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Ethernet1/0.1
 encapsulation dot1Q 200
 no ip directed-broadcast
 interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowrire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows a Frame Relay attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
connect fr-ppp-pw Serial6/0 250 l2transport
 interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowrire 100 group GR_1 priority 1
```

```
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows a PPP attachment circuit xconnect with L2VPN IP interworking and a backup pseudowire:

```
interface Serial7/0
 encapsulation ppp
 interface pseudowire 100
 source template type pseudowire ether-pw
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

Example L2VPN Pseudowire Redundancy and Layer 2 Tunneling Protocol Version 3 using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to configure a backup peer for an xconnect session:

```
interface pseudowire 773
 encapsulation l2tpv3
 ip local interface GigabitEthernet0/0/0.773
!
interface pseudowire 774
 encapsulation l2tpv3
 ip local interface GigabitEthernet0/0/1.774
!
interface GigabitEthernet0/0/0.780
 encapsulation dot1Q 780
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.73.14 100
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows how to configure a Gigabit Ethernet port with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/2
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.70.83 50
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
```

```
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows how to configure a Gigabit Ethernet VLAN with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.100
 encapsulation dot1q 100
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.70.83 60
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows how to configure a Gigabit Ethernet Q-in-Q with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.200
 encapsulation dot1q 200 second-dot1q 400
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.70.83 70
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows how to configure a Gigabit Ethernet Q-in-any with L2VPN pseudowire redundancy and L2TPv3:

```
interface GigabitEthernet0/0/0.300
 encapsulation dot1q 300 second-dot1q any
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.70.83 80
!
l2vpn xconnect context con1
 member pseudowire 100 group GR_1 priority 1
 member pseudowire 1001 group GR_1 priority 2
 member GigabitEthernet0/0/2 service-instance 1
 redundancy delay 0 0 group GR_1
 interworking ip
```

The following example shows how to configure an HDLC with L2VPN pseudowire redundancy and L2TPv3

```
interface Serial0/2/0:0
 no ip address
 interface pseudowire 100
 source template type pseudowire ether-pw
 neighbor 10.22.71.83 40
!
l2vpn xconnect context con1
```

```
l2vpn xconnect context con1
member pseudowire 100 group GR_1 priority 1
member pseudowire 1001 group GR_1 priority 2
member GigabitEthernet0/0/2 service-instance 1
redundancy delay 0 0 group GR_1
interworking ip
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Wide-area networking commands	*Cisco IOS Wide-Area Networking Command Reference*
Cisco IOS XE Multiprotocol Label Switching configuration tasks	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*
Cisco IOS XE Wide-area networking configuration tasks	*Cisco IOS XE Wide-Area Networking Configuration Guide*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

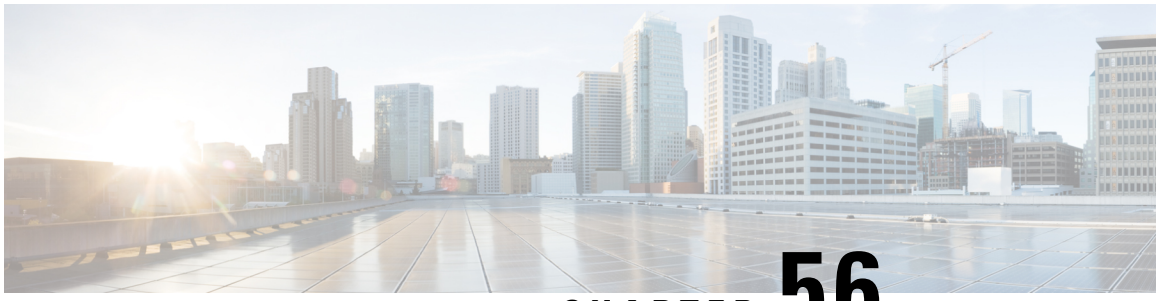
Feature Information for L2VPN Pseudowire Redundancy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 119: Feature Information for L2VPN Pseudowire Redundancy

Feature Name	Releases	Feature Information
L2VPN Pseudowire Redundancy	XE 2.3 XE 3.3S	This feature enables you to set up your network to detect a failure in the network and reroute the Layer 2 service to another endpoint that can continue to provide service. In Cisco IOS XE Release 2.3, this feature was integrated into the Cisco ASR 1000 Series Aggregation Service Routers. In Cisco IOS XE Release 3.3S, this feature supports Layer 2 Tunneling Protocol Version 3 (L2TPv3). The following commands were introduced or modified: **backup delay (L2VPN local switching), backup peer, show xconnect, xconnect backup force-switchover, xconnect logging redundancy.**
L2VPN Pseudowire Redundancies	Cisco IOS XE Fuji 16.9.1	In Cisco IOS XE Fuji 16.9.1, this feature is supported on Cisco 1000 Series ISRs.



CHAPTER 56

Pseudowire Group Switchover

The Pseudowire Group Switchover feature allows all pseudowires in a group to be quickly switched over to backup pseudowires. This group switchover is triggered by a single “group down” status message received from a remote peer.

Prerequisites for Pseudowire Group Switchover

-
- Label Distribution Protocol (LDP) must be implemented on the network.
- Each xconnect must have a backup pseudowire configured.

Restrictions for Pseudowire Group Switchover

The Pseudowire Group Switchover feature is supported on Cisco IOS XE Release 3.10S and later releases. This feature is supported on Cisco ASR 903 Series routers on the following attachment circuits:

- Ethernet VLAN
- Asynchronous Transfer Mode (ATM)
- Circuit Emulation over MPLS (CEM)

Information About Pseudowire Group Switchover

Introduction to Pseudowire Group Switchover

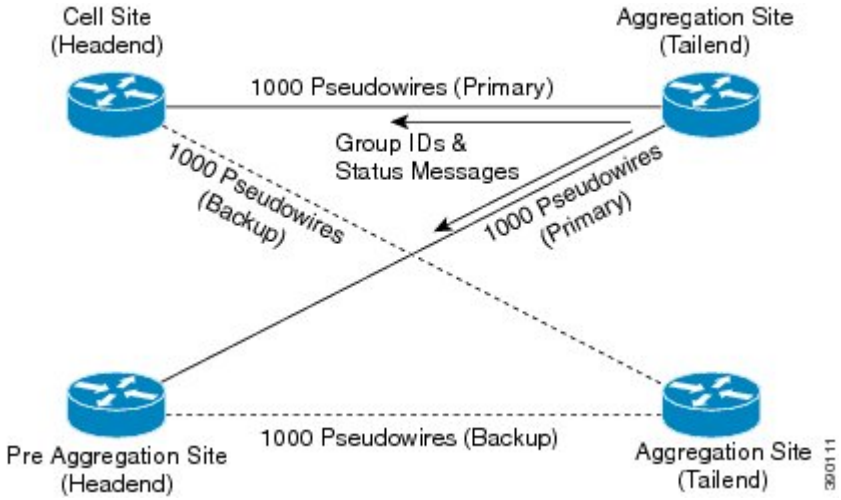
The Pseudowire Group Switchover feature allows you to reduce the switchover time from main pseudowires to backup pseudowires when a fault is encountered. The reduced switchover time is achieved by grouping Label Distribution Protocol (LDP) status messages and internal interprocess communication (IPC) messages.

When the remote peer detects an attachment circuit failure, it sends an LDP status message. When this status message is received, the designated backup pseudowires take over. Packets are then routed through the backup pseudowires.

Pseudowires can be grouped together by assigning a group ID. When an LDP status message is received by a pseudowire group, the entire group switches over, thus reducing switchover time.

Note The Pseudowire Group Switchover feature is enabled by default and cannot be disabled.

Figure 81: Primary and Backup Pseudowire Groups



How to Configure Predictive Switchover

Predictive switchover allows switchovers from a main pseudowire to a backup pseudowire with a remote "standby" status, without waiting for an “up” status from the remote peer.

Predictive switchover is configured by enabling redundancy predictive mode in global configuration mode or xconnect configuration mode.

Configuring Predictive Switchover (Global Configuration Mode)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn**
4. **redundancy predictive enabled**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn** **Example:** `Device(config)# l2vpn`	Enters l2vpn configuration mode.
Step 4	**redundancy predictive enabled** **Example:** `Device(config-l2vpn)# redundancy predictive enabled`	Enables redundancy predictive mode. • By default, redundancy predictive mode is disabled.
Step 5	**end** **Example:** `Device(config-l2vpn)# end`	Exits l2vpn configuration mode and returns to privileged EXEC mode.

Configuring Predictive Switchover (Xconnect Configuration Mode)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn xconnect context** *context-name*
4. **redundancy predictive enabled**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con1`	Creates an L2VPN cross-connect context and enters xconnect configuration mode.
Step 4	**redundancy predictive enabled** **Example:** `Device(config-xconnect)# redundancy predictive enabled`	Enables redundancy predictive mode.
Step 5	**end** **Example:** `Device(config-xconnect)# end`	Exits xconnect configuration mode and returns to privileged EXEC mode.

Verifying a Pseudowire Group Switchover Configuration

You can use **show** commands to view information about a pseudowire group switchover configuration.

The following example shows how to display information about Any Transport over MPLS (AToM) virtual circuits (VCs):

```
Device# show l2vpn atom vc destination 2.1.1.2 group remote 6

                                      Service
Interface  Dest Address     VC ID     Type     Name                    Status
---------  ---------------  ---------- ------ ------------------------ ----------
pw100001   2.1.1.2          1234000    p2p      Et1/0.1-1001           UP
```

The following example shows how to display the status of the pseudowire switching point:

```
Device# show l2vpn atom vc destination 2.1.1.2 group remote 6 detail

pseudowire100001 is up, VC status is up PW type: Ethernet
  Create time: 5d20h, last status change time: 5d20h
    Last label FSM state change time: 5d20h
  Destination address: 2.1.1.2 VC ID: 1234000
    Output interface: Et0/0, imposed label stack {2001}
  Preferred path: not configured
  Default path: active
  Next hop: 20.0.0.2
Member of xconnect service Et1/0.1-1001, group right
  Associated member Et1/0.1 is up, status is up
  Interworking type is Ethernet
```

```
  Service id: 0x6d000002
Signaling protocol: LDP, peer 2.1.1.2:0 up
  Targeted Hello: 10.1.1.1(LDP Id) -> 2.1.1.2, LDP is UP
  Graceful restart: not configured and not enabled
  Non stop routing: not configured and not enabled
  PWid FEC (128), VC ID: 1234000
  Status TLV support (local/remote) : enabled/supported
    LDP route watch : enabled
    Label/status state machine : established, LruRru
    Local dataplane status received : No fault
    BFD dataplane status received : Not sent
    BFD peer monitor status received : No fault
    Status received from access circuit : No fault
    Status sent to access circuit : No fault
    Status received from pseudowire i/f : No fault
    Status sent to network peer : No fault
    Status received from network peer : No fault
    Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        2007                           2001
Group ID     0                              6
Interface
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x12                           0x12
               LSPV [2], BFD/Raw [5]          LSPV [2], BFD/Raw [5]
VCCV CC type 0x07                           0x07
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
Dataplane:
  SSM segment/switch IDs: 12309/4115 (used), PWID: 1
Rx Counters
  106563 input transit packets, 9803650 bytes
  0 drops, 0 seq err
Tx Counters
  0 output transit packets, 0 bytes
  0 drops
```

The following example lists the active and standby segment pairs associated with each peer IP address and group identifier:

```
Device# show ssm group

Active              Standby
IP Address       Group ID       Segment/Switch      Segment/Switch
=============================================================================
2.1.1.2              6            8215/4115           4116/8210
```

The following example displays the number of active and standby segment pairs associated with each peer IP address and group identifier:

```
Device# show ssm group 2.1.1.2 6 summary

IP Address        Group ID      Group Members
=============================================
2.1.1.2               6              1
```

The following example displays the number of pseudowires programmed in the hardware, with grouping information:

```
Device# show platform hardware pp active pw eompls group brief

Brief L2VPN EoMPLS Pseudo Wire Group Info

IP address           Group ID            Count
-----------------------------------------------------
0x47474747           100695488            90
```

Troubleshooting a Pseudowire Group Switchover Configuration

Use the **debug platform software atom brief** command to view information about the following configurations:

- Add Group
- Delete From Group
- Group Switchovers

Note We recommend that you use the **debug platform software atom brief** command only under Cisco Technical Assistance Center (TAC) supervision.

Configuration Examples for Predictive Switchover

Example: Configuring Predictive Switchover (Global Configuration Mode)

```
Device> enable
Device# configure terminal
Device(config)# l2vpn
Device(config-l2vpn)# redundancy predictive enabled
Device(config-l2vpn)# end
```

Example: Configuring Predictive Switchover (Xconnect Configuration Mode)

```
Device> enable
Device# configure terminal
Device(config)# l2vpn xconnect context con1
Device(config-xconnect)# redundancy predictive enabled
Device(config-xconnect)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4447	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

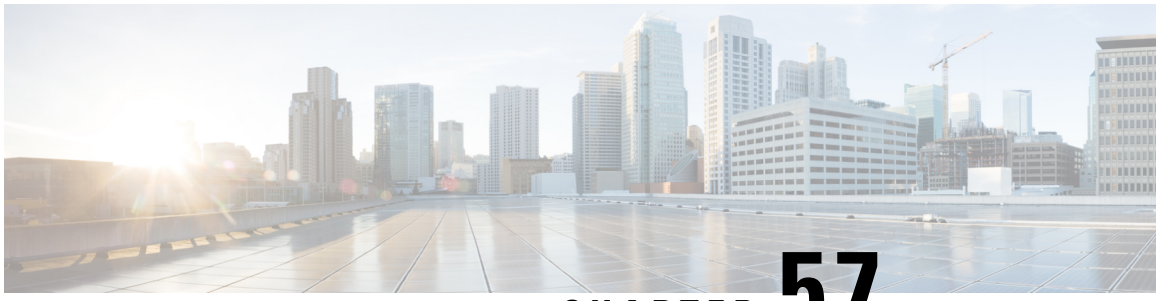
Feature Information for Pseudowire Group Switchover

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 120: Feature Information for Pseudowire Group Switchover

Feature Name	Releases	Feature Information
Pseudowire Group Switchover	Cisco IOS XE Release 3.10S	This feature allows all pseudowires in a group to be quickly switched over to backup pseudowires. This group switchover is triggered by a single “group down” status message received from a remote peer. The following commands were introduced or modified: **redundancy predictive**, **show ssm group**.



CHAPTER 57

L2VPN Pseudowire Switching

This feature module explains how to configure L2VPN Pseudowire Switching, which extends layer 2 virtual private network (L2VPN) pseudowires across an interautonomous system (inter-AS) boundary or across two separate multiprotocol label switching (MPLS) networks.

Restrictions for L2VPN Pseudowire Switching

- In Cisco IOS XE Release 2.4, Pseudowire Switching is supported on Ethernet over MPLS attachment circuits.
- L2VPN Pseudowire Switching is supported with AToM.
- Only static, on-box provisioning is supported.
- Sequencing numbers in AToM packets are not processed by L2VPN Pseudowire Switching. The feature blindly passes the sequencing data through the xconnect packet paths, a process that is called transparent sequencing. The endpoint PE-CE connections enforce the sequencing.
- You can ping the adjacent next-hop PE router. End-to-end LSP pings are not supported.
- Do not configure IP or Ethernet interworking on a router where L2VPN Pseudowire Switching is enabled. Instead, configure interworking on the routers at the edge PEs of the network.
- The control word negotiation results must match. If either segment does not negotiate the control word, the control word is disabled for both segments.
- AToM Graceful Restart is negotiated independently on each pseudowire segment. If there is a transient loss of the LDP session between two AToM PE routers, packets continue to flow.
- Per-pseudowire quality of service (QoS) is not supported. Traffic Engineering (TE) tunnel selection is supported.

- Attachment circuit interworking is not supported.

Information About L2VPN Pseudowire Switching

How L2VPN Pseudowire Switching Works

L2VPN Pseudowire Switching allows the user to extend L2VPN pseudowires across an inter-AS boundary or across two separate MPLS networks, as shown in the figures below. L2VPN Pseudowire Switching connects two or more contiguous pseudowire segments to form an end-to-end multihop pseudowire. This end-to-end pseudowire functions as a single point-to-point pseudowire.

As shown in the second figure below, L2VPN Pseudowire Switching enables you to keep the IP addresses of the edge PE routers private across inter-AS boundaries. You can use the IP address of the autonomous system boundary routers (ASBRs) and treat them as pseudowire aggregation (PE-agg) routers. The ASBRs join the pseudowires of the two domains.

L2VPN Pseudowire Switching also enables you to keep different administrative or provisioning domains to manage the end-to-end service. At the boundaries of these networks, PE-agg routers delineate the management responsibilities.

Figure 82: L2VPN Pseudowire Switching in an Intra-AS Topology

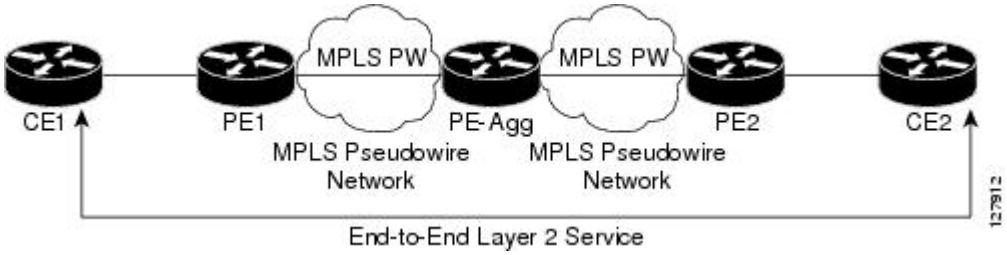
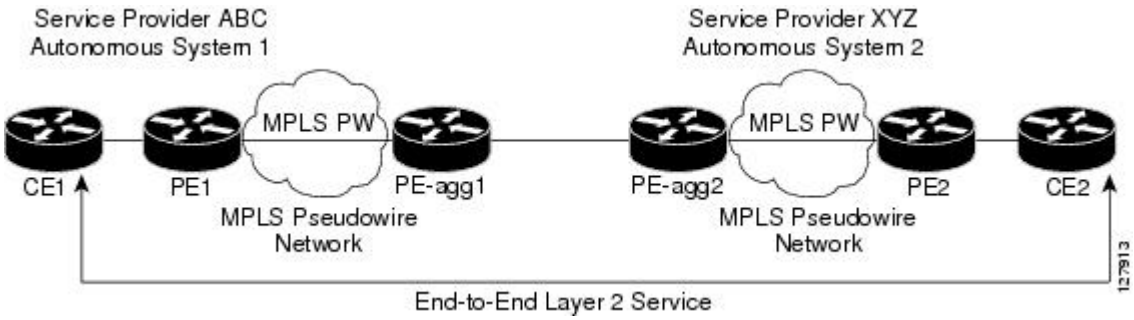


Figure 83: L2VPN Pseudowire Switching in an Inter-AS Topology



How Packets Are Manipulated at the Aggregation Point

Switching AToM packets between two AToM pseudowires is the same as switching any MPLS packet. The MPLS switching data path switches AToM packets between two AToM pseudowires. The following list explains exceptions:

- The outgoing virtual circuit (VC) label replaces the incoming VC label in the packet. New Internal Gateway Protocol (IGP) labels and Layer 2 encapsulation are added.
- The incoming VC label time-to-live (TTL) field is decremented by one and copied to the outgoing VC label TTL field.
- The incoming VC label EXP value is copied to the outgoing VC label EXP field.
- The outgoing VC label 'Bottom of Stack' S bit in the outgoing VC label is set to1.
- AToM control word processing is not performed at the L2VPN Pseudowire Switching aggregation point. Sequence numbers are not validated. Use the Router Alert label for LSP Ping; do not require control word inspection to determine an LSP Ping packet.

How to Configure L2VPN Pseudowire Switching

Configuring

Use the following procedure to configure L2VPN Pseudowire Switching on each of the PE-agg routers.

Before you begin

- This procedure assumes that you have configured basic AToM L2VPNs. This procedure does not explain how to configure basic AToM L2VPNs that transport Layer 2 packets over an MPLS backbone. For information on the basic configuration, see Any Transport over MPLS.
- For inter-Autonomous configurations, ASBRs require a labeled interface.

Note In this configuration, you are limited to two **neighbor**commands after entering the **l2 vfi**command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **point-to-point**
4. **neighbor** *ip-address vcid* **encapsulation mpls** | **pw-class** *pw-class-name*
5. **exit**
6. **exit**
7. **show mpls l2transport vc** [**vcid** [*vc-id* | [*vc-id-min vc-id-max*]] [**interface** *name*[*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**]
8. **show vfi** [*vfi-name*]
9. **ping** [*protocol*] [**tag**] {*host-name*| *system-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *name* **point-to-point** **Example:** `Router(config)# l2 vfi atomtunnel point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 4	**neighbor** *ip-address vcid* **encapsulation mpls** | **pw-class** *pw-class-name* **Example:** `Router(config-vfi)# neighbor 10.0.0.1 100 pw-class mpls`	Sets up an emulated VC. Specify the IP address and the VC ID of the remote router. Also specify the pseudowire class to use for the emulated VC. **Note** Only two **neighbor**commands are allowed for each **l2 vfi point-to-point** command.
Step 5	**exit** **Example:** `Router(config-vfi)# exit`	Exits VFI configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**show mpls l2transport vc** [**vcid** [*vc-id* | [*vc-id-min vc-id-max*]] [**interface** *name*[*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**] **Example:** `Router# show mpls l2transport vc`	Verifies that the L2VPN Pseudowire Switching session has been established.
Step 8	**show vfi** [*vfi-name*] **Example:** `Router# show vfi atomtunnel`	Verifies that a point-to-point VFI has been established.
Step 9	**ping** [*protocol*] [**tag**] {*host-name*| *system-address*} **Example:**	When issued from the CE routers, this command verifies end-to-end connectivity.

	Command or Action	Purpose
	`Router# ping 10.1.1.1`	

Examples

The following example displays the output of the **show mpls l2transport vc** command:

```
Router# show mpls l2transport vc
Local intf     Local circuit              Dest address    VC ID Status
-------------  -------------------------- --------------- ----- ----
MPLS PW        10.0.1.1:100               10.0.1.1        100   UP
MPLS PW        10.0.1.1:100               10.0.1.1        100   UP
```

The following example displays the output of the **show vfi**command:

```
Router# show vfi
VFI name: test, type: point-to-point
 Neighbors connected via pseudowires:
   Router ID        Pseudowire ID
   10.0.1.1         100
   10.0.1.1         100
```

How to Configure L2VPN Pseudowire Switching using the commands associated with the L2VPN Protocol-Based CLIs feature

Perform this task to configure L2VPN Pseudowire Switching on each of the PE-agg routers. In this configuration, you are limited to two **neighbor** commands after entering the **l2vpn xconnect** command.

Before you begin

- This task assumes that you have configured basic AToM L2VPNs. This task does not explain how to configure basic AToM L2VPNs that transport Layer 2 packets over an MPLS backbone. For information on the basic configuration, see the “Any Transport over MPLS” section.
- For interautonomous configurations, autonomous system boundary routers (ASBRs) require a labeled interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pseudowire** *number*
4. **encapsulation mpls**
5. **neighbor** *peer-address vcid-value*
6. **exit**

7. **interface pseudowire** *number*
8. **encapsulation mpls**
9. **neighbor** *peer-address vcid-value*
10. **exit**
11. **l2vpn xconnect context** *context-name*
12. **member pseudowire** *interface-number*
13. **member** *ip-address vcid* **encapsulation mpls**
14. **member pseudowire** *interface-number*
15. **member** *ip-address vcid* **encapsulation mpls**
16. **exit**
17. **exit**
18. **show l2vpn atom vc** [**vcid** [*vc-id* | *vc-id-min vc-id-max*]] [**interface** *type number* [*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**]
19. **ping** [*protocol*] [**tag**] {*hostname*| *system-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 5	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 7	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 200`	Specifies the pseudowire interface and enters interface configuration mode.
Step 8	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 9	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.0.0.2 124`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 11	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 12	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 13	**member** *ip-address vcid* **encapsulation mpls** **Example:** `Device(config-xconnect)# member 10.0.0.1 123 encapsulation mpls`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection. **Note** Only two **member** commands are allowed for each **l2vpn xconnect context** command.
Step 14	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 200`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 15	**member** *ip-address vcid* **encapsulation mpls** **Example:** `Device(config-xconnect)# member 10.0.0.2 124 encapsulation mpls`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection. **Note** Only two **member** commands are allowed for each **l2vpn xconnect context** command.

	Command or Action	Purpose
Step 16	**exit** **Example:** `Device(config-xconnect)# exit`	Exits Xconnect configuration mode.
Step 17	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 18	**show l2vpn atom vc** [**vcid** [*vc-id* | *vc-id-min vc-id-max*]] [**interface** *type number* [*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**] **Example:** `Device# show l2vpn atom vc`	Displays information about Any Transport over MPLS (AToM) virtual circuits (VCs) and static pseudowires that have been enabled to route Layer 2 packets on a device.
Step 19	**ping** [*protocol*] [**tag**] {*hostname*| *system-address*} **Example:** `Device# ping 10.1.1.1`	When issued from the CE routers, verifies end-to-end connectivity.

Configuring

Use the following procedure to configure L2VPN Pseudowire Switching on each of the PE-agg routers.

Before you begin

- This procedure assumes that you have configured basic AToM L2VPNs. This procedure does not explain how to configure basic AToM L2VPNs that transport Layer 2 packets over an MPLS backbone. For information on the basic configuration, see Any Transport over MPLS.
- For inter-Autonomous configurations, ASBRs require a labeled interface.

Note In this configuration, you are limited to two **neighbor**commands after entering the **l2 vfi**command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **point-to-point**
4. **neighbor** *ip-address vcid* **encapsulation mpls** | **pw-class** *pw-class-name*
5. **exit**

6. **exit**
7. **show mpls l2transport vc** [**vcid** [*vc-id* | [*vc-id-min vc-id-max*]] [**interface** *name*[*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**]
8. **show vfi** [*vfi-name*]
9. **ping** [*protocol*] [**tag**] {*host-name*| *system-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *name* **point-to-point** **Example:** `Router(config)# l2 vfi atomtunnel point-to-point`	Creates a point-to-point Layer 2 virtual forwarding interface (VFI) and enters VFI configuration mode.
Step 4	**neighbor** *ip-address vcid* **encapsulation mpls** | **pw-class** *pw-class-name* **Example:** `Router(config-vfi)# neighbor 10.0.0.1 100 pw-class mpls`	Sets up an emulated VC. Specify the IP address and the VC ID of the remote router. Also specify the pseudowire class to use for the emulated VC. **Note** Only two **neighbor**commands are allowed for each **l2 vfi point-to-point** command.
Step 5	**exit** **Example:** `Router(config-vfi)# exit`	Exits VFI configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**show mpls l2transport vc** [**vcid** [*vc-id* | [*vc-id-min vc-id-max*]] [**interface** *name*[*local-circuit-id*]] [**destination** *ip-address* | *name*] [**detail**] **Example:** `Router# show mpls l2transport vc`	Verifies that the L2VPN Pseudowire Switching session has been established.

	Command or Action	Purpose
Step 8	**show vfi** [*vfi-name*] **Example:** `Router# show vfi atomtunnel`	Verifies that a point-to-point VFI has been established.
Step 9	**ping** [*protocol*] [**tag**] {*host-name*| *system-address*} **Example:** `Router# ping 10.1.1.1`	When issued from the CE routers, this command verifies end-to-end connectivity.

Examples

The following example displays the output of the **show mpls l2transport vc** command:

```
Router# show mpls l2transport vc
Local intf     Local circuit              Dest address    VC ID Status
-------------  -------------------------- --------------- ----- ----
MPLS PW        10.0.1.1:100               10.0.1.1        100   UP
MPLS PW        10.0.1.1:100               10.0.1.1        100   UP
```

The following example displays the output of the **show vfi**command:

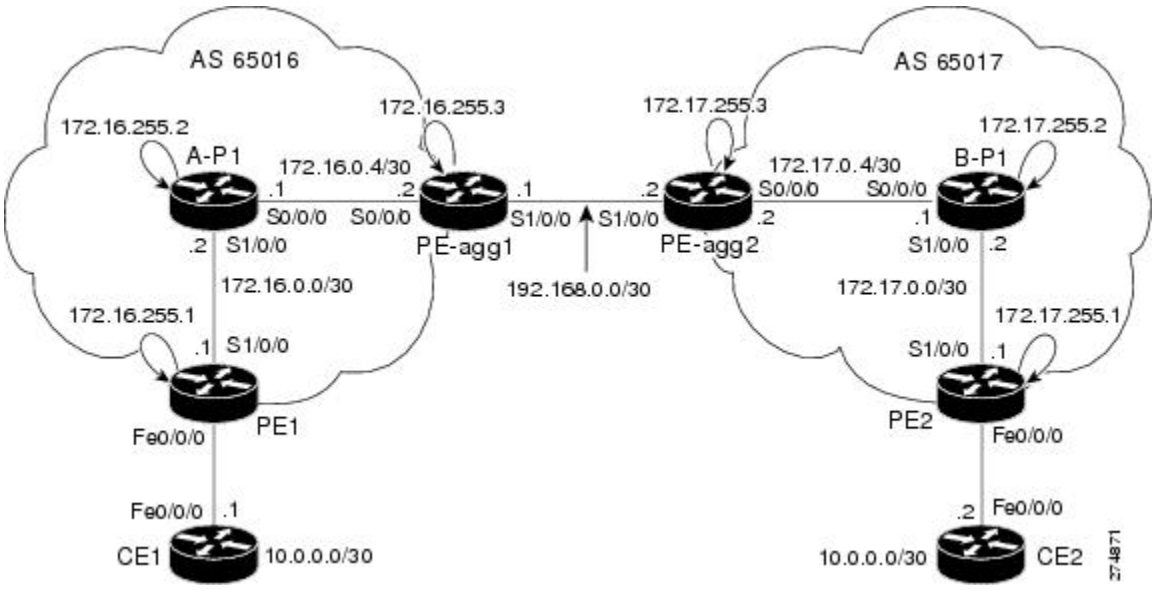
```
Router# show vfi
VFI name: test, type: point-to-point
 Neighbors connected via pseudowires:
   Router ID        Pseudowire ID
   10.0.1.1         100
   10.0.1.1         100
```

Configuration Examples for L2VPN Pseudowire Switching

L2VPN Pseudowire Switching in an Inter-AS Configuration Example

Two separate autonomous systems are able to pass L2VPN packets, because the two PE-agg routers have been configured with L2VPN Pseudowire Switching. This example configuration is shown in the figure below.

Figure 84: L2VPN Pseudowire Switching in an InterAutonomous System



CE1	CE2

CE1	CE2
version 12.0	version 12.0
service timestamps debug uptime	service timestamps debug uptime
service timestamps log uptime	service timestamps log uptime
service password-encryption	service password-encryption
!	!
hostname [ce1]	hostname [ce2]
!	!
boot-start-marker	boot-start-marker
boot-end-marker	boot-end-marker
!	!
enable secret 5 1o9N6$LSrxHufTn0vjCY0nW8hQX.	enable secret 5 1YHo6$LQ4z5PdrF5B9dnL75Xvvm1
!	!
ip subnet-zero	ip subnet-zero
ip cef	ip cef
no ip domain-lookup	no ip domain-lookup
!	!
interface FastEthernet0/0/0	interface FastEthernet0/0/0
ip address 10.0.0.1 255.255.255.252	ip address 10.0.0.2 255.255.255.252
no ip directed-broadcast	no ip directed-broadcast
!	!
ip classless	ip classless
!	!
control-plane	control-plane
!	!

CE1	CE2
line con 0 exec-timeout 0 0 line aux 0 line vty 0 4 login ! no cns aaa enable end	line con 0 exec-timeout 0 0 line aux 0 line vty 0 4 login ! no cns aaa enable end

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
L2VPN pseudowire redundancy	"L2VPN Pseudowire Redundancy" feature module in the *MPLS Layer 2 VPNs Configuration Guide*.
H-VPLS	" Configuring VPLS " in the "Configuring Multiprotocol Label Switching on the Optical Services Modules" chapter in the *Optical Services Modules Installation and Configuration Notes*, 12.2SR document.
MPLS traffic engineering	"MPLS Traffic Engineering Fast Reroute Link and Node Protection" feature module in the *MPLS Traffic Engineering: Path, Link, and Node Protection Configuration Guide* (part of the Multiprotocol Label Switching Configuration Guide Library)

Standards

Standard	Title
http://www.ietf.org/rfc/rfc4447.txt	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

Standard	Title
http://www3.ietf.org/proceedings/06mar/IDs/draft-ietf-l2vpn-vpls-ldp-08.txt	*Virtual Private LAN Services over MPLS*
http://www.ietf.org/internet-drafts/draft-ietf-pwe3-segmented-pw-02.txt	*Segmented Pseudo Wire*
draft-ietf-pwe3-vccv-10.txt	*Pseudo Wire Virtual Circuit Connectivity Verification (VCCV)*
draft-ietf-pwe3-oam-msg-map-03.txt	*Pseudo Wire (PW) OAM Message Mapping*

MIBs

MIB	MIBs Link
Pseudowire Emulation Edge-to-Edge MIBs for Ethernet, Frame Relay, and ATM Services	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

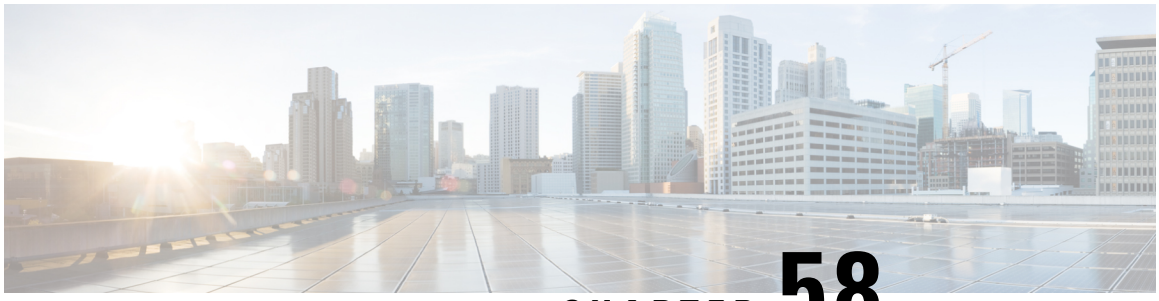
Feature Information for L2VPN Pseudowire Switching

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 121: Feature Information for L2VPN Pseudowire Switching

Feature Name	Releases	Feature Information
L2VPN Pseudowire Switching	Cisco IOS XE Release 2.4	The L2VPN Pseudowire Switching feature extends layer 2 virtual private network (L2VPN) pseudowires across an interautonomous system (inter-AS) boundary or across two separate multiprotocol label switching (MPLS) networks. In Cisco IOS XE Release 2.4, the L2VPN Pseudowire Switching feature is supported with Ethernet over MPLS. The following commands were introduced or modified: **l2 vfi point-to-point**, **neighbor**(L2VPN Pseudowire Switching), **show vfi**.
L2VPN Pseudowire-Switching	Cisco IOS XE Fuji 16.9.1	In Cisco IOS XE Fuji 16.9.1, the L2VPN Pseudowire Switching feature is supported on Cisco 1000 Series ISRs.



CHAPTER 58

Xconnect as a Client of BFD

The Xconnect as a Client of Bidirectional Forwarding Detection (BFD) feature provides a trigger for redundant pseudowire switchover based on BFD's fast failure detection capabilities.

Information About Xconnect as a Client of BFD

Xconnect as a Client of BFD

Redundant pseudowires are deployed to provide fault tolerance and resiliency to L2VPN-backhauled connections. The speed at which a system recovers from failures, especially when scaled to large numbers of pseudowires, is critical to many service providers and service level agreements (SLAs). The configuration of a trigger for redundant pseudowire switchover reduces the time that it takes a large number of pseudowires to failover. A fundamental component of bidirectional forwarding detection (BFD) capability is enabled by fast-failure detection (FFD).

The configuration of this feature refers to a BFD configuration, such as the following (the second URL in the **bfd map** command is the loopback URL in the **monitor peer bfd** command):

```
bfd-template multi-hop mh
 interval min-tx 200 min-rx 200 multiplier 3 !
bfd map ipv4  10.1.1.0/24  10.1.1.1/32 mh
```

How to Configure Xconnect as a Client of BFD

Configuring Xconnect as a Client of BFD

Perform this task to configure a trigger for redundant pseudowire switchover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class mpls-ffd**
   - Enters pseudowire class configuration mode.
4. **encapsulation mpls**
5. **monitor peer bfd** [**local interface** *interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class mpls-ffd** • Enters pseudowire class configuration mode. **Example:** `Device(config)# pseudowire-class mpls-ffd`	Establishes a pseudowire class for MPLS fast-failure detection.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation to be MPLS.
Step 5	**monitor peer bfd** [**local interface** *interface-type interface-number*] **Example:** `Device(config-pw-class)# monitor peer bfd local interface loopback 0`	Enables the pseudowire fast-failure detection capability.

Configuration Examples for Xconnect as a Client of BFD

Example: Xconnect as a Client of BFD

Pseudowire Class Configuration

The following example shows pseudowire fast-failure detection enabled for a pseudowire class:

```
pseudowire-class mpls-ffd
 encapsulation mpls
 monitor peer bfd local interface Loopback0
```

Template Configuration

The following example shows pseudowire fast-failure detection enabled in a template:

```
template type pseudowire 1
 encapsulation mpls
 monitor peer bfd local interface Ethernet0/1
```

Interface Configuration

The following example shows pseudowire fast-failure detection enabled for an interface:

```
interface pseudowire100
 encapsulation mpls
 neighbor 10.10.1.1 21190
 monitor peer bfd local interface Ethernet0/1
```

Additional References

Related Documents

Related Topic	Document Title
Any Transport over MPLS	Any Transport over MPLS
High Availability for AToM	AToM Graceful Restart
L2VPN Interworking	L2VPN Interworking
Layer 2 local switching	Layer 2 Local Switching
PWE3 MIB	Pseudowire Emulation Edge-to-Edge MIBs for Ethernet and Frame Relay Services
Packet sequencing	Any Transport over MPLS (AToM) Sequencing Support

Related Topic	Document Title
BFD configuration	IP Routing BFD Configuration Guide

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

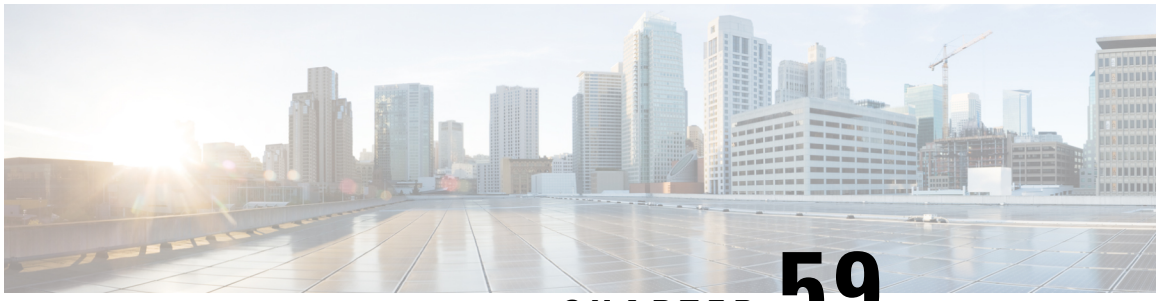
Feature Information for Xconnect as a Client of BFD

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 122: Feature Information for Xconnect as a Client of BFD

Feature Name	Releases	Feature Information
Xconnect as a Client of BFD	Cisco IOS XE Release 3.8S	This feature provides fast-failure detection for L2VPN pseudowire redundancy. The following command was introduced: **monitor peer bfd**.



CHAPTER 59

H-VPLS N-PE Redundancy for QinQ Access

The H-VPLS N-PE Redundancy for QinQ Access feature enables two network provider edge (N-PE) devices to provide failover services to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Having redundant N-PE devices provides improved stability and reliability against link and node failures.

Prerequisites for H-VPLS N-PE Redundancy for QinQ Access

- Before configuring this feature, configure your hierarchical virtual private LAN service (H-VPLS) network and make sure it is operating correctly.
- Make sure that the PE-to-customer edge (CE) interface is configured with a list of allowed VLANs.
- To provide faster convergence, you can enable the MPLS Traffic Engineering—Fast Reroute feature in the Multiprotocol Label Switching (MPLS) core.
- Enable the L2VPN Pseudowire Redundancy feature on the user provider edge (U-PE) devices for MPLS access.
- When configuring Multiple Spanning Tree Protocol (MSTP), specify that one of the network provider edge (N-PE) devices is the root by assigning it the lowest priority using the **spanning-tree mst** *instance-id* **priority** *priority* command.
- When configuring MSTP, make sure that each device participating in the spanning tree is in the same region and is the same revision by issuing the **revision**, **name**, and **instance** commands in MST configuration mode.

Restrictions for H-VPLS N-PE Redundancy for QinQ Access

- This feature cannot be used with the VPLS Autodiscovery feature on pseudowires that attach to network provider edge (N-PE) devices. When you create the virtual private LAN service (VPLS), you can manually create the virtual forwarding instance (VFI).
- You cannot configure more than one pseudowire to carry the bridge protocol data unit (BPDU) packets between two redundant network provider edge (N-PE) devices on the same Virtual Private LAN service (VPLS) site.
- You cannot configure a local loopback address as a neighbor when you configure the H-VPLS N-PE Redundancy feature on N-PE devices. If you do so, the following error message is displayed:

```
VPLS local switching to peer address not supported
```

- Only two N-PE devices can be connected to each U-PE device.
- The spanning-tree mode must be Multiple Spanning Tree Protocol (MSTP) for the H-VPLS N-PE Redundancy feature. If the spanning-tree mode changes, the H-VPLS N-PE Redundancy feature might not work correctly, even though the pseudowire that carries the BPDU packet still exists and the H-VPLS N-PE Redundancy feature is still configured.

Information About H-VPLS N-PE Redundancy for QinQ Access

How H-VPLS N-PE Redundancy for QinQ Access Works

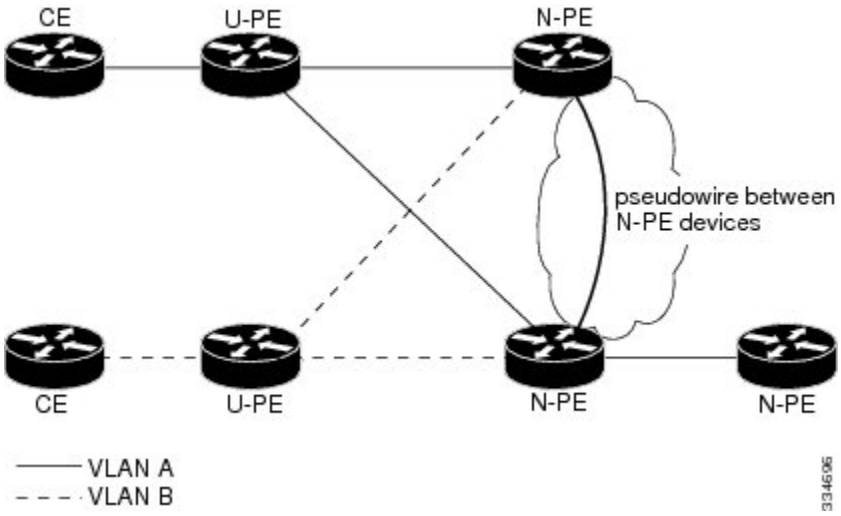
In a network configured with the H-VPLS N-PE Redundancy feature, the user provider edge (U-PE) device is connected to two network provider edge (N-PE) devices. This feature provides a level of redundancy that can tolerate both link and device faults. If a failure occurs in the network that disables one N-PE device from transmitting data, the other N-PE device takes over. This feature works with both QinQ access based on Multiple Spanning Tree Protocol (MSTP) and Multiprotocol Label Switching (MPLS) access based on pseudowire redundancy.

H-VPLS N-PE Redundancy with QinQ Access Based on MSTP

The H-VPLS N-PE Redundancy with QinQ Access feature uses the Multiple Spanning Tree Protocol (MSTP) running on the network provider edge (N-PE) devices and user provider edge (U-PE) devices in a hierarchical Virtual Private LAN service (H-VPLS) network. A pseudowire running between N-PE devices carries only MSTP bridge protocol data units (BPDUs). The pseudowire running between the N-PE devices is always up and is used to create a loop path between N-PE devices so that MSTP blocks one of the redundant paths between the U-PE device and the N-PE devices. If the primary N-PE device or the path to it fails, MSTP enables the path to the backup N-PE device.

The figure below shows an H-VPLS network with redundant access. Each U-PE device has two connections, one to each N-PE device. Between the two N-PE devices is a pseudowire to provide a loop path for MSTP BPDUs. The network topology allows for the backup N-PE device to take over if the primary N-PE device or the path to it fails.

Figure 85: H-VPLS N-PE Redundancy with QinQ Access Based on MSTP



How to Configure H-VPLS N-PE Redundancy for QinQ Access

Configuring the VPLS Pseudowire Between the N-PE Devices using the commands associated with the L2VPN Protocol-Based CLIs feature

Configuring network provider edge (N-PE) redundancy in a hierarchical Virtual Private LAN service (H-VPLS) network requires that you configure the VPLS pseudowire for transporting bridge protocol data unit (BPDU) packets. For the core pseudowire between the N-PE devices, you configure a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) and attach the VFI to a bridge-domain (described here). Then, in the next task, you bind the service instance to the bridge-domain. This configuration provides a redundancy that provides improved reliability against link and node failures.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *vpn id*
5. **member** *ip-address* **encapsulation mpls**
6. **forward permit l2protocol all**
7. **exit**
8. **bridge-domain** *bridge-id*
9. **member vfi** *vfi-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context VPLS-10`	Establishes a L2VPN VFI between two or more separate networks, and enters L2VFI configuration mode.
Step 4	**vpn id** *vpn id* **Example:** `Device(config-vfi)# vpn id 10`	Sets a VPN ID on the Virtual Private LAN Services (VPLS) instance. • Use the same VPN ID for the PE devices that belong to the same VPN. • Make sure the VPN ID is unique for each VPN in the service provider network. The range is from 1 to 4294967295.
Step 5	**member** *ip-address* **encapsulation mpls** **Example:** `Device(config-vfi)# member 102.102.102.102 encapsulation mpls`	Specifies the devices that form a point-to-point L2VPN VFI connection. • *ip-address*—IP address of the VFI neighbor. • **encapsulation mpls**—Specifies Multiprotocol Label Switching (MPLS) as the data encapsulation method.
Step 6	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Creates a pseudowire that is to be used to transport BPDU packets between the two N-PE devices.
Step 7	**exit** **Example:** `Device(config-vfi)# exit`	Returns to global configuration mode.
Step 8	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 10`	Configures components on a bridge domain, and enters bridge-domain configuration mode.
Step 9	**member vfi** *vfi-name* **Example:** `Device(config-bdomain)# member vfi VPLS-10`	Configures the VFI member in the bridge-domain.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Configuring the VPLS Pseudowire Between the N-PE Devices using the commands associated with the L2VPN Protocol-Based CLIs feature

Configuring network provider edge (N-PE) redundancy in a hierarchical Virtual Private LAN service (H-VPLS) network requires that you configure the VPLS pseudowire for transporting bridge protocol data unit (BPDU) packets. For the core pseudowire between the N-PE devices, you configure a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) and attach the VFI to a bridge-domain (described here). Then, in the next task, you bind the service instance to the bridge-domain. This configuration provides a redundancy that provides improved reliability against link and node failures.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *vpn id*
5. **member** *ip-address* **encapsulation mpls**
6. **forward permit l2protocol all**
7. **exit**
8. **bridge-domain** *bridge-id*
9. **member vfi** *vfi-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context VPLS-10`	Establishes a L2VPN VFI between two or more separate networks, and enters L2VFI configuration mode.

	Command or Action	Purpose
Step 4	**vpn id** *vpn id* **Example:** `Device(config-vfi)# vpn id 10`	Sets a VPN ID on the Virtual Private LAN Services (VPLS) instance. • Use the same VPN ID for the PE devices that belong to the same VPN. • Make sure the VPN ID is unique for each VPN in the service provider network. The range is from 1 to 4294967295.
Step 5	**member** *ip-address* **encapsulation mpls** **Example:** `Device(config-vfi)# member 102.102.102.102 encapsulation mpls`	Specifies the devices that form a point-to-point L2VPN VFI connection. • *ip-address*—IP address of the VFI neighbor. • **encapsulation mpls**—Specifies Multiprotocol Label Switching (MPLS) as the data encapsulation method.
Step 6	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Creates a pseudowire that is to be used to transport BPDU packets between the two N-PE devices.
Step 7	**exit** **Example:** `Device(config-vfi)# exit`	Returns to global configuration mode.
Step 8	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 10`	Configures components on a bridge domain, and enters bridge-domain configuration mode.
Step 9	**member vfi** *vfi-name* **Example:** `Device(config-bdomain)# member vfi VPLS-10`	Configures the VFI member in the bridge-domain.
Step 10	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Binding the Service Instance to the Bridge-Domain

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet**

5. **encapsulation dot1q** *vlan-id*
6. **exit**
7. **bridge-domain** *bridge-id*
8. **member** *interface-type-number* **service-instance** *service-id*
9. **end**

DETAILED STEPS

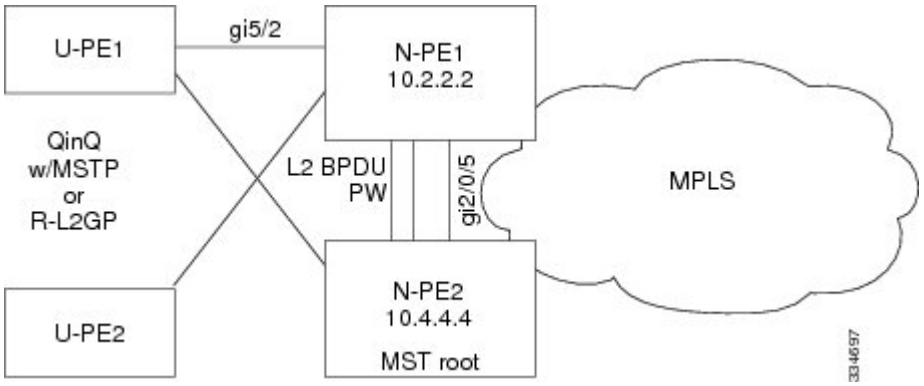
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/1/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Configures an Ethernet service instance on the interface, and enters Ethernet service configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 10`	Enables IEEE 802.1Q encapsulation of traffic on the specified interface in a VLAN.
Step 6	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to global configuration mode.
Step 7	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 10`	Configures components on the bridge domain, and enters bridge-domain configuration mode.
Step 8	**member** *interface-type-number* **service-instance** *service-id* **Example:** `Device(config-bdomain)# member GigabitEthernet0/1/0 service-instance 10`	Binds the service instance to the bridge-domain instance.
Step 9	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Configuration Examples for H-VPLS N-PE Redundancy for QinQ Access

Example: H-VPLS N-PE Redundancy for QinQ Access

The figure below shows a configuration that is set up for the H-VPLS N-PE Redundancy with QinQ Access feature.

Figure 86: H-VPLS N-PE Redundancy with QinQ Access Topology



The table below shows the configuration of two network provider edge (N-PE) devices.

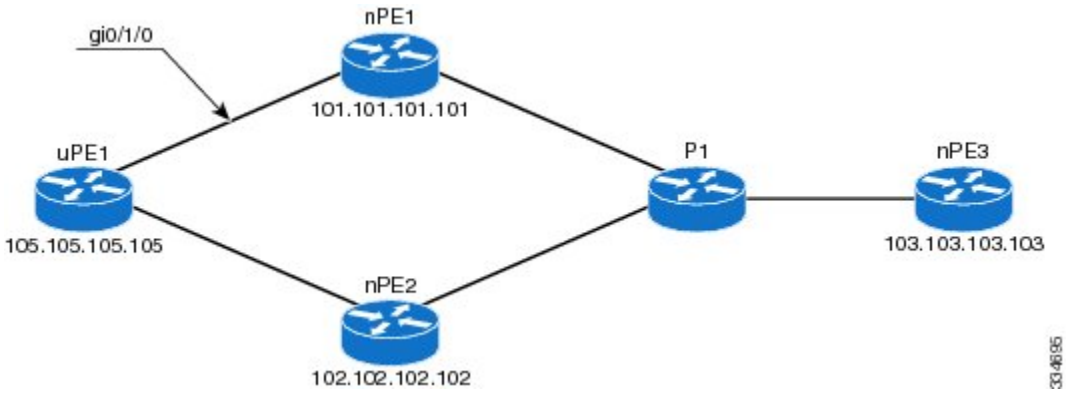
Table 123: Example: H-VPLS N-PE Redundancy for QinQ Access

N-PE1	N-PE2
l2vpn vfi context VPLS-10 vpn id 10 member 10.4.4.4 encapsulation mpls forward permit l2protocol all ! bridge-domain 10 member vfi VPLS-10 member GigabitEthernet5/2 service-instance 10 ! interface GigabitEthernet5/2 service instance 10 ethernet encapsulation dot1q 10 ! spanning-tree mode mst spanning-tree extend system-id ! spanning-tree mst configuration name myMstName revision 10 instance 1 vlan 10	l2vpn vfi context VPLS-10 vpn id 10 member 10.2.2.2 encapsulation mpls forward permit l2protocol all ! bridge-domain 10 member vfi VPLS-10 member GigabitEthernet2/0/5 service-instance 10 ! interface GigabitEthernet2/0/5 service instance 10 ethernet encapsulation dot1q 10 ! spanning-tree mode mst spanning-tree extend system-id ! spanning-tree mst configuration name myMstName revision 10 instance 1 vlan 20 ! spanning-tree mst 1 priority 0

Example: H-VPLS N-PE Redundancy for MPLS Access using the commands associated with the L2VPN Protocol-Based CLIs feature

The figure below shows a configuration that is set up for the H-VPLS N-PE Redundancy with MPLS Access feature. Because there is no option to configure multihoming on access VPLS, the **xconnect** command is used with priority on uPE1.

Figure 87: H-VPLS N-PE Redundancy with MPLS Access Topology



nPE1 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 102.102.102.102 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE2 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE3 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 102.102.102.102 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
```

uPE1 Configuration

```
interface GigabitEthernet0/1/0
 service instance 10 ethernet
 encapsulation dot1q 10
!
l2vpn xconnect context XC-10
 member GigabitEthernet0/1/0 service-instance 10
 member 101.101.101.101 10 encapsulation mpls group pwred priority 9
 member 102.102.102.102 10 encapsulation mpls group pwred priority 10
```

Sample Output on uPE1

```
Device# show l2vpn service peer 101.101.101.101 vcid 10

Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down    IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering    NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                  Prio  St  XC St
  ---------          -----       -------------                  ----  --  -----
VPWS name: foo, State: UP
  Eth1/1.1                       Eth1/1.1:100(Eth VLAN)         0     UP  UP
  pw101              blue        102.1.1.1:100(MPLS)            2     UP  UP
  pw102              blue        103.1.1.1:100(MPLS)            5     SB  IA
  pw103              blue        104.1.1.1:100(MPLS)            8     SB  IA
  pw104              blue        105.1.1.1:100(MPLS)            11    SB  IA

Device# show l2vpn service peer 102.102.102.102 vcid 10

Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down             AD=Admin Down    IA=Inactive
        SB=Standby  HS=Hot Standby      RV=Recovering    NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                  Prio  St  XC St
  ---------          -----       -------------                  ----  --  -----
VPWS name: foo, State: UP
  Eth1/1.1                       Eth1/1.1:100(Eth VLAN)         0     UP  UP
  pw101              blue        102.1.1.1:100(MPLS)            2     UP  UP
  pw102              blue        103.1.1.1:100(MPLS)            5     SB  IA
  pw103              blue        104.1.1.1:100(MPLS)            8     SB  IA
  pw104              blue        105.1.1.1:100(MPLS)            11    SB  IA
```

Additional References for L2VPN VPLS Inter-AS Option B

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Related Topic	Document Title
IP Routing (BGP) commands	Cisco IOS IP Routing: BGP Command Reference
Concepts and tasks related to configuring the VPLS Autodiscovery: BGP Based feature.	*VPLS Autodiscovery BGP Based*
BGP support for the L2VPN address family	*BGP Support for the L2VPN Address Family*
VPLS	"VPLS Overview" section in the *Configuring Multiprotocol Label Switching on the Optical Services Modules* document
L2VPN multisegment pseudowires, MPLS OAM support for L2VPN multisegment pseudowires, MPLS OAM support for L2VPN inter-AS option B	*L2VPN Multisegment Pseudowires*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing standards has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4360	*BGP Extended Communities Attribute*
RFC 4364	*BGP/MPLS IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for H-VPLS N-PE Redundancy for QinQ Access

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 124: Feature Information for H-VPLS N-PE Redundancy for QinQ Access

Feature Name	Releases	Feature Information
H-VPLS N-PE Redundancy for QinQ Access	12.2(33)SRC 12.2(50)SY Cisco IOS XE Release 3.8S	The H-VPLS N-PE Redundancy for QinQ Access feature provides the capability to dual-home a given user provider edge (U-PE) device to two network provide edge (N-PE) devices in order to provide protection against link and node failures. In Cisco IOS Release 12.2(33)SRC, this feature was introduced on the Cisco 7600 series routers. In Cisco IOS Release 12.2(50)SY, this feature was integrated. In Cisco IOS XE Release 3.8S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **forward permit l2protocol**, **show mpls l2transport vc**.

Glossary

CE device—customer edge device. A device that belongs to a customer network, which connects to a PE device to utilize MPLS VPN network services.

LAN—local-area network. High-speed, low-error data network covering a relatively small geographic area. LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited areas.

MPLS—Multiprotocol Label Switching. A packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

MSTP—Multiple Spanning Tree Protocol. MSTP enables multiple VLANs to be mapped to the same spanning-tree instance, reducing the number of spanning-tree instances needed to support a large number of VLANs.

N-PE—network provider edge device. This device acts as a gateway between the MPLS core and edge domains.

PE device—provider edge device. The PE device is the entry point into the service provider network. The PE device is typically deployed on the edge of the network and is administered by the service provider.

pseudowire—A pseudowire is a virtual connection that, in the context of VPLS, connects two SVIs. It is a mechanism that carries the elements of an emulated service from one PE device to one or more PE devices over a packet switched network (PSN). A pseudowire is bidirectional and consists of a pair of unidirectional MPLS virtual circuits (VCs). A pseudowire can be used to connect a point-to-point circuit.

QinQ—An IEEE 802.1Q VLAN tunnel. A mechanism for constructing multipoint Layer 2 VPN using Ethernet switches.

redundancy—The duplication of devices, services, or connections so that, in the event of a failure, they can perform the work of those that failed.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

spanning tree—Loop-free subset of a network topology.

U-PE—user provider edge device. This device connects CE devices to the service.

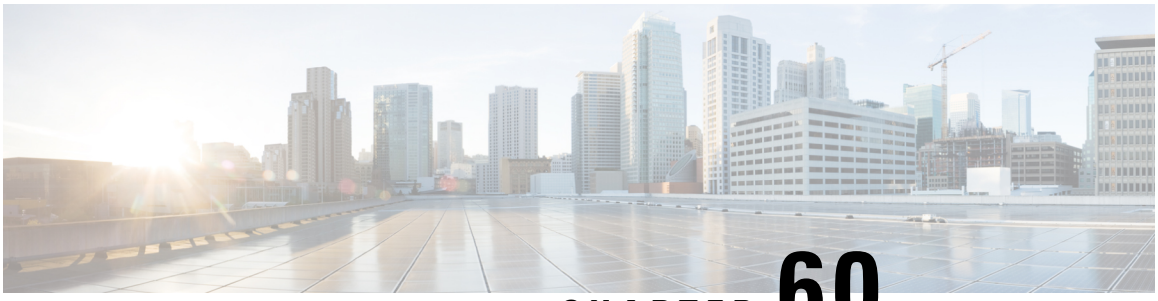
VFI—virtual forwarding instance. A VFI is a collection of data structures used by the data plane, software-based or hardware-based, to forward packets to one or more VCs.

VLAN—Virtual LAN. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments.

VPLS—Virtual Private LAN Service. VPLS describes an architecture that delivers Layer 2 service that emulates an Ethernet LAN across a wide-area network (WAN) and inherits the scaling characteristics of a LAN.

VPLS redundancy—Also called N-PE redundancy. Allows U-PEs to be dual-honed (to their N-PEs) in a loop-free topology with MPLS or QinQ as the access or aggregation domain.

VPN—Virtual Private Network. Allows IP traffic to travel securely over public TCP/IP networks and the Internet by encapsulating and encrypting all IP packets. VPN uses a tunnel to encrypt all information at the IP level.



CHAPTER 60

H-VPLS N-PE Redundancy for MPLS Access

The H-VPLS N-PE Redundancy for MPLS Access feature enables two network provider edge (N-PE) devices to provide failover services to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Having redundant N-PE devices provides improved stability and reliability against link and node failures.

Prerequisites for H-VPLS N-PE Redundancy for MPLS Access

- Before configuring this feature, configure your hierarchical virtual private LAN service (H-VPLS) network and make sure it is operating correctly.
- To provide faster convergence, you can enable the MPLS Traffic Engineering—Fast Reroute feature in the Multiprotocol Label Switching (MPLS) core.
- Enable the L2VPN Pseudowire Redundancy feature on the user provider edge (U-PE) devices for MPLS access.

Restrictions for H-VPLS N-PE Redundancy for MPLS Access

- This feature cannot be used with the VPLS Autodiscovery feature on pseudowires that attach to user provider edge (U-PE) devices. When you create the virtual private LAN service (VPLS), you can manually create the virtual forwarding interface (VFI).
- You cannot configure more than one pseudowire to carry the bridge protocol data unit (BPDU) information between the network provider edge (N-PE) devices.

- You cannot configure a local loopback address as a neighbor when you configure the H-VPLS N-PE Redundancy feature on N-PE devices.
- Only two N-PE devices can be connected to each U-PE device.

Information About H-VPLS N-PE Redundancy for MPLS Access

How H-VPLS N-PE Redundancy for MPLS Access

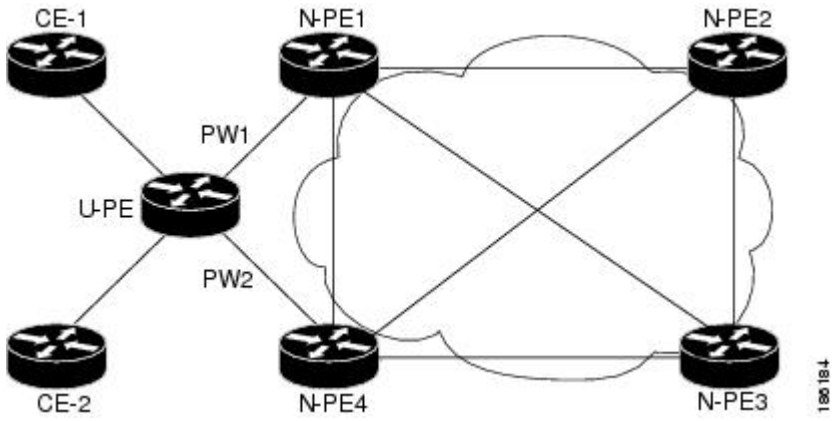
In a network configured with the H-VPLS N-PE Redundancy feature, the user provider edge (U-PE) device is connected to two network provider edge (N-PE) devices. This feature provides a level of redundancy that can tolerate both link and device faults. If a failure occurs in the network that disables one N-PE device from transmitting data, the other N-PE device takes over.

H-VPLS N-PE Redundancy with MPLS Access Based on Pseudowire Redundancy

For the H-VPLS Redundancy with MPLS Access feature based on pseudowire redundancy, the Multiprotocol Label Switching (MPLS) network has pseudowires to the virtual private LAN service (VPLS) core network provider edge (N-PE) devices.

As shown in the figure below, one pseudowire transports data between the user provider edge (U-PE) device and its peer N-PE devices. When a failure occurs along the path of the U-PE device, the backup pseudowire and the redundant N-PE device become active and start transporting data.

Figure 88: H-VPLS N-PE Redundancy for MPLS Access Based on Pseudowire Redundancy



How to Configure H-VPLS N-PE Redundancy for MPLS Access

Specifying the Devices in the Layer 2 VPN VFI

Repeat this task on each N-PE device that is part of the pseudowire redundancy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *vpn id*
5. **member** *ip-address* **encapsulation mpls**
6. **exit**
7. **bridge-domain** *bridge-id*
8. **member vfi** *vfi-name*
9. **member** *ip-address* [*vc-id*] **encapsulation mpls**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context VPLS-10`	Establishes a L2VPN VFI between two or more separate networks, and enters L2VFI configuration mode.
Step 4	**vpn id** *vpn id* **Example:** `Device(config-vfi)# vpn id 10`	Sets a VPN ID on the Virtual Private LAN Services (VPLS) instance. • Use the same VPN ID for the PE devices that belong to the same VPN. • Make sure the VPN ID is unique for each VPN in the service provider network. The range is from 1 to 4294967295.
Step 5	**member** *ip-address* **encapsulation mpls** **Example:** `Device(config-vfi)# member 102.102.102.102 encapsulation mpls`	Specifies the device that forms a point-to-point L2VPN VFI connection. • *ip-address*—IP address of the VFI neighbor (the N-PE device). • **encapsulation mpls**—Specifies Multiprotocol Label Switching (MPLS) as the data encapsulation method.
Step 6	**exit** **Example:**	Returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-vfi)# exit`	
Step 7	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 10`	Configures components on a bridge domain, and enters bridge-domain configuration mode.
Step 8	**member vfi** *vfi-name* **Example:** `Device(config-bdomain)# member vfi VPLS-10`	Configures the VFI member in the bridge-domain.
Step 9	**member** *ip-address* [*vc-id*] **encapsulation mpls** **Example:** `Device(config-vfi)# member 105.105.105.105 10 encapsulation mpls`	Specifies the device that forms a point-to-point Layer 2 VPN (L2VPN) VFI connection. • *ip-address*—IP address of the VFI neighbor (U-PE device). • *vc-id*—Virtual circuit identifier. • **encapsulation mpls**—Specifies MPLS as the data encapsulation method.
Step 10	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Specifying the N-PE Devices That Form the Layer 2 VPN Cross Connection With the U-PE

Perform this task on the U-PE device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **exit**
7. **exit**
8. **l2vpn xconnect context** *context-name*
9. **member gigabitethernet** *interface-number* [**service-instance** *id*]
10. **member** *ip-address vc-id* **encapsulation mpls** [**group** *group-name* [**priority** *number*]]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/1/0`	Specifies the interface to configure, and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Configures an Ethernet service instance on theinterface, and enters Ethernet service configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 10`	Defines the matching criteria to map 802.1Q frames ingress on the interface to the appropriate service instance.
Step 6	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 8	**l2vpn xconnect context** *context-name* **Example:** `Device(config)# l2vpn xconnect context XC-10`	Creates a Layer 2 VPN (L2VPN) cross-connect context, and enters xconnect configuration mode.
Step 9	**member gigabitethernet** *interface-number* [**service-instance** *id*] **Example:** `Device(config-xconnect)# member GigabitEthernet0/1/0 service-instance 10`	Specifies devices that form a Layer 2 VPN (L2VPN) cross connect. • **service-instance** *id*—(Optional) Specifies the service instance identifier.
Step 10	**member** *ip-address vc-id* **encapsulation mpls** [**group** *group-name* [**priority** *number*]] **Example:** `Device(config-xconnect)# member 101.101.101.101 10 encapsulation mpls group pwred priority 9`	Specifies devices that form a Layer 2 VPN (L2VPN) cross connect. • *ip-address*—IP address of the peer N-PE device. • *vc-id*—Virtual circuit identifier.

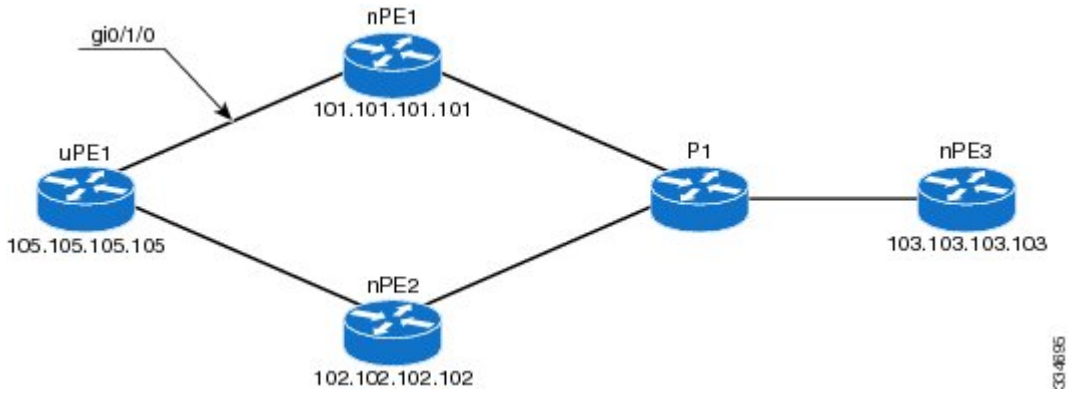
	Command or Action	Purpose
	`Device(config-xconnect)# member 102.102.102.102 10 encapsulation mpls group pwred priority 10`	• **encapsulation mpls**—Specifies Multiprotocol Label Switching (MPLS) as the data encapsulation method. • **group** *group-name*—Specifies the cross-connect member redundancy group name. • **priority** *number*—Specifies the cross-connect member priority. The range is from 0 to 16. The highest priority is 0. Lowest priority is 16.
Step 11	**end** **Example:** `Device(config-xconnect)# end`	Returns to privileged EXEC mode.

Configuration Examples for H-VPLS N-PE Redundancy for MPLS Access

Example: H-VPLS N-PE Redundancy for MPLS Access

The figure below shows a configuration that is set up for the H-VPLS N-PE Redundancy with MPLS Access feature. Since there is no option to configure multihoming on access VPLS, the **xconnect** command is used with priority on uPE1. Please let me know if you need any other info.

Figure 89: H-VPLS N-PE Redundancy with MPLS Access Topology



nPE1 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 102.102.102.102 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
```

```
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE2 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 103.103.103.103 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
 member 105.105.105.105 10 encapsulation mpls
```

nPE3 Configuration

```
l2vpn vfi context VPLS-10
 vpn id 10
 member 101.101.101.101 encapsulation mpls
 member 102.102.102.102 encapsulation mpls
!
bridge-domain 10
 member vfi VPLS-10
```

uPE1 Configuration

```
interface GigabitEthernet0/1/0
 service instance 10 ethernet
 encapsulation dot1q 10
!
l2vpn xconnect context XC-10
 member GigabitEthernet0/1/0 service-instance 10
 member 101.101.101.101 10 encapsulation mpls group pwred priority 9
 member 102.102.102.102 10 encapsulation mpls group pwred priority 10
```

Sample Output on uPE1

```
Device# show xconnect peer 101.101.101.101 vcid 10

Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
UP pri   ac Gi0/1/0:10(Eth VLAN)         UP mpls 101.101.101.101:10           UP

Device# show xconnect peer 102.102.102.102 vcid 10

Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
IA pri   ac Gi0/1/0:10(Eth VLAN)         UP mpls 102.102.102.102:10           SB
Device#
```

Additional References for L2VPN VPLS Inter-AS Option B

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
IP Routing (BGP) commands	Cisco IOS IP Routing: BGP Command Reference
Concepts and tasks related to configuring the VPLS Autodiscovery: BGP Based feature.	*VPLS Autodiscovery BGP Based*
BGP support for the L2VPN address family	*BGP Support for the L2VPN Address Family*
VPLS	"VPLS Overview" section in the *Configuring Multiprotocol Label Switching on the Optical Services Modules* document
L2VPN multisegment pseudowires, MPLS OAM support for L2VPN multisegment pseudowires, MPLS OAM support for L2VPN inter-AS option B	*L2VPN Multisegment Pseudowires*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing standards has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4360	*BGP Extended Communities Attribute*
RFC 4364	*BGP/MPLS IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for H-VPLS N-PE Redundancy for MPLS Access

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 125: Feature Information for H-VPLS N-PE Redundancy for MPLS Access

Feature Name	Releases	Feature Information
H-VPLS N-PE Redundancy for MPLS Access	Cisco IOS XE Release 3.6S	The H-VPLS N-PE Redundancy for MPLS Access feature enables two network provider edge (N-PE) devices to provide redundancy to a user provider edge (U-PE) device in a hierarchical virtual private LAN service (H-VPLS). Having redundant N-PE devices provides improved stability and reliability against link and node failures. In Cisco IOS XE Release 3.6S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **forward permit l2protocol**, **show mpls l2transport vc**.

Glossary

CE device—customer edge device. A device that belongs to a customer network, which connects to a PE device to utilize MPLS VPN network services.

LAN—local-area network. High-speed, low-error data network covering a relatively small geographic area. LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited areas.

MPLS—Multiprotocol Label Switching. A packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

MSTP—Multiple Spanning Tree Protocol. MSTP enables multiple VLANs to be mapped to the same spanning-tree instance, reducing the number of spanning-tree instances needed to support a large number of VLANs.

N-PE—network provider edge device. This device acts as a gateway between the MPLS core and edge domains.

PE device—provider edge device. The PE device is the entry point into the service provider network. The PE device is typically deployed on the edge of the network and is administered by the service provider.

pseudowire—A pseudowire is a virtual connection that, in the context of VPLS, connects two SVIs. It is a mechanism that carries the elements of an emulated service from one PE device to one or more PE devices over a packet switched network (PSN). A pseudowire is bidirectional and consists of a pair of unidirectional MPLS virtual circuits (VCs). A pseudowire can be used to connect a point-to-point circuit.

QinQ—An IEEE 802.1Q VLAN tunnel. A mechanism for constructing multipoint Layer 2 VPN using Ethernet switches.

redundancy—The duplication of devices, services, or connections so that, in the event of a failure, they can perform the work of those that failed.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

spanning tree—Loop-free subset of a network topology.

U-PE—user provider edge device. This device connects CE devices to the service.

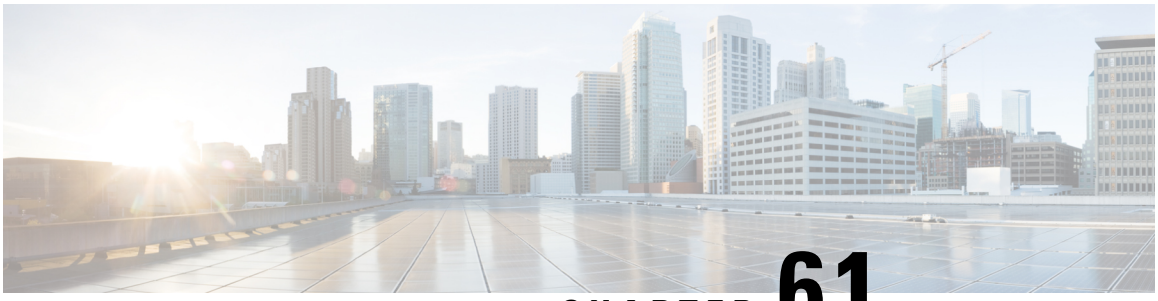
VFI—virtual forwarding instance. A VFI is a collection of data structures used by the data plane, software-based or hardware-based, to forward packets to one or more VCs.

VLAN—Virtual LAN. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments.

VPLS—Virtual Private LAN Service. VPLS describes an architecture that delivers Layer 2 service that emulates an Ethernet LAN across a wide-area network (WAN) and inherits the scaling characteristics of a LAN.

VPLS redundancy—Also called N-PE redundancy. Allows U-PEs to be dual-honed (to their N-PEs) in a loop-free topology with MPLS or QinQ as the access or aggregation domain.

VPN—Virtual Private Network. Allows IP traffic to travel securely over public TCP/IP networks and the Internet by encapsulating and encrypting all IP packets. VPN uses a tunnel to encrypt all information at the IP level.



CHAPTER 61

VPLS MAC Address Withdrawal

The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. A Label Distribution Protocol (LDP)-based MAC address withdrawal message is used for this purpose. A MAC list Type Length Value (TLV) is part of the MAC address withdrawal message. No configuration is needed.

Information About VPLS MAC Address Withdrawal

VPLS MAC Address Withdrawal

The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. A Label Distribution Protocol (LDP)-based MAC address withdrawal message is used for this purpose. A MAC list Type Length Value (TLV) is part of the MAC address withdrawal message.

The **debug mpls ldp messages** and **debug mpls ldp session io** commands support monitoring of MAC address withdrawal messages being exchanged between LDP peers. Any Transport over Multiprotocol Label Switching (AToM) might provide other means to display or monitor MAC address withdrawal messages. The Tag Distribution Protocol (TDP) is not supported because AToM uses only LDP for the MAC address withdrawal message.

PE devices learn the remote MAC addresses and directly attached MAC addresses on customer-facing ports by deriving the topology and forwarding information from packets originating at customer sites. To display the number of MAC address withdrawal messages, enter the **show mpls l2transport vc detail** command, as shown in the following example:

```
Device# show mpls l2transport vc detail

Local interface: VFI TEST VFI up
  MPLS VC type is VFI, interworking type is Ethernet
  Destination address: 10.1.1.1, VC ID: 1000, VC status: up
    Output interface: Se2/0, imposed label stack {17}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
```

```
  Create time: 00:04:34, last status change time: 00:04:15
  Signaling protocol: LDP, peer 10.1.1.1:0 up
    Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.1
    MPLS VC labels: local 16, remote 17
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
    MAC Withdraw: sent 5, received 3
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

VPLS MAC Address Withdrawal Using Commands Associated with L2VPN Protocol-Based Feature

The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. A Label Distribution Protocol (LDP)-based MAC address withdrawal message is used for this purpose. A MAC list Type Length Value (TLV) is part of the MAC address withdrawal message.

The **debug mpls ldp messages** and **debug mpls ldp session io** commands support monitoring of MAC address withdrawal messages being exchanged between LDP peers. Any Transport over Multiprotocol Label Switching (AToM) might provide other means to display or monitor MAC address withdrawal messages. The Tag Distribution Protocol (TDP) is not supported because AToM uses only LDP for the MAC address withdrawal message.

PE devices learn the remote MAC addresses and directly attached MAC addresses on customer-facing ports by deriving the topology and forwarding information from packets originating at customer sites. To display the number of MAC address withdrawal messages, enter the **show l2vpn atom vc detail** command, as shown in the following example:

```
Device# show l2vpn atom vc detail

Local interface: VFI TEST VFI up
  MPLS VC type is VFI, interworking type is Ethernet
  Destination address: 10.1.1.1, VC ID: 1000, VC status: up
    Output interface: Se2/0, imposed label stack {17}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:04:34, last status change time: 00:04:15
  Signaling protocol: LDP, peer 10.1.1.1:0 up
    Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.1
    MPLS VC labels: local 16, remote 17
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
    MAC Withdraw: sent 5, received 3
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

How MAC Address Withdrawal Works with H-VPLS N-PE Redundancy with MPLS Access

If the pseudowire between the user provider edge (U-PE) device and network provider edge (N-PE) device fails, the L2VPN Pseudowire Redundancy feature on the U-PE device activates the standby pseudowire. In addition, the U-PE device sends a Label Distribution Protocol (LDP) MAC address withdrawal request to the new N-PE device, which forwards the message to all pseudowires in the virtual private LAN service (VPLS) core and flushes its MAC address table.

If a on the N-PE device fails, the L2VPN Pseudowire Redundancy feature activates the standby pseudowire and the U-PE device sends a MAC withdrawal message to the newly active N-PE device.

How MAC Address Withdrawal Works with H-VPLS N-PE Redundancy with QinQ Access

If a failure occurs in the customer-switched network, a spanning-tree Topology Change Notification (TCN) is issued to the network provider edge (N-PE) device, which issues a Label Distribution Protocol (LDP)-based MAC address withdrawal message to the peer N-PE devices and flushes its MAC address table.

Additional References for Any Transport over MPLS

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

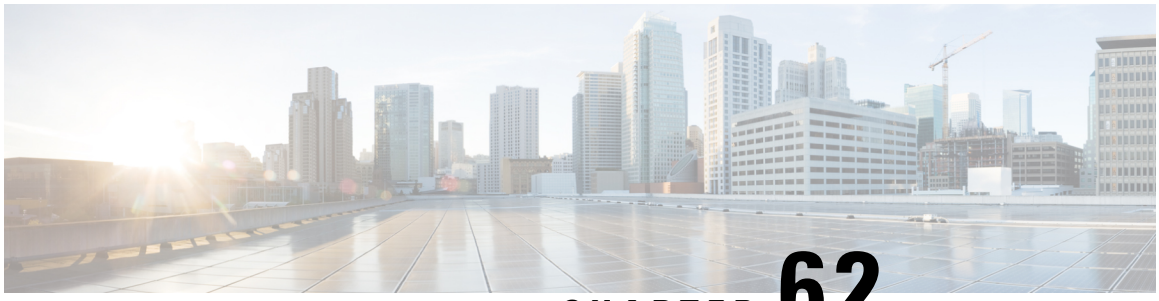
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VPLS MAC Address Withdrawal

Table 126: Feature Information for VPLS MAC Address Withdrawal

Feature Name	Releases	Feature Information
VPLS MAC Address Withdrawal	Cisco IOS XE Release 3.5S	The VPLS MAC Address Withdrawal feature provides faster convergence by removing (or unlearning) MAC addresses that have been dynamically learned. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. No commands were introduced or modified.



CHAPTER 62

Configuring Virtual Private LAN Services

Virtual Private LAN Services (VPLS) enables enterprises to link together their Ethernet-based LANs from multiple sites via the infrastructure provided by their service provider.

This module explains VPLS and how to configure it.

Prerequisites for Virtual Private LAN Services

Before you configure Virtual Private LAN Services (VPLS), ensure that the network is configured as follows:

- Configure IP routing in the core so that provider edge (PE) devices can reach each other via IP.
- Configure Multiprotocol Label Switching (MPLS) in the core so that a label switched path (LSP) exists between PE devices.
- Configure a loopback interface for originating and terminating Layer 2 traffic. Ensure that PE devices can access the loopback interface of the other device. Note that the loopback interface is not required in all cases. For example, tunnel selection does not need a loopback interface when VPLS is directly mapped to a traffic engineering (TE) tunnel.
- Identify peer PE devices and attach Layer 2 circuits to VPLS at each PE device.

Restrictions for Virtual Private LAN Services

The following general restrictions apply to all transport types under Virtual Private LAN Services (VPLS):

- If you do not enable the EFP feature template, then there is no traffic flow between EFP and VFI (when EFP is with Split Horizon group and VFI is default). But when you enable the EFP feature template, then there is traffic flow between EFP and VFI because of design limitations.
- Supported maximum values:

- Total number of virtual forwarding instances (VFIs): 4096 (4 K)

- Software-based data plane is not supported.
- Load sharing and failover on redundant customer-edge-provider-edge (CE-PE) links are not supported.

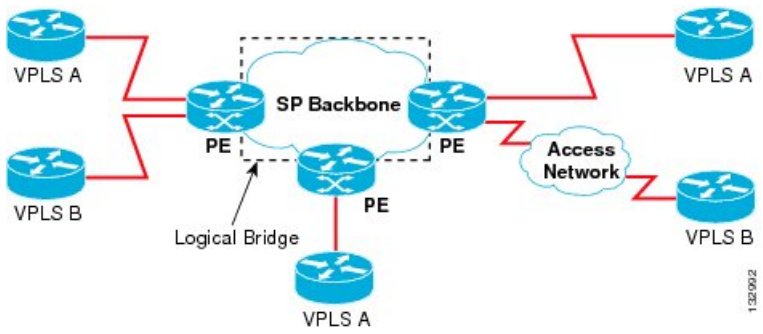
Information About Virtual Private LAN Services

VPLS Overview

Virtual Private LAN Services (VPLS) enables enterprises to link together their Ethernet-based LANs from multiple sites via the infrastructure provided by their service provider. From the enterprise perspective, the service provider's public network looks like one giant Ethernet LAN. For the service provider, VPLS provides an opportunity to deploy another revenue-generating service on top of the existing network without major capital expenditures. Operators can extend the operational life of equipment in their network.

VPLS uses the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All customer edge (CE) devices appear to connect to a logical bridge emulated by the provider core (see the figure below).

Figure 90: VPLS Topology



Full-Mesh Configuration

A full-mesh configuration requires a full mesh of tunnel label switched paths (LSPs) between all provider edge (PE) devices that participate in Virtual Private LAN Services (VPLS). With a full mesh, signaling overhead and packet replication requirements for each provisioned virtual circuit (VC) on a PE can be high.

You set up a VPLS by first creating a virtual forwarding instance (VFI) on each participating PE device. The VFI specifies the VPN ID of a VPLS domain, the addresses of other PE devices in the domain, and the type of tunnel signaling and encapsulation mechanism for each peer PE device.

The set of VFIs formed by the interconnection of the emulated VCs is called a VPLS instance; it is the VPLS instance that forms the logic bridge over a packet switched network. After the VFI has been defined, it needs to be bound to an attachment circuit to the CE device. The VPLS instance is assigned a unique VPN ID.

PE devices use the VFI to establish a full-mesh LSP of emulated VCs to all other PE devices in the VPLS instance. PE devices obtain the membership of a VPLS instance through static configuration using the Cisco IOS CLI.

A full-mesh configuration allows the PE device to maintain a single broadcast domain. When the PE device receives a broadcast, multicast, or unknown unicast packet on an attachment circuit (AC), it sends the packet out on all other ACs and emulated circuits to all other CE devices participating in that VPLS instance. The CE devices see the VPLS instance as an emulated LAN.

To avoid the problem of a packet looping in the provider core, PE devices enforce a "split-horizon" principle for emulated VCs. In a split horizon, if a packet is received on an emulated VC, it is not forwarded on any other emulated VC.

The packet forwarding decision is made by looking up the Layer 2 VFI of a particular VPLS domain.

A VPLS instance on a particular PE device receives Ethernet frames that enter on specific physical or logical ports and populates a MAC table similarly to how an Ethernet switch works. The PE device can use the MAC address to switch these frames into the appropriate LSP for delivery to the another PE device at a remote site.

If the MAC address is not available in the MAC address table, the PE device replicates the Ethernet frame and floods it to all logical ports associated with that VPLS instance, except the ingress port from which it just entered. The PE device updates the MAC table as it receives packets on specific ports and removes addresses not used for specific periods.

Static VPLS Configuration

Virtual Private LAN Services (VPLS) over Multiprotocol Label Switching-Transport Profile (MPLS-TP) tunnels allows you to deploy a multipoint-to-multipoint layer 2 operating environment over an MPLS-TP network for services such as Ethernet connectivity and multicast video. To configure static VPLS, you must specify a static range of MPLS labels using the **mpls label range** command with the **static** keyword.

H-VPLS

Hierarchical VPLS (H-VPLS) reduces signaling and replication overhead by using full-mesh and hub-and-spoke configurations. Hub-and-spoke configurations operate with split horizon to allow packets to be switched between pseudowires (PWs), effectively reducing the number of PWs between provider edge (PE) devices.

Note Split horizon is the default configuration to avoid broadcast packet looping.

Supported Features

Multipoint-to-Multipoint Support

In a multipoint-to-multipoint network, two or more devices are associated over the core network. No single device is designated as the Root node; all devices are considered as Root nodes. All frames can be exchanged directly between the nodes.

Non-Transparent Operation

A virtual Ethernet connection (VEC) can be transparent or non-transparent with respect to Ethernet protocol data units (PDUs). The VEC non-transparency allows users to have a Frame Relay-type service between Layer 3 devices.

Circuit Multiplexing

Circuit multiplexing allows a node to participate in multiple services over a single Ethernet connection. By participating in multiple services, the Ethernet connection is attached to multiple logical networks. Some examples of possible service offerings are VPN services between sites, Internet services, and third-party connectivity for intercompany communications.

MAC-Address Learning, Forwarding, and Aging

Provider edge (PE) devices must learn remote MAC addresses and directly attached MAC addresses on ports that face the external network. MAC address learning accomplishes this by deriving the topology and forwarding information from packets originating at customer sites. A timer is associated with stored MAC addresses. After the timer expires, the entry is removed from the table.

Jumbo Frame Support

Jumbo frame support provides support for frame sizes between 1548 and 9216 bytes. You use the CLI to establish the jumbo frame size for any value specified in the above range. The default value is 1500 bytes in any Layer 2/VLAN interface. You can configure jumbo frame support on a per-interface basis.

Q-in-Q Support and Q-in-Q to EoMPLS Support

With 802.1Q tunneling (Q-in-Q), the customer edge (CE) device issues VLAN-tagged packets and VPLS forwards these packets to a far-end CE device. Q-in-Q refers to the fact that one or more 802.1Q tags may be located in a packet within the interior of the network. As packets are received from a CE device, an additional VLAN tag is added to incoming Ethernet packets to segregate traffic from different CE devices. Untagged packets originating from a CE device use a single tag within the interior of the VLAN switched network, whereas previously tagged packets originating from the CE device use two or more tags.

VPLS Services

Transparent LAN Service

Transparent LAN Service (TLS) is an extension to the point-to-point port-based Ethernet over Multiprotocol Label Switching (EoMPLS), which provides bridging protocol transparency (for example, bridge protocol data units [BPDUs]) and VLAN values. Bridges see this service as an Ethernet segment. With TLS, the PE device forwards all Ethernet packets received from the customer-facing interface (including tagged and untagged packets, and BPDUs) as follows:

- To a local Ethernet interface or an emulated virtual circuit (VC) if the destination MAC address is found in the Layer 2 forwarding table.
- To all other local Ethernet interfaces and emulated VCs belonging to the same VPLS domain if the destination MAC address is a multicast or broadcast address or if the destination MAC address is not found in the Layer 2 forwarding table.

Note You must enable Layer 2 protocol tunneling to run the Cisco Discovery Protocol (CDP), the VLAN Trunking Protocol (VTP), and the Spanning-Tree Protocol (STP).

Ethernet Virtual Connection Service

Ethernet Virtual Connection Service (EVCS) is an extension to the point-to-point VLAN-based Ethernet over MPLS (EoMPLS) that allows devices to reach multiple intranet and extranet locations from a single physical port. With EVCS, the provider edge (PE) device forwards all Ethernet packets with a particular VLAN tag received from the customer-facing interface (excluding bridge protocol data units [BPDUs]) as follows:

- To a local Ethernet interface or to an emulated virtual circuit (VC) if the destination MAC address is found in the Layer 2 forwarding table.
- To all other local Ethernet interfaces and emulated VCs belonging to the same Virtual Private LAN Services (VPLS) domain if the destination MAC address is a multicast or a broadcast address or if the destination MAC address is not found in the Layer 2 forwarding table.

Note Because it has only local significance, the demultiplexing VLAN tag that identifies a VPLS domain is removed before the packet is forwarded to the outgoing Ethernet interfaces or emulated VCs.

VPLS Integrated Routing and Bridging

Virtual Private LAN Services (VPLS) integrated routing and bridging routes Layer 3 traffic and switches Layer 2 frames for pseudowire connections between provider edge (PE) devices using a VPLS multipoint PE device. The ability to route frames to and from these interfaces supports the termination of a pseudowire into a Layer 3 network (VPN or global) on the same switch or to tunnel Layer 3 frames over a Layer 2 tunnel (VPLS).

To configure routing support for a pseudowire, configure an IP address and other Layer 3 features for the Layer 3 domain in interface configuration mode.

Note VPLS integrated routing and bridging does not support multicast routing. VPLS integrated routing and bridging is also known as routed pseudowire and routed VPLS.

The following example shows how to assign IP address 10.10.10.1 to a bridge domain interface (BDI).

```
interface bdi 100
  ip address 10.10.10.1 255.255.255.0
```

VPLS and Type 4 dummy VLAN Tag

From Cisco IOS XE Everest 16.4.1 release, VPLS VC type 4 mode (with autodiscovery) can be used to configure a dummy VLAN tag. This feature can be used to modify the VLAN ID to filter based on the VLAN ID. The dummy VLAN ID is 0 in default VPLS type 4 mode, and can be set to any value from 1 to 4094.

Refer to the section titled *"Example: MAC ACL with Dummy VLAN ID"* in this chapter for the configuration example.

How to Configure Virtual Private LAN Services

Provisioning a Virtual Private LAN Services (VPLS) link involves provisioning the associated attachment circuit and a virtual forwarding instance (VFI) on a provider edge (PE) device.

In Cisco IOS XE Release 3.7S, the L2VPN Protocol-Based CLIs feature was introduced. This feature provides a set of processes and an improved infrastructure for developing and delivering Cisco IOS software on various Cisco platforms. This feature introduces new commands and modifies or replaces existing commands to achieve a consistent functionality across Cisco platforms and provide cross-Operating System (OS) support.

This section consists of tasks that use the commands existing prior to Cisco IOS XE Release 3.7S and a corresponding task that uses the commands introduced or modified by the L2VPN Protocol-Based CLIs feature.

Configuring PE Layer 2 Interfaces on CE Devices

You can configure the Ethernet flow point (EFP) as a Layer 2 virtual interface. You can also select tagged or untagged traffic from a customer edge (CE) device.

Configuring 802.1Q Access Ports for Tagged Traffic from a CE Device

Note When Ethernet Virtual Connection Service (EVCS) is configured, a provider edge (PE) device forwards all Ethernet packets with a particular VLAN tag to a local Ethernet interface or emulated virtual circuit (VC) if the destination MAC address is found in the Layer 2 forwarding table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **negotiation auto**
6. **service instance** *si-id* **ethernet**
7. **encapsulation dot1q** *vlan-id*
8. **bridge-domain** *bd-id*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 5	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 6	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Specifies the service instance ID and enters service instance configuration mode.
Step 7	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 200`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance. Ensure that the interface on the adjoining customer edge (CE) device is on the same VLAN as this PE device.
Step 8	**bridge-domain** *bd-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 9	**end** **Example:** `Device(config-if-srv)# end`	Exits service instance configuration mode and returns to privileged EXEC mode.

Configuring 802.1Q Access Ports for Tagged Traffic from a CE Device: Alternate Configuration

Note When Ethernet Virtual Connection Service (EVCS) is configured, the PE device forwards all Ethernet packets with a particular VLAN tag to a local Ethernet interface or an emulated virtual circuit (VC) if the destination MAC address is found in the Layer 2 forwarding table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **negotiation auto**
6. **service instance** *si-id* **ethernet**
7. **encapsulation dot1q** *vlan-id*
8. **exit**
9. **exit**
10. **bridge-domain** *bd-id*
11. **member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.

	Command or Action	Purpose
Step 5	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 6	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Specifies a service instance ID and enters service instance configuration mode.
Step 7	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 200`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance. • Ensure that the interface on the adjoining customer edge (CE) device is on the same VLAN as this provider edge (PE) device.
Step 8	**exit** **Example:** `Device(config-if-srv)# exit`	Exits service instance configuration mode and returns to interface configuration mode.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**bridge-domain** *bd-id* **Example:** `Device(config)# bridge-domain 100`	Specifies the bridge domain ID and enters bridge-domain configuration mode.
Step 11	**member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*] **Example:** `Device(config-bdomain)# member gigabitethernet0/0/1 service-instance 1000`	Binds a service instance to a bridge domain instance.
Step 12	**end** **Example:** `Device(config-bdomain)# end`	Exits bridge-domain configuration mode and returns to privileged EXEC mode.

Configuring Access Ports for Untagged Traffic from a CE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **negotiation auto**
6. **service instance** *si-id* **ethernet**
7. **encapsulation untagged**
8. **bridge-domain** *bd-id*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 5	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 6	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Specifies a service instance ID and enters service instance configuration mode.

	Command or Action	Purpose
Step 7	**encapsulation untagged** **Example:** `Device(config-if-srv)# encapsulation untagged`	Defines the matching criteria to map untagged ingress Ethernet frames on an interface to the appropriate service instance. • Ensure that the interface on the adjoining customer edge (CE) device is on the same VLAN as this provider edge (PE) device.
Step 8	**bridge-domain** *bd-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance or MAC tunnel to a bridge domain instance.
Step 9	**end** **Example:** `Device(config-if-srv)# end`	Exits service instance configuration mode and returns to privileged EXEC mode.

Configuring 802.1Q Access Ports for Tagged Traffic from a CE Device: Alternate Configuration

Note When Ethernet Virtual Connection Service (EVCS) is configured, the PE device forwards all Ethernet packets with a particular VLAN tag to a local Ethernet interface or an emulated virtual circuit (VC) if the destination MAC address is found in the Layer 2 forwarding table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **negotiation auto**
6. **service instance** *si-id* **ethernet**
7. **encapsulation dot1q** *vlan-id*
8. **exit**
9. **exit**
10. **bridge-domain** *bd-id*
11. **member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 5	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 6	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Specifies a service instance ID and enters service instance configuration mode.
Step 7	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 200`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance. • Ensure that the interface on the adjoining customer edge (CE) device is on the same VLAN as this provider edge (PE) device.
Step 8	**exit** **Example:** `Device(config-if-srv)# exit`	Exits service instance configuration mode and returns to interface configuration mode.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 10	**bridge-domain** *bd-id* **Example:** `Device(config)# bridge-domain 100`	Specifies the bridge domain ID and enters bridge-domain configuration mode.
Step 11	**member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*] **Example:** `Device(config-bdomain)# member gigabitethernet0/0/1 service-instance 1000`	Binds a service instance to a bridge domain instance.
Step 12	**end** **Example:** `Device(config-bdomain)# end`	Exits bridge-domain configuration mode and returns to privileged EXEC mode.

Configuring Q-in-Q EFP

Note When a thread-local storage (TLS) is configured, the provider edge (PE) device forwards all Ethernet packets received from the customer edge (CE) device to all local Ethernet interfaces and emulated virtual circuits (VCs) that belong to the same Virtual Private LAN Services (VPLS) domain if the MAC address is not found in the Layer 2 forwarding table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **negotiation auto**
6. **service instance** *si-id* **ethernet**
7. **encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id*
8. **bridge-domain** *bd-id*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/2`	Specifies an interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 5	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 6	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Specifies a service instance ID and enters service instance configuration mode.
Step 7	**encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 200 second-dot1q 400`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance. • Ensure that the interface on the adjoining CE device is on the same VLAN as this PE device.
Step 8	**bridge-domain** *bd-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance or a MAC tunnel to a bridge domain instance.
Step 9	**end** **Example:** `Device(config-if-srv)# end`	Exits service instance configuration mode and returns to privileged EXEC mode.

Configuring Q-in-Q EFP: Alternate Configuration

Note When a thread-local storage (TLS) is configured, the provider edge (PE) device forwards all Ethernet packets received from the customer edge (CE) device to all local Ethernet interfaces and emulated virtual circuits (VCs) belonging to the same Virtual Private LAN Services (VPLS) domain if the MAC address is not found in the Layer 2 forwarding table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **negotiation auto**
6. **service instance** *si-id* **ethernet**
7. **encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id*
8. **exit**
9. **exit**
10. **bridge-domain** *bd-id*
11. **member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/2`	Specifies an interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.

	Command or Action	Purpose
Step 5	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 6	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Specifies a service instance ID and enters service instance configuration mode.
Step 7	**encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 200 second-dot1q 400`	Defines the matching criteria to map Q-in-Q ingress frames on an interface to the appropriate service instance. • Ensure that the interface on the adjoining CE device is on the same VLAN as this PE device.
Step 8	**exit** **Example:** `Device(config-if-srv)# exit`	Exits service instance configuration mode and returns to interface configuration mode.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**bridge-domain** *bd-id* **Example:** `Device(config)# bridge-domain 100`	Specifies the bridge domain ID and enters bridge-domain configuration mode.
Step 11	**member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*] **Example:** `Device(config-bdomain)# member gigabitethernet0/0/2 service-instance 1000`	Binds a service instance to a bridge domain instance.
Step 12	**end** **Example:** `Device(config-bdomain)# end`	Exits bridge-domain configuration mode and returns to privileged EXEC mode.

Configuring MPLS on a PE Device

To configure Multiprotocol Label Switching (MPLS) on a provider edge (PE) device, configure the required MPLS parameters.

Note Before configuring MPLS, ensure that IP connectivity exists between all PE devices by configuring Interior Gateway Protocol (IGP), Open Shortest Path First (OSPF), or Intermediate System to Intermediate System (IS-IS) between PE devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol** {**ldp** | **tdp**}
4. **mpls ldp logging neighbor-changes**
5. **mpls ldp discovery hello holdtime** *seconds*
6. **mpls ldp router-id** *interface-type-number* [**force**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol** {**ldp**	**tdp**} **Example:** `Device(config)# mpls label protocol ldp`	Specifies the label distribution protocol for the platform.
Step 4	**mpls ldp logging neighbor-changes** **Example:** `Device(config)# mpls ldp logging neighbor-changes`	(Optional) Generates system error logging (syslog) messages when LDP sessions go down.
Step 5	**mpls ldp discovery hello holdtime** *seconds* **Example:** `Device(config)# mpls ldp discovery hello holdtime 5`	Configures the interval between the transmission of consecutive LDP discovery hello messages or the hold time for an LDP transport connection.
Step 6	**mpls ldp router-id** *interface-type-number* [**force**] **Example:**	Specifies a preferred interface for the LDP router ID.

	Command or Action	Purpose
	`Device(config)# mpls ldp router-id loopback0 force`	
Step 7	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a VFI on a PE Device

The virtual forwarding interface (VFI) specifies the VPN ID of a Virtual Private LAN Services (VPLS) domain, the addresses of other provider edge (PE) devices in the domain, and the type of tunnel signaling and encapsulation mechanism for each peer.

Note Only Multiprotocol Label Switching (MPLS) encapsulation is supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **manual**
4. **vpn id** *vpn-id*
5. **neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**]
6. **bridge-domain** *bd-id*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *name* **manual** **Example:** `Device(config)# l2 vfi vfi110 manual`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks and enters VFI configuration mode.

	Command or Action	Purpose
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 110`	Configures a VPN ID for a VPLS domain. • The emulated VCs bound to this Layer 2 virtual routing and forwarding (VRF) instance use this VPN ID for signaling.
Step 5	**neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**] **Example:** `Device(config-vfi)# neighbor 172.16.10.2 4 encapsulation mpls`	Specifies the type of tunnel signaling and encapsulation mechanism for each VPLS peer. **Note** Split horizon is the default configuration to avoid broadcast packet looping and to isolate Layer 2 traffic. Use the **no-split-horizon** keyword to disable split horizon and to configure multiple VCs per spoke into the same VFI.
Step 6	**bridge-domain** *bd-id* **Example:** `Device(config-vfi)# bridge-domain 100`	Specifies a bridge domain.
Step 7	**end** **Example:** `Device(config-vfi)# end`	Exits VFI configuration mode and returns to privileged EXEC mode.

Configuring a VFI on a PE Device: Alternate Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *id*
5. **member** *ip-address* [*vc-id*] **encapsulation mpls**
6. **exit**
7. **bridge-domain** *bd-id*
8. **member vfi** *vfi-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context vfi110`	Establishes a L2VPN VFI between two or more separate networks, and enters VFI configuration mode.
Step 4	**vpn id** *id* **Example:** `Device(config-vfi)# vpn id 110`	Configures a VPN ID for a Virtual Private LAN Services (VPLS) domain. The emulated virtual circuits (VCs) bound to this Layer 2 virtual routing and forwarding (VRF) instance use this VPN ID for signaling.
Step 5	**member** *ip-address* [*vc-id*] **encapsulation mpls** **Example:** `Device(config-vfi)# member 172.16.10.2 4 encapsulation mpls`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection and Multiprotocol Label Switching (MPLS) as the encapsulation type.
Step 6	**exit** **Example:** `Device(config-vfi)# exit`	Exits VFI configuration mode and returns to global configuration mode.
Step 7	**bridge-domain** *bd-id* **Example:** `Device(config)# bridge-domain 100`	Specifies a bridge domain and enters bridge-domain configuration mode.
Step 8	**member vfi** *vfi-name* **Example:** `Device(config-bdomain)# member vfi vfi110`	Binds a VFI instance to a bridge domain instance.
Step 9	**end** **Example:** `Device(config-bdomain)# end`	Exits bridge-domain configuration mode and returns to privileged EXEC mode.

Configuring Static Virtual Private LAN Services

To configure static Virtual Private LAN Services (VPLS), perform the following tasks:

- Configuring a Pseudowire for Static VPLS
- Configuring VFI for Static VPLS
- Configuring a VFI for Static VPLS: Alternate Configuration
- Configuring an Attachment Circuit for Static VPLS
- Configuring an Attachment Circuit for Static VPLS: Alternate Configuration
- Configuring an MPLS-TP Tunnel for Static VPLS with TP
- Configuring a VFI for Static VPLS: Alternate Configuration

Configuring a Pseudowire for Static VPLS

The configuration of pseudowires between provider edge (PE) devices helps in the successful transmission of the Layer 2 frames between PE devices.

Use the pseudowire template to configure the virtual circuit (VC) type for the virtual path identifier (VPI) pseudowire. In the following task, the pseudowire will go through a Multiprotocol Label Switching (MPLS)-Tunneling Protocol (TP) tunnel.

The pseudowire template configuration specifies the characteristics of the tunneling mechanism that is used by the pseudowires, which are:

- Encapsulation type
- Control protocol
- Payload-specific options
- Preferred path

Perform this task to configure a pseudowire template for static Virtual Private LAN Services (VPLS).

Note Ensure that you perform this task before configuring the virtual forwarding instance (VFI) peer. If the VFI peer is configured before the pseudowire class, the configuration is incomplete until the pseudowire class is configured. The **show running-config** command displays an error stating that configuration is incomplete.

```
Device# show running-config | sec vfi

l2 vfi config manual
 vpn id 1000
 ! Incomplete point-to-multipoint vfi config
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** *name*
4. **encapsulation mpls**
5. **signaling protocol none**
6. **preferred-path interface Tunnel-tp** *interface-number*

7. **exit**
8. **interface pseudowire** *number*
9. **source template type pseudowire** *name*
10. **neighbor** *peer-address vcid-value*
11. **label** *local-pseudowire-label remote-pseudowire-label*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire static-vpls`	Specifies the template type as pseudowire and enters template configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-template)# encapsulation mpls`	Specifies the tunneling encapsulation. • For Any Transport over MPLS (AToM), the encapsulation type is MPLS.
Step 5	**signaling protocol none** **Example:** `Device(config-template)# signaling protocol none`	Specifies that no signaling protocol is configured for the pseudowire class.
Step 6	**preferred-path interface Tunnel-tp** *interface-number* **Example:** `Device(config-template)# preferred-path interface Tunnel-tp 1`	(Optional) Specifies the path that traffic uses: an MPLS Traffic Engineering (TE) tunnel or destination IP address and Domain Name Server (DNS) name.
Step 7	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 8	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 1`	Establishes a pseudowire interface and enters interface configuration mode.
Step 9	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire static-vpls`	Configures the source template type of the configured pseudowire.
Step 10	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and VC ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 11	**label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-if)# label 301 17`	Configures an Any Transport over MPLS (AToM) static pseudowire connection by defining local and remote circuit labels.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring VFI for Static VPLS

Note Ensure that you perform this task after configuring the pseudowire. If the VFI peer is configured before the pseudowire, the configuration is incomplete until the pseudowire is configured. The output of the **show running-config** command displays an error stating that configuration is incomplete.

```
Device# show running-config | sec vfi

l2 vfi config manual
 vpn id 1000
 ! Incomplete point-to-multipoint vfi config
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label range** *minimum-value maximum-value* [**static** *minimum-static-value maximum-static-value*]
4. **pseudowire-class** [*pw-class-name*]
5. **encapsulation mpls**
6. **protocol** {**l2tpv2** | **l2tpv3** | **none**} *[l2tp-class-name]*

7. **exit**
8. **l2 vfi** *vfi-name* **manual**
9. **vpn id** *vpn-id*
10. **neighbor** *ip-address* **pw-class** *pw-name*
11. **mpls label** *local-pseudowire-label remote-pseudowire-label*
12. **mpls control-word**
13. **neighbor** *ip-address* **pw-class** *pw-name*
14. **mpls label** *local-pseudowire-label remote-pseudowire-label*
15. **mpls control-word**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *minimum-value maximum-value* [**static** *minimum-static-value maximum-static-value*] **Example:** `Device(config)# mpls label range 16 200 static 300 500`	Configures the range of local labels available for use with Multiprotocol Label Switching (MPLS) applications on packet interfaces.
Step 4	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class static_vpls`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 5	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation as MPLS.
Step 6	**protocol** {**l2tpv2** | **l2tpv3** | **none**} *[l2tp-class-name]* **Example:** `Device(config-pw-class)# protocol none`	Specifies that no signaling protocol will be used in Layer 2 Tunneling Protocol Version 3 (L2TPv3) sessions.
Step 7	**exit** **Example:**	Exits pseudowire class configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-pw-class)# exit`	
Step 8	**l2 vfi** *vfi-name* **manual** **Example:** `Device(config)# l2 vfi static-vfi manual`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks, and enters Layer 2 VFI manual configuration mode.
Step 9	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Specifies the VPN ID.
Step 10	**neighbor** *ip-address* **pw-class** *pw-name* **Example:** `Device(config-vfi)# neighbor 10.3.4.4 pw-class static_vpls`	Specifies the IP address of the peer and the pseudowire class.
Step 11	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 301 17`	Configures an Any Transport over MPLS (AToM) static pseudowire connection by defining local and remote circuit labels.
Step 12	**mpls control-word** **Example:** `Device(config-vfi)# mpls control-word`	(Optional) Enables the MPLS control word in an AToM static pseudowire connection.
Step 13	**neighbor** *ip-address* **pw-class** *pw-name* **Example:** `Device(config-vfi)# neighbor 2.3.4.3 pw-class static_vpls`	Specifies the IP address of the peer and the pseudowire class.
Step 14	**mpls label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-vfi)# mpls label 302 18`	Configures an AToM static pseudowire connection by defining local and remote circuit labels.
Step 15	**mpls control-word** **Example:** `Device(config-vfi)# mpls control-word`	(Optional) Enables the MPLS control word in an AToM static pseudowire connection.

	Command or Action	Purpose
Step 16	**end** **Example:** `Device(config-vfi)# end`	Exits Layer 2 VFI manual configuration mode and returns to privileged EXEC mode.

Configuring a VFI for Static VPLS: Alternate Configuration

Note Ensure that you perform this task after configuring the pseudowire. If the VFI peer is configured before the pseudowire, the configuration is incomplete until the pseudowire is configured. The output of the **show running-config** command displays an error stating that configuration is incomplete.

```
Device# show running-config | sec vfi

l2 vfi config manual
 vpn id 1000
 ! Incomplete point-to-multipoint vfi config
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *vfi-name*
4. **vpn id** *vpn-id*
5. **exit**
6. **interface** *type number*
7. **encapsulation mpls**
8. **neighbor** *ip-address vc-id*
9. **label** *local-pseudowire-label remote-pseudowire-label*
10. **control-word** {**include** | **exclude**}
11. **exit**
12. **bridge-domain** *bd-id*
13. **member vfi** *vfi-name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**l2vpn vfi context** *vfi-name* **Example:** `Device(config)# l2vpn vfi context vpls1`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks and enters VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Specifies the VPN ID.
Step 5	**exit** **Example:** `Device(config-vfi)# exit`	Exits VFI configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface pseudowire 100`	Specifies an interface and enters interface configuration mode.
Step 7	**encapsulation mpls** **Example:** `Device(config-if)# encapsulation mpls`	Specifies an encapsulation type for tunneling Layer 2 traffic over a pseudowire.
Step 8	**neighbor** *ip-address vc-id* **Example:** `Device(config-if)# neighbor 10.3.4.4 100`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 9	**label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-if)# label 301 17`	Configures an Any Transport over MPLS (AToM) static pseudowire connection by defining local and remote circuit labels.
Step 10	**control-word** {**include** | **exclude**} **Example:** `Device(config-if)# control-word include`	(Optional) Enables the Multiprotocol Label Switching (MPLS) control word in an AToM dynamic pseudowire connection.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 12	**bridge-domain** *bd-id* **Example:** `Device(config)# bridge-domain 24`	Specifies the bridge domain ID and enters bridge-domain configuration mode.
Step 13	**member vfi** *vfi-name* **Example:** `Device(config-bdomain)# member vfi vpls1`	Binds a service instance to a bridge domain instance.
Step 14	**end** **Example:** `Device(config-bdomain)# end`	Exits bridge-domain configuration mode and returns to privileged EXEC mode.

Configuring an Attachment Circuit for Static VPLS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot/interface*
4. **service instance** *si-id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **rewrite ingress tag pop** *number* [**symmetric**]
7. **bridge-domain** *bd-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot/interface* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode. • Ensure that the interfaces between the customer edge (CE) and provider edge (PE) devices that run Ethernet

	Command or Action	Purpose
		over MPLS (EoMPLS) are in the same subnet. All other interfaces and backbone devices do not need to be in the same subnet.
Step 4	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 100 ethernet`	Configures an Ethernet service instance on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 200`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance. • Ensure that the interface on the adjoining CE device is on the same VLAN as this PE device.
Step 6	**rewrite ingress tag pop** *number* [**symmetric**] **Example:** `Device(config-if-srv)# rewrite ingress tag pop 1 symmetric`	(Optional) Specifies the encapsulation adjustment to be performed on a frame ingressing a service instance and the tag to be removed from a packet.
Step 7	**bridge-domain** *bd-id* **Example:** `Device(config-if-srv)# bridge-domain 24`	(Optional) Binds a service instance or a MAC tunnel to a bridge domain instance.
Step 8	**end** **Example:** `Device(config-if-srv)# end`	Exits service instance configuration mode and returns to privileged EXEC mode.

Configuring an Attachment Circuit for Static VPLS: Alternate Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot/interface*
4. **service instance** *si-id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **rewrite ingress tag pop** *number* [**symmetric**]
7. **exit**
8. **exit**
9. **bridge-domain** *bd-id*
10. **member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot/interface* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode. • Ensure that the interfaces between the customer edge (CE) and provider edge (PE) devices that are running Ethernet over MPLS (EoMPLS) are in the same subnet. All other interfaces and backbone devices do not need to be in the same subnet.
Step 4	**service instance** *si-id* **ethernet** **Example:** `Device(config-if)# service instance 10 ethernet`	Specifies a service instance ID and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 200`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance. • Ensure that the interface on the adjoining CE device is on the same VLAN as this PE device.
Step 6	**rewrite ingress tag pop** *number* [**symmetric**] **Example:** `Device(config-if-srv)# rewrite ingress tag pop 1 symmetric`	(Optional) Specifies the encapsulation adjustment to be performed on a frame ingressing a service instance and the tag to be removed from a packet.
Step 7	**exit** **Example:** `Device(config-if-srv)# exit`	Exits service instance configuration mode and returns to interface configuration mode.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 9	**bridge-domain** *bd-id* **Example:** `Device(config)# bridge-domain 100`	Specifies the bridge domain ID and enters bridge-domain configuration mode.
Step 10	**member** *interface-type-number* **service-instance** *service-id* [**split-horizon group** *group-id*] **Example:** `Device(config-bdomain)# member gigabitethernet0/0/1 service-instance 1000`	(Optional) Binds a service instance to a bridge domain instance.
Step 11	**end** **Example:** `Device(config-bdomain)# end`	Exits bridge-domain configuration mode and returns to privileged EXEC mode.

Configuring an MPLS-TP Tunnel for Static VPLS with TP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface Tunnel-tp** *number*
4. **no ip address**
5. **no keepalive**
6. **tp destination** *ip-address*
7. **bfd** *bfd-template*
8. **working-lsp**
9. **out-label** *number* **out-link** *number*
10. **lsp-number** *number*
11. **exit**
12. **protect-lsp**
13. **out-label** *number* **out-link** *number*
14. **in-label** *number*
15. **lsp-number** *number*
16. **exit**
17. **exit**
18. **interface** *type number*
19. **ip address** *ip-address ip-mask*
20. **mpls tp link** *link-num* {**ipv4** *ip-address* | **tx-mac** *mac-address*}
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface Tunnel-tp** *number* **Example:** `Device(config)# interface Tunnel-tp 4`	Configures a Multiprotocol Label Switching (MPLS) transport profile tunnel and enters interface configuration mode. • Use the same interface as you configured for the pseudowire class.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Disables the IP address configuration.
Step 5	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables the keepalive configuration.
Step 6	**tp destination** *ip-address* **Example:** `Device(config-if)# tp destination 10.22.22.22`	Configures the tunnel destination.
Step 7	**bfd** *bfd-template* **Example:** `Device(config-if)# bfd tp`	Binds a single-hop Bidirectional Forwarding Detection (BFD) template to an interface.
Step 8	**working-lsp** **Example:** `Device(config-if)# working-lsp`	Configures the working label switched path (LSP) and enters working interface configuration mode.
Step 9	**out-label** *number* **out-link** *number* **Example:** `Device(config-if-working)# out-label 16 out-link 100`	Configures the out link and out label for the working LSP.

	Command or Action	Purpose
Step 10	**lsp-number** *number* **Example:** `Device(config-if-working)# lsp-number 0`	Configures the ID number for the working LSP.
Step 11	**exit** **Example:** `Device(config-if-working)# exit`	Exits working interface configuration mode and returns to interface configuration mode.
Step 12	**protect-lsp** **Example:** `Device(config-if)# protect-lsp`	Enters protection configuration mode for the label switched path (LSP) and enters protect interface configuration mode.
Step 13	**out-label** *number* **out-link** *number* **Example:** `Device(config-if-protect)# out-label 11 out-link 500`	Configures the out link and out label for the protect LSP.
Step 14	**in-label** *number* **Example:** `Device(config-if-protect)# in-label 600`	Configures the in label for the protect LSP.
Step 15	**lsp-number** *number* **Example:** `Device(config-if-protect)# lsp-number 1`	Configures the ID number for the working protect LSP.
Step 16	**exit** **Example:** `Device(config-if-protect)# exit`	Exits protect interface configuration mode and returns to interface configuration mode.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**interface** *type number* **Example:** `Device(config-if)# interface GigabitEthernet 0/1/0`	Configures a interface and enters interface configuration mode.

	Command or Action	Purpose
Step 19	**ip address** *ip-address ip-mask* **Example:** `Device(config)# ip address 10.0.0.1 255.255.255.0`	(Optional) Configures the IP address and mask if not using an IP-less core.
Step 20	**mpls tp link** *link-num* {**ipv4** *ip-address* | **tx-mac** *mac-address*} **Example:** `Device(config-if)# mpls tp link 10 tx-mac 0100.0c99.8877`	Configures Multiprotocol Label Switching (MPLS) transport profile (TP) link parameters.
Step 21	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Virtual Private LAN Services

Example: Configuring 802.1Q Access Ports for Tagged Traffic from a CE Device

This example shows how to configure the tagged traffic:

```
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# service instance 10 ethernet
Device(config-if-srv)# encapsulation dot1q 200
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# end
```

Example: Configuring 802.1Q Access Ports for Tagged Traffic from a CE Device: Alternate Configuration

The following example shows how to configure the tagged traffic:

```
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# service instance 10 ethernet
Device(config-if-srv)# encapsulation dot1q 200
Device(config-if-srv)# exit
Device(config-if)# exit
Device(config)# bridge-domain 100
Device(config-bdomain)# member gigabitethernet0/0/1 service-instance 1000
```

```
Device(config-bdomain)# end
```

Example: Configuring Access Ports for Untagged Traffic from a CE Device

The following example shows how to configure access ports for untagged traffic:

```
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# service instance 10 ethernet
Device(config-if-srv)# encapsulation untagged
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# end
```

The following example shows a virtual forwarding interface (VFI) configuration:

```
Device(config)# l2 vfi VPLSA manual
Device(config-vfi)# vpn id 110
Device(config-vfi)# neighbor 10.11.11.11 encapsulation mpls
Device(config-vfi)# neighbor 10.33.33.33 encapsulation mpls
Device(config-vfi)# neighbor 10.44.44.44 encapsulation mpls
Device(config-vfi)# bridge-domain 110
Device(config-vfi)# end
```

The following example shows a VFI configuration for hub and spoke.

```
Device(config)# l2 vfi VPLSB manual
Device(config-vfi)# vpn id 111
Device(config-vfi)# neighbor 10.99.99.99 encapsulation mpls
Device(config-vfi)# neighbor 10.12.12.12 encapsulation mpls
Device(config-vfi)# neighbor 10.13.13.13 encapsulation mpls no-split-horizon
Device(config-vfi)# bridge-domain 111
Device(config-vfi)# end
```

The output of the **show mpls 12transport vc** command displays various information related to a provide edge (PE) device. The VC ID in the output represents the VPN ID; the VC is identified by the combination of the destination address and the VC ID as shown in the command output. The output of the **show mpls l2transport vc detail** command displays detailed information about virtual circuits (VCs) on a PE device.

```
Device# show mpls l2transport vc 201

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
VFI VPLSA      VFI                  10.11.11.11     110        UP
VFI VPLSA      VFI                  10.33.33.33     110        UP
VFI VPLSA      VFI                  10.44.44.44     110        UP
```

The following sample output from the **show vfi** command displays the VFI status:

```
Device# show vfi VPLSA

VFI name: VPLSA, state: up
  Local attachment circuits:
    Vlan2
```

```
  Neighbors connected via pseudowires:
  Peer Address     VC ID     Split-horizon
  10.11.11.11          110             Y
  10.33.33.33          110             Y
  10.44.44.44          110             Y


Device# show vfi VPLSB

VFI name: VPLSB, state: up
  Local attachment circuits:
    Vlan2
  Neighbors connected via pseudowires:
  Peer Address     VC ID     Split-horizon
  10.99.99.99       111             Y
  10.12.12.12       111             Y
  10.13.13.13       111             N
```

Example: Configuring Access Ports for Untagged Traffic from a CE Device: Alternate Configuration

The following example shows how to configure the untagged traffic.

```
Device(config)# interface GigabitEthernet 0/4/4
Device(config-if)# no ip address
Device(config-if)# negotiation auto
Device(config-if)# service instance 10 ethernet
Device(config-if-srv)# encapsulation untagged
Device(config-if-srv)# exit
Device(config-if)# exit
Device(config)# bridge-domain 100
Device(config-bdomain)# member GigabitEthernet0/4/4 service-instance 10
Device(config-if-srv)# end
```

Example: Configuring Q-in-Q EFP

The following example shows how to configure the tagged traffic.

```
Device(config)# interface GigabitEthernet 0/0/2
Device(config-if)# no ip address
Device(config-if)# negotiate auto
Device(config-if)# service instance 10 ethernet
Device(config-if-srv)# encapsulation dot1q 200 second-dot1q 400
Device(config-if-srv)# bridge-domain 100
Device(config-if-srv)# end
```

Use the **show spanning-tree vlan** command to verify that the ports are not in a blocked state. Use the **show vlan id** command to verify that a specific port is configured to send and receive specific VLAN traffic.

Example: Configuring Q-in-Q in EFP: Alternate Configuration

The following example shows how to configure the tagged traffic:

```
Device(config)# interface GigabitEthernet 0/4/4
Device(config-if)# no ip address
Device(config-if)# nonegotiate auto
Device(config-if)# service instance 10 ethernet
Device(config-if-srv)# encapsulation dot1q 200 second-dot1q 400
Device(config-if-srv)# exit
Device(config-if)# exit
Device(config)# bridge-domain 100
Device(config-bdomain)# member GigabitEthernet0/4/4 service-instance 1000
Device(config-bdomain)# end
```

Use the **show spanning-tree vlan** command to verify that the port is not in a blocked state. Use the **show vlan id** command to verify that a specific port is configured to send and receive a specific VLAN traffic.

Example: Configuring MPLS on a PE Device

The following example shows a global Multiprotocol Label Switching (MPLS) configuration:

```
Device(config)# mpls label protocol ldp
Device(config)# mpls ldp logging neighbor-changes
Device(config)# mpls ldp discovery hello holdtime 5
Device(config)# mpls ldp router-id Loopback0 force
```

The following sample output from the **show ip cef** command displays the Label Distribution Protocol (LDP) label assigned:

```
Device# show ip cef 192.168.17.7

192.168.17.7/32, version 272, epoch 0, cached adjacency to POS4/1
0 packets, 0 bytes
  tag information set
    local tag: 8149
    fast tag rewrite with PO4/1, point2point, tags imposed: {4017}
  via 10.3.1.4, POS4/1, 283 dependencies
    next hop 10.3.1.4, POS4/1
    valid cached adjacency
    tag rewrite with PO4/1, point2point, tags imposed: {4017}
```

Example: VFI on a PE Device

The following example shows a virtual forwarding instance (VFI) configuration:

```
Device(config)# l2 vfi vfi110 manual
Device(config-vfi)# vpn id 110
Device(config-vfi)# neighbor 172.16.10.2 4 encapsulation mpls
Device(config-vfi)# neighbor 10.16.33.33 encapsulation mpls
Device(config-vfi)# neighbor 198.51.100.44 encapsulation mpls
Device(config-vfi)# bridge-domain 100
Device(config-vfi)# end
```

The following example shows a VFI configuration for a hub-and-spoke configuration:

```
Device(config)# l2 vfi VPLSA manual
Device(config-vfi)# vpn id 110
Device(config-vfi)# neighbor 10.9.9.9 encapsulation mpls
Device(config-vfi)# neighbor 192.0.2.12 encapsulation mpls
Device(config-vfi)# neighbor 203.0.113.4 encapsulation mpls no-split-horizon
Device(config-vfi)# bridge-domain 100
Device(config-vfi)# end
```

The **show mpls 12transport vc** command displays information about the provider edge (PE) device. The **show mpls l2transport vc detail** command displays detailed information about the virtual circuits (VCs) on a PE device.

```
Device# show mpls l2transport vc 201

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
VFI test1      VFI                  209.165.201.1   201        UP
VFI test1      VFI                  209.165.201.2   201        UP
VFI test1      VFI                  209.165.201.3   201        UP
```

The **show vfi** *vfi-name* command displays VFI status. The VC ID in the output represents the VPN ID; the VC is identified by the combination of the destination address and the VC ID as in the example below.

```
Device# show vfi VPLS-2

VFI name: VPLS-2, state: up
  Local attachment circuits:
    Vlan2
  Neighbors connected via pseudowires:
  Peer Address     VC ID     Split-horizon
  10.1.1.1          2             Y
  10.1.1.2          2             Y
  10.2.2.3          2             N
```

Example: VFI on a PE Device: Alternate Configuration

The following example shows how to configure a virtual forwarding interface (VFI) on a provider edge (PE) device:

```
Device(config)# l2vpn vfi context vfi110
Device(config-vfi)# vpn id 110
Device(config-vfi)# member 172.16.10.2 4 encapsulation mpls
Device(config-vfi)# member 10.33.33.33 encapsulation mpls
Device(config-vfi)# member 10.44.44.44 encapsulation mpls
Device(config-vfi)# exit
Device(config)# bridge-domain 100
Device(config-bdomain)# member vfi vfi110
Device(config-bdomain)# end
```

The following example shows how to configure a hub-and-spoke VFI configuration:.

```
Device(config)# l2vpn vfi context VPLSA
Device(config-vfi)# vpn id 110
```

```
Device(config-vfi)#  member 10.9.9.9 encapsulation mpls
Device(config-vfi)#  member 172.16.10.2 4 encapsulation mpls
Device(config-vfi)#  exit
Device(config)# bridge-domain 100
Device(config-bdomain)# member vfi VPLSA
Device(config-bdomain)# member GigabitEthernet0/0/0 service-instance 100
Device(config-bdomain)# member 10.33.33.33 10 encapsulation mpls
Device(config-bdomain)# end
```

The **show l2vpn atom vc** command displays information about the PE device. The command also displays information about Any Transport over MPLS (AToM) virtual circuits (VCs) and static pseudowires that are enabled to route Layer 2 packets on a device.

```
Device# show l2vpn atom vc

Local intf    Local circuit           Dest address    VC ID      Status
------------- ----------------------- --------------- ---------- ----------
Et0/0.1       Eth VLAN 101            10.0.0.2        101        UP
Et0/0.1       Eth VLAN 101            10.0.0.3        201        DOWN
```

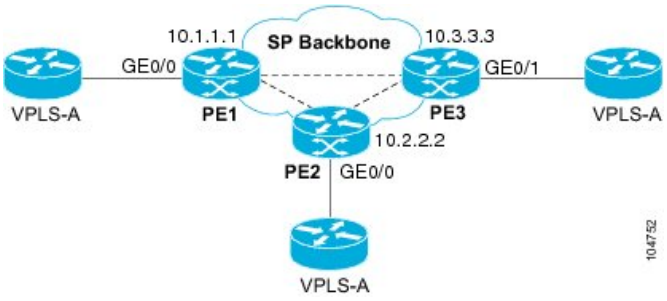
The **show l2vpn vfi** command displays the VFI status. The VC ID in the output represents the VPN ID; the VC is identified by the combination of the destination address and the VC ID as in the example below.

```
Device# show l2vpn vfi VPLS-2

Legend: RT= Route-target

VFI name: serviceCore1, State: UP, Signaling Protocol: LDP
  VPN ID: 100, VPLS-ID: 9:10, Bridge-domain vlan: 100
  RD: 9:10, RT: 10.10.10.10:150
  Pseudo-port Interface: Virtual-Ethernet1000

  Neighbors connected via pseudowires:
  Interface    Peer Address    VC ID      Discovered Router ID   Next Hop
  Pw2000       10.0.0.1        10         10.0.0.1               10.0.0.1
  Pw2001       10.0.0.2        10         10.1.1.2               10.0.0.2
  Pw2002       10.0.0.3        10         10.1.1.3               10.0.0.3
  Pw5          10.0.0.4        10         -                      10.0.0.4
```

Example: Full-Mesh VPLS Configuration

In a full-mesh configuration, each provider edge (PE) device creates a multipoint-to-multipoint forwarding relationship with all other PE devices in the Virtual Private LAN Services (VPLS) domain using a virtual forwarding interface (VFI). An Ethernet or a VLAN packet received from the customer network can be forwarded to one or more local interfaces and/or emulated virtual circuits (VCs) in the VPLS domain. To avoid a broadcast packet loop in the network, packets received from an emulated VC cannot be forwarded to any emulated VC in the VPLS domain on a PE device. Ensure that Layer 2 split horizon is enabled to avoid a broadcast packet loop in a full-mesh network.

Figure 91: Full-Mesh VPLS Configuration



PE 1 Configuration

The following examples shows how to create virtual switch instances (VSIs) and associated VCs:

```
l2 vfi PE1-VPLS-A manual
 vpn id 100
 neighbor 10.2.2.2 encapsulation mpls
 neighbor 10.3.3.3 encapsulation mpls
 bridge domain 100
!
interface Loopback 0
 ip address 10.1.1.1 255.255.0.0
```

The following example shows how to configure the customer edge (CE) device interface (there can be multiple Layer 2 interfaces in a VLAN):

```
interface GigabitEthernet 0/0/0
 no ip address
 negotiation auto
 service instance 10 ethernet
 encapsulation dot1q 200
 bridge-domain 100
```

PE 2 Configuration

The following example shows how to create VSIs and associated VCs.

```
l2 vfi PE2-VPLS-A manual
 vpn id 100
 neighbor 10.1.1.1 encapsulation mpls
 neighbor 10.3.3.3 encapsulation mpls
 bridge domain 100
!
interface Loopback 0
 ip address 10.2.2.2 255.255.0.0
```

The following example shows how to configure the CE device interface (there can be multiple Layer 2 interfaces in a VLAN):

```
interface GigabitEthernet 0/0/0
 no ip address
 negotiation auto
```

```
 service instance 10 ethernet
 encapsulation dot1q 200
 bridge-domain 100
```

PE 3 Configuration

The following example shows how to create VSIs and associated VCs:

```
l2 vfi PE3-VPLS-A manual
 vpn id 112
 neighbor 10.1.1.1 encapsulation mpls
 neighbor 10.2.2.2 encapsulation mpls
 bridge domain 100
!
interface Loopback 0
 ip address 10.3.3.3 255.255.0.0
```

The following example shows how to configure the CE device interface (there can be multiple Layer 2 interfaces in a VLAN).

```
interface GigabitEthernet 0/0/1
 no ip address
 negotiation auto
 service instance 10 ethernet
 encapsulation dot1q 200
 bridge-domain 100
!
```

The following sample output from the **show mpls l2 vc** command provides information about the status of the VC:

```
Device# show mpls l2 vc

Local intf       Local circuit        Dest address    VC ID      Status
-------------    -------------------- --------------- ---------- ----------
VFI PE1-VPLS-A   VFI                  10.2.2.2         100        UP
VFI PE1-VPLS-A   VFI                  10.3.3.3         100        UP
```

The following sample output from the **show vfi** command provides information about the VFI:

```
Device# show vfi PE1-VPLS-A

VFI name: VPLSA, state: up
  Local attachment circuits:
    Vlan200
  Neighbors connected via pseudowires:
    10.2.2.2  10.3.3.3
```

The following sample output from the **show mpls 12transport vc** command provides information about virtual circuits:

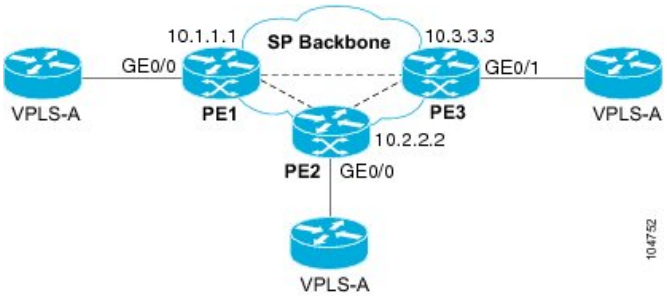
```
Device# show mpls l2transport vc detail

Local interface: VFI PE1-VPLS-A up
  Destination address: 10.2.2.2, VC ID: 100, VC status: up
    Tunnel label: imp-null, next hop point2point
```

```
    Output interface: Se2/0, imposed label stack {18}
  Create time: 3d15h, last status change time: 1d03h
  Signaling protocol: LDP, peer 10.2.2.2:0 up
    MPLS VC labels: local 18, remote 18
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

Example: Full-Mesh Configuration : Alternate Configuration

In a full-mesh configuration, each provider edge (PE) router creates a multipoint-to-multipoint forwarding relationship with all other PE routers in the Virtual Private LAN Services (VPLS) domain using a virtual forwarding interface (VFI). An Ethernet or virtual LAN (VLAN) packet received from the customer network can be forwarded to one or more local interfaces and/or emulated virtual circuits (VCs) in the VPLS domain. To avoid broadcasted packets looping in the network, no packet received from an emulated VC can be forwarded to any emulated VC of the VPLS domain on a PE router. That is, Layer 2 split horizon should always be enabled as the default in a full-mesh network.

Figure 92: VPLS Configuration Example



PE 1 Configuration

The following example shows how to create virtual switch instances (VSIs) and associated VCs and to configure the CE device interface (there can be multiple Layer 2 interfaces in a VLAN):

```
interface gigabitethernet 0/0/0
 service instance 100 ethernet
 encap dot1q 100
 no shutdown
!
l2vpn vfi context PE1-VPLS-A
 vpn id 100
 neighbor 10.2.2.2 encapsulation mpls
 neighbor 10.3.3.3 encapsulation mpls
!
bridge-domain 100
 member gigabitethernet0/0/0 service-instance 100
 member vfi PE1-VPLS-A
```

PE 2 Configuration

The following example shows how to create VSIs and associated VCs and to configure the CE device interface (there can be multiple Layer 2 interfaces in a VLAN):

```
interface gigabitethernet 0/0/0
 service instance 100 ethernet
 encap dot1q 100
 no shutdown
!
l2vpn vfi context PE2-VPLS-A
 vpn id 100
 neighbor 10.1.1.1 encapsulation mpls
 neighbor 10.3.3.3 encapsulation mpls
!
bridge-domain 100
 member gigabitethernet0/0/0 service-instance 100
 member vfi PE2-VPLS-A
```

PE 3 Configuration

The following example shows how to create of the VSIs and associated VCs and to configure the CE device interface (there can be multiple Layer 2 interfaces in a VLAN):

```
interface gigabitethernet 0/0/0
 service instance 100 ethernet
 encap dot1q 100
 no shutdown
!
l2vpn vfi context PE3-VPLS-A
 vpn id 100
 neighbor 10.1.1.1 encapsulation mpls
 neighbor 10.2.2.2 encapsulation mpls
!
bridge-domain 100
 member gigabitethernet0/0/0 service-instance 100
 member vfi PE3-VPLS-A
```

The following sample output from the **show mpls l2 vc** command provides information on the status of the VC:

```
Device# show mpls l2 vc

Local intf       Local circuit  Dest address    VC ID      Status
-------------    -------------- --------------- ---------- ----------
VFI PE3-VPLS-A   VFI            10.2.2.2        100        UP
VFI PE3-VPLS-A   VFI            10.3.3.3        100        UP
```

The following sample output from the **show l2vpn vfi** command provides information about the VFI:

```
Device# show l2vpn vfi VPLS-2

Legend: RT= Route-target

VFI name: serviceCore1, State: UP, Signaling Protocol: LDP
  VPN ID: 100, VPLS-ID: 9:10, Bridge-domain vlan: 100
  RD: 9:10, RT: 10.10.10.10:150
```

```
  Pseudo-port Interface: Virtual-Ethernet1000

  Neighbors connected via pseudowires:
  Interface    Peer Address    VC ID      Discovered Router ID   Next Hop
  Pw2000       10.0.0.1        10         10.0.0.1               10.0.0.1
  Pw2001       10.0.0.2        10         10.1.1.2               10.0.0.2
  Pw2002       10.0.0.3        10         10.1.1.3               10.0.0.3
  Pw5          10.0.0.4        10         -                      10.0.0.4
```

The following sample output from the **show l2vpn atom vc** command provides information on the virtual circuits:

```
Device# show l2vpn atom vc

Local intf    Local circuit           Dest address    VC ID      Status
------------- ----------------------- --------------- ---------- ----------
Et0/0.1       Eth VLAN 101            10.0.0.2        101        UP
Et0/0.1       Eth VLAN 101            10.0.0.3        201        DOWN
```

Example: MAC ACL with Dummy VLAN ID

PE basic configuration for VPLS type 4

```
 router bgp 100
 bgp log-neighbor-changes
 neighbor 19.0.0.1 remote-as 100
 neighbor 19.0.0.1 update-source Loopback0
 !
 address-family ipv4
  neighbor 19.0.0.1 activate
  neighbor 19.0.0.1 send-community extended
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 19.0.0.1 activate
 exit-address-family
l2vpn vfi context vlan_tag
 vpn id 10
 autodiscovery bgp signaling ldp template vlan_tag
!
mpls label protocol ldp
bridge-domain 10
 member GigabitEthernet2/1/0 service-instance 10
  remote circuit id 191
 member vfi vlan_tag
template type pseudowire vlan_tag
 encapsulation mpls
 vc type vlan
 control-word include
interface GigabitEthernet2/1/0
 no ip address
 negotiation auto
 service instance 10 ethernet
  encapsulation dot1q 10
 !
interface GigabitEthernet2/1/4
 ip address 108.0.0.2 255.255.255.0
 negotiation auto
 mpls ip
```

```
!

//Change the circuit ID and check if the download ID is correct//
bridge-domain 10
 member gigabitEthernet 2/1/0 service-instance 10
  remote circuit id 1982 <<< Set the dummy VLAN
```

Verifying the Configuration

Here's a sample output for the **show** command to verify the configured VLAN ID.

```
Device# show platform hardware qfp active feature bridge-domain client 10 interface

QFP L2BD datapath interface information
Name: GigabitEthernet2/1/0.EFP10
IF handle: 26, Input uidb: 245752
Flags: 0X000038
Split-horizon cfged: No, shg id: 0
STP state: Unknown/Bad
Mac security enabled:
MAC limit: 65536, MAC learned: 0
BD PPE addr: 0X8CBF3C00
efp circuit id: 1982 <<< The configured VLAN ID
```

Feature Information for Configuring Virtual Private LAN Services

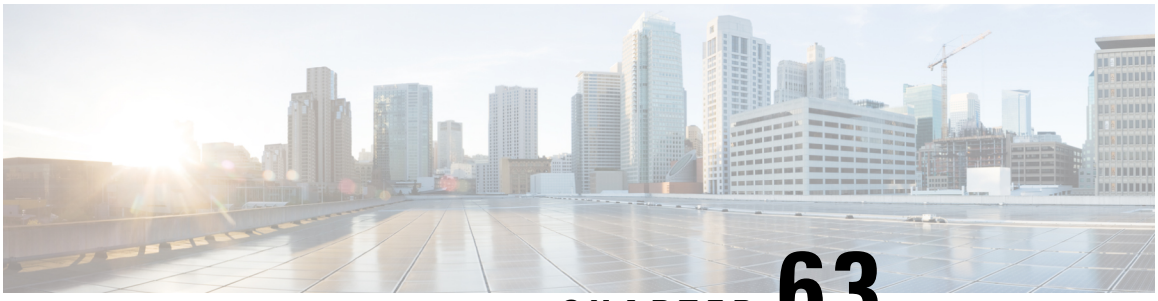
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 127: Feature Information for Configuring Virtual Private LAN Services

Feature Name	Releases	Feature Information
Virtual Private LAN Services (VPLS)	Cisco IOS XE Release 3.5S	This feature enables you to configure dynamic Virtual Private LAN Services (VPLS). VPLS is a class of VPN that supports the connection of multiple sites in a single bridged domain over a managed IP/MPLS network. In Cisco IOS XE Release 3.5S, this feature was introduced on the Cisco ASR 903 Series Aggregation Services Routers.

Feature Name	Releases	Feature Information
L2VPN Protocol-Based CLIs	Cisco IOS XE Release 3.7S	In Cisco IOS XE Release 3.7S, the L2VPN Protocol-Based CLIs feature was introduced. This feature provides a set of processes and an improved infrastructure for developing and delivering Cisco IOS software on various Cisco platforms. This feature introduces new commands and modifies or replaces existing commands to achieve a consistent functionality across Cisco platforms and provide cross-Operating System support.
Static VPLS over MPLS-TP	Cisco IOS XE Release 3.6S	This features enables static VPLS to use MPLS Transport Profile. In Cisco IOS XE Release 3.6S, this feature was introduced on the Cisco ASR 903 Series Aggregation Services Routers.
Type 4 PWE VLAN Rewrite	Cisco IOS XE Everest Release 16.4.1	From Cisco IOS XE Everest 16.4.1 release, VPLS VC type 4 mode (with autodiscovery) can be used to configure a dummy VLAN tag. This feature can be used to modify the VLAN ID to filter based on the VLAN ID.



CHAPTER 63

Routed Pseudo-Wire and Routed VPLS

This feature module explains how to configure Routed Pseudo-Wire and Routed VPLS .

Configuring Routed Pseudo-Wire and Routed VPLS

RPW and Routed VPLS can route Layer 3 traffic as well as switch Layer 2 frames for pseudowire connections between provider edge (PE) devices. Both point-to-point PE connections, in the form of Ethernet over MPLS (EoMPLS), and Virtual Private LAN Services (VPLS) multipoint PE connections are supported. The ability to route frames to and from these interfaces supports termination of a pseudowire into a Layer 3 network (VPN or global) on the same switch, or to tunnel Layer 3 frames over a Layer 2 tunnel (EoMPLS or VPLS). The feature supports faster network convergence in the event of a physical interface or device failure through the MPLS Traffic Engineering (MPLS-TE) and Fast Reroute (FRR) features. In particular, the feature enables MPLS TE-FRR protection for Layer 3 multicast over a VPLS domain.

When the RPW is configured in A-VPLS mode, TE/FRR is not supported because A-VPLS runs over ECMP and the ECMP convergence is comparable to TE/FRR.

To configure routing support for the pseudowire, configure an IP address and other Layer 3 features for the Layer 3 domain (VPN or global) in the virtual LAN (VLAN) interface configuration. The following example assigns the IP address 10.10.10.1 to the VLAN 100 interface, and enables Multicast PIM. (Layer 2 forwarding is defined by the VFI VFI100.)

```
interface bdi 100


  ip address 10.10.10.1 255.255.255.0
```

The following example assigns an IP address 20.20.20.1 of the VPN domain VFI200. (Layer 2 forwarding is defined by the VFI VFI200.)

```
interface bdi 200


  ip address 20.20.20.1 255.255.255.0
```

Verifying Routed Pseudo-Wire and Routed VPLS Configuration

You can use the **show mpls platform** command to view information about a routed pseudowire and routed VPLS configuration.

The following example shows how to display information about a routed pseudowire and routed VPLS configuration:

SUMMARY STEPS

1. show mpls platform vpls 100

DETAILED STEPS

show mpls platform vpls 100

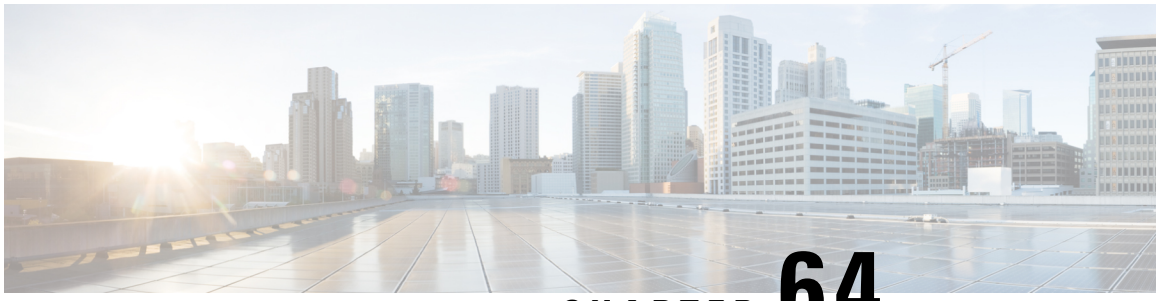
Example:

```
Device# show mpls platform vpls 100

----------------------------
VPLS VLAN 100 (BD 100): V4
  VC info (#spoke VCs 0) :
    Imp:  tcam 224    (68    ) adj 131076 (0x20004) [peer 1.1.1.1 ID vc_id 100 2:1]  \
stats 0/0 0/0
    Disp: tcam 324    (66    ) adj 114692 (0x1C004) [in_label 16] stats 0/0
------------------------------
BD Flood Manager: VLAN/BD 100, 3 peers, V4
  CMET handle 0x8 top 8 (0x8) bottom 3280 (0xCD0)
  Ingr flood: tcam 64/0x40 (sw 15) adj 196608   (0x30000) elif 0x701C0064 stats 0/0  \
0/0
  Egr  flood: tcam 65/0x41 (sw 72) adj 180228   (0x2C004) elif 0x701C0064 stats 0/0  \
0/0
    BD ports:       adj 32868    (0x8064) elif 0x20000064 stats 3/208
  Ingr local: tcam 32/0x20 (sw 13) adj 180224   (0x2C000) elif 0x20000064 stats 0/0
  Egr  local: tcam 33/0x21 (sw 14) adj 180225   (0x2C001) elif 0x20000064 stats 0/0
  IRB Ingr V4 Mcast control 162/0xA2 (sw 79), adj 196609   (0x30001)
      Egr  V4 Mcast control 164/0xA4 (sw 84), adj 180229   (0x2C005)
      Ingr V4 Mcast data 192/0xC0 (sw 80), adj 1966
(0x30000)
      Egr  V4 Mcast data 194/0xC2 (sw 85), adj 180228   (0x2C004)
      Ingr V4 Bcast 34/0x22 (sw 81), adj 196609   (0x30001)
      Egr  V4 Bcast 35/0x23 (sw 86), adj 180229   (0x2C005)
  IRB Ingr V6 Mcast control 608/0x260 (sw 82), adj 196608   (0x30000)
      Egr  V6 Mcast control 612/0x264 (sw 89), adj 180228   (0x2C004)
      Ingr V6 Mcast data 672/0x2A0 (sw 83), adj 196608   (0x30000)
      Egr  V6 Mcast data 676/0x2A4 (sw 90), adj 180228   (0x2C004)
      ip2irb local 36/0x24 (sw 87), adj 180226   (0x2C002) stats 0/0
      ip2irb flood 66/0x42 (sw 88), adj 180230   (0x2C006) stats 0/0
BD Flood Manager: 1 BDs, LTL base 0x90E, LTL clients: VPLS
                : Wildcard entry tcam 288 (12) adj 78089 (0x13109)
```

Feature Information for Routed Pseudo-Wire and Routed VPLS

Table 128: Feature Information for Routed Pseudo-Wire and Routed VPLS

Feature Name	Releases	Feature Information
Routed Pseudo-Wire and Routed VPLS	12.2(33)SRB 12.2(33)SXJ1 15.0(1)SY 15.2(4)M Cisco IOS XE Release 3.6S	This feature routes Layer 3 traffic as well as switch Layer 2 frames for pseudowire connections between provider edge (PE) devices. In Cisco IOS Release 12.2(33)SRB, this feature was introduced on the Cisco 7600 series routers. In Cisco IOS Release 12.2(33)SXJ1, this feature was integrated. This feature is supported on WAN cards. The following command was modified: **show mpls platform** In Cisco IOS Release 15.0(1)SY, this feature was integrated. In Cisco IOS Release 15.2(4)M, this feature was integrated. In Cisco IOS XE Release 3.6S, support was added for the Cisco ASR 1000 Series Routers.



CHAPTER 64

VPLS Autodiscovery BGP Based

VPLS Autodiscovery enables Virtual Private LAN Service (VPLS) provider edge (PE) devices to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also automatically detects when PE devices are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

This module describes how to configure BGP-based VPLS Autodiscovery.

Restrictions for VPLS Autodiscovery BGP Based

- Virtual Private LAN Service (VPLS) Autodiscovery supports only IPv4 addresses.
- VPLS Autodiscovery uses Forwarding Equivalence Class (FEC) 129 to convey endpoint information. Manually configured pseudowires use FEC 128.
- VPLS Autodiscovery is not supported with Layer 2 Tunnel Protocol Version 3 (L2TPv3).
- You can configure both autodiscovered and manually configured pseudowires in a single virtual forwarding instance (VFI). However, you cannot configure different pseudowires on the same peer PE device.
- After enabling VPLS Autodiscovery, if you manually configure a neighbor by using the **neighbor** command and both peers are in autodiscovery mode, each peer will receive discovery data for that VPLS. To prevent peers from receiving data for the VPLS domain, manually configure route target (RT) values.
- If you manually configure multiple pseudowires and target different IP addresses on the same PE device for each pseudowire, do not use the same virtual circuit (VC) ID to identify pseudowires that terminate at the same PE device.
- If you manually configure a neighbor on one PE device, you cannot configure the same pseudowire in the other direction by using autodiscovery on another PE device.

- Tunnel selection is not supported with autodiscovered neighbors.
- Up to 16 RTs are supported per VFI.
- The same RT is not allowed in multiple VFIs on the same PE device.
- The Border Gateway Protocol (BGP) autodiscovery process does not support dynamic, hierarchical VPLS. User-facing PE (U-PE) devices cannot discover network-facing PE (N-PE) devices, and N-PE devices cannot discover U-PE devices.
- Pseudowires for autodiscovered neighbors have split horizon enabled. (A split horizon is enabled by default on all interfaces. A split horizon blocks route information from being advertised by a device, irrespective of the interface from which the information originates.) Therefore, manually configure pseudowires for hierarchical VPLS. Ensure that U-PE devices do not participate in BGP autodiscovery for these pseudowires.
- Do not disable split horizon on autodiscovered neighbors. Split horizon is required with VPLS Autodiscovery.
- The provisioned peer address must be a /32 address bound to the peer's Label Distribution Protocol (LDP) router ID.
- A peer PE device must be able to access the IP address that is used as the local LDP router ID. Even if the IP address is not used in the **xconnect** command on the peer PE device, the IP address must be reachable.

Information About VPLS Autodiscovery BGP Based

How VPLS Works

Virtual Private LAN Service (VPLS) allows Multiprotocol Label Switching (MPLS) networks to provide multipoint Ethernet LAN services, also known as Transparent LAN Services (TLS). All customer sites in a VPLS appear to be on the same LAN, even though these sites might be in different geographic locations.

How the VPLS Autodiscovery BGP Based Feature Works

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. Autodiscovery and signaling functions use the Border Gateway Protocol (BGP) to find and track PE devices.

BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching

(MPLS) network. For more information about BGP and the L2VPN address family in relation to VPLS Autodiscovery, see the following chapters in the *IP Routing: BGP Configuration Guide*:

- "BGP Support for the L2VPN Address Family" chapter

How Enabling VPLS Autodiscovery Differs from Manually Configuring VPLS

With VPLS Autodiscovery enabled, you no longer need to manually set up Virtual Private LAN Service (VPLS). The commands that you use to set up VPLS Autodiscovery are similar to those that you use to manually configure VPLS, as shown in the table below. VPLS Autodiscovery uses **neighbor** commands in L2VPN address family mode to distribute endpoint information to configure a pseudowire.

Table 129: Manual VPLS Configuration Versus VPLS Autodiscovery Configuration

Manual Configuration of VPLS	VPLS Autodiscovery BGP Based

`l2 vfi vpls1 manual`
` vpn id 100`
` neighbor 10.10.10.1 encapsulation mpls`
` neighbor 10.10.10.0 encapsulation mpls`
` exit`
	`l2 vfi vpls1 autodiscovery`
	` vpn id 100`
	` exit`
	`router bgp 1`
	` no bgp default ipv4-unicast`
	` bgp log-neighbor-changes`
	` bgp update-delay 1`
	` neighbor 10.1.1.2 remote-as 1`
	` neighbor 10.1.1.2 update-source Loopback1`
	` .`
	` .`
	` .`
	` address-family l2vpn vpls`
	` neighbor 10.1.1.2 activate`
	` neighbor 10.1.1.2 send-community extended`
	`exit-address-family`

Configure VPLS Autodiscovery by using the **l2 vfi autodiscovery** command. This command allows a virtual forwarding instance (VFI) to learn and advertise pseudowire endpoints. As a result, you no longer need to enter the **neighbor** command in L2 VFI configuration mode.

However, the **neighbor** command is still supported with VPLS Autodiscovery in L2 VFI configuration mode. You can use the **neighbor** command to allow PE devices that do not participate in the autodiscovery process to join the VPLS domain. You can also use the **neighbor** command with PE devices that have been configured using the Tunnel Selection feature. In addition, you can use the **neighbor** command in hierarchical VPLS configurations that have user-facing PE (U-PE) devices that do not participate in the autodiscovery process and have split-horizon forwarding disabled.

How Enabling VPLS Autodiscovery Differs from Manually Configuring VPLS using the commands associated with the L2VPN Protocol-Based CLIs feature

With VPLS Autodiscovery enabled, you no longer need to manually set up Virtual Private LAN Service (VPLS). The commands that you use to set up VPLS Autodiscovery are similar to those that you use to manually configure VPLS, as shown in the table below. VPLS Autodiscovery uses **neighbor** commands in L2VPN address family mode to distribute endpoint information to configure a pseudowire.

Table 130: Manual VPLS Configuration Versus VPLS Autodiscovery Configuration

Manual Configuration of VPLS	VPLS Autodiscovery BGP Based
l2vpn vfi context vpls1 vpn id 100 neighbor 10.10.10.1 encapsulation mpls neighbor 10.10.10.0 encapsulation mpls exit	l2vpn vfi context vpls1 vpn id 100 autodiscovery bgp signaling ldp exit router bgp 1 no bgp default ipv4-unicast bgp log-neighbor-changes bgp update-delay 1 neighbor 10.1.1.2 remote-as 1 neighbor 10.1.1.2 update-source Loopback1 . . . address-family l2vpn vpls neighbor 10.1.1.2 activate neighbor 10.1.1.2 send-community extended exit-address-family

Configure VPLS Autodiscovery by using the **autodiscovery** command. This command allows a virtual forwarding instance (VFI) to learn and advertise pseudowire endpoints. As a result, you no longer need to enter the **neighbor** command in L2 VFI configuration mode.

However, the **neighbor** command is still supported with VPLS Autodiscovery in L2 VFI configuration mode. You can use the **neighbor** command to allow PE devices that do not participate in the autodiscovery process to join the VPLS domain. You can also use the **neighbor** command with PE devices that have been configured using the Tunnel Selection feature. In addition, you can use the **neighbor** command in hierarchical VPLS configurations that have user-facing PE (U-PE) devices that do not participate in the autodiscovery process and have split-horizon forwarding disabled.

show Commands Affected by VPLS Autodiscovery BGP Based

The following **show** commands were enhanced for VPLS Autodiscovery:

- The **show mpls l2transport vc detail** command was updated to include Forwarding Equivalence Class (FEC) 129 signaling information for autodiscovered Virtual Private LAN Service (VPLS) pseudowires.
- The **show vfi** command was enhanced to display information related to autodiscovered virtual forwarding instances (VFIs). The new output includes the VPLS ID, the route distinguisher (RD), the route target (RT), and router IDs of discovered peers.
- The **show xconnect** command was updated with the **rib** keyword to provide Routing Information Base (RIB) information about pseudowires.

BGP VPLS Autodiscovery Support on a Route Reflector

By default, routes received from an internal BGP (iBGP) peer are not sent to another iBGP peer unless a full mesh configuration is formed between all BGP devices within an autonomous system (AS). This results in scalability issues. Using Border Gateway Protocol (BGP) route reflectors leads to much higher levels of scalability. Configuring a route reflector allows a device to advertise or reflect the iBGP learned routes to other iBGP speakers.

Virtual Private LAN Service (VPLS) Autodiscovery supports BGP route reflectors. A BGP route reflector can be used to reflect BGP VPLS prefixes without VPLS being explicitly configured on the route reflector.

A route reflector does not participate in autodiscovery; that is, no pseudowires are set up between the route reflector and the PE devices. A route reflector reflects VPLS prefixes to other PE devices so that these PE devices do not need to have a full mesh of BGP sessions. The network administrator configures only the BGP VPLS address family on a route reflector. For an example configuration of VPLS Autodiscovery support on a route reflector, see the "Example: BGP VPLS Autodiscovery Support on Route Reflector" section.

N-PE Access to VPLS Using MST

When a Virtual Private LAN Service (VPLS) network uses multihoming (network-facing PE [N-PE] VPLS redundancy) to prevent a single point of failure of an N-PE device, a bridging loop is introduced. One of the N-PE devices can be set as a Multiple Spanning Tree (MST) root to break the loop. In most cases, the two N-PE devices are also separated by a distance that makes direct physical link impossible. You can configure a virtual link (usually through the same VPLS core network) between the two N-PE devices to pass an MST bridge protocol data unit (BPDU) for path calculation, break the loop, and maintain convergence. The virtual link is created using a special pseudowire between the active and redundant N-PE devices.

While setting up an MST topology for a VPLS PE device, ensure the following:

- The **spanning-tree mode mst** command is enabled on all PE devices (N-PE and user-facing PE [U-PE]) participating in the MST topology.
- A special pseudowire is configured between the two N-PE devices, and these two devices are in the up state.
- The special pseudowire is a manually created virtual forwarding instance (VFI).
- The configuration (inlcuding the MST instance, the Ethernet virtual circuit [EVC], and the VLAN) on all PE devices is the same.
- One of the N-PE devices, and not one of the U-PE devices, is the root for the MST instance.
- The name and revision for the MST configuration are configured to synchronize with the standby Route Processor (RP).

How to Configure VPLS Autodiscovery BGP Based

Enabling VPLS Autodiscovery BGP Based

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices that are part of the same VPLS domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *vfi-name* **autodiscovery**
4. **vpn id** *vpn-id*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *vfi-name* **autodiscovery** **Example:** `Device(config)# l2 vfi vpls1 autodiscovery`	Enables VPLS Autodiscovery on a PE device and enters L2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.
Step 5	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. • Commands take effect after the device exits L2 VFI configuration mode.

Enabling VPLS Autodiscovery BGP Based using the commands associated with the L2VPN Protocol-Based CLIs feature

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices that are part of the same VPLS domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *vfi-name*
4. **vpn id** *vpn-id*
5. **autodiscovery bgp signaling** {**ldp** | **bgp**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *vfi-name* **Example:** `Device(config)# l2vpn vfi context vpls1`	Establishes an L2VPN VFI context and enters L2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.
Step 5	**autodiscovery bgp signaling** {**ldp** | **bgp**} **Example:** `Device(config-vfi)# autodiscovery bgp signaling ldp`	Enables the VPLS Autodiscovery: BGP Based feature on the PE device.
Step 6	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. • Commands take effect after the device exits L2 VFI configuration mode.

Configuring VPLS BGP Signaling

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *vpn-id*
5. **autodiscovery bgp signaling** {**bgp** | **ldp**} [**template** *template-name*]
6. **ve id** *ve-id*
7. **ve range** *ve-range*
8. **exit**
9. **exit**
10. **router bgp** *autonomous-system-number*
11. **bgp graceful-restart**
12. **neighbor** *ip-address* **remote-as** *autonomous-system-number*

13. **address-family l2vpn** [**vpls**]
14. **neighbor** *ip-address* **activate**
15. **neighbor** *ip-address* **send-community** [**both** | **standard** | **extended**]
16. **neighbor** *ip-address* **suppress-signaling-protocol ldp**
17. **end**
18. **show bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context vfi1`	Establishes a L2VPN virtual forwarding interface (VFI) between two or more separate networks and enters Layer 2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Configures a VPN ID for the VPLS domain.
Step 5	**autodiscovery bgp signaling** {**bgp**	**ldp**} [**template** *template-name*] **Example:** `Device(config-vfi)# autodiscovery bgp signaling bgp`	Enables BGP signaling and discovery or LDP signaling and enters L2VPN VFI autodiscovery configuration mode. **Note** For the VPLS BGP Signaling feature use the **autodiscovery bgp signaling bgp** command.
Step 6	**ve id** *ve-id* **Example:** `Device(config-vfi-autodiscovery)# ve id 1001`	Specifies the VPLS endpoint (VE) device ID value. The VE ID identifies a VFI within a VPLS service. The VE device ID value is from 1 to 16384.
Step 7	**ve range** *ve-range* **Example:** `Device(config-vfi-autodiscovery)# ve range 12`	Specifies the VE device ID range value. The VE range overrides the minimum size of VE blocks. The default minimum size is 10. Any configured VE range must be higher than 10.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Device(config-vfi-autodiscovery)# exit`	Exits L2VPN VFI autodiscovery configuration mode and enters L2VPN VFI configuration mode.
Step 9	**exit** **Example:** `Device(config-vfi)# exit`	Exits L2VPN VFI configuration mode and enters global configuration mode.
Step 10	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode to create or configure a BGP routing process.
Step 11	**bgp graceful-restart** **Example:** `Device(config-router)# bgp graceful-restart`	Enables the BGP graceful restart capability and BGP nonstop forwarding (NSF) awareness.
Step 12	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 remote-as 100`	Configures peering with a BGP neighbor in the specified autonomous system.
Step 13	**address-family l2vpn** [**vpls**] **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The optional **vpls** keyword specifies that VPLS endpoint provisioning information is to be distributed to BGP peers. In this example, an L2VPN VPLS address family session is created.
Step 14	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.1 activate`	Enables the neighbor to exchange information for the L2VPN VPLS address family with the local device.
Step 15	**neighbor** *ip-address* **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 10.10.10.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.

	Command or Action	Purpose
Step 16	**neighbor** *ip-address* **suppress-signaling-protocol ldp** **Example:** `Device(config-router-af)# neighbor 10.10.10.1 suppress-signaling-protocol ldp`	Suppresses LDP signaling and enables BGP signaling. • In this example LDP signaling is suppressed (and BGP signaling enabled) for the neighbor at 10.10.10.1.
Step 17	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 18	**show bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*} **Example:** `Device# show bgp l2vpn vpls all`	(Optional) Displays information about the L2VPN VPLS address family.

Configuring BGP to Enable VPLS Autodiscovery

The Border Gateway Protocol (BGP) Layer 2 VPN (L2VPN) address family supports a separate L2VPN Routing Information Base (RIB) that contains endpoint provisioning information for Virtual Private LAN Service (VPLS) Autodiscovery. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time a Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **bgp log-neighbor-changes**
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. Repeat Steps 6 and 7 to configure other BGP neighbors.
9. **address-family l2vpn** [**vpls**]
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **send-community** {**both** | **standard** | **extended**}
12. Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.
13. **exit-address-family**
14. **end**
15. **show vfi**
16. **show ip bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured using the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast** router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.
Step 5	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 remote-as 65000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 update-source loopback1`	(Optional) Configures a device to select a specific source or interface to receive routing table updates. • This example uses a loopback interface. The advantage of this configuration is that the loopback interface is not affected by the effects of a flapping interface.
Step 8	Repeat Steps 6 and 7 to configure other BGP neighbors.	—
Step 9	**address-family l2vpn** [**vpls**] **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The optional **vpls** keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers. • In this example, an L2VPN VPLS address family session is created.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **send-community** {**both** | **standard** | **extended**} **Example:** `Device(config-router-af)# neighbor 10.10.10.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.
Step 12	Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.	—
Step 13	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 14	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 15	**show vfi** **Example:** `Device# show vfi`	Displays information about the configured VFI instances.
Step 16	**show ip bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*} **Example:** `Device# show ip bgp l2vpn vpls all`	Displays information about the L2VPN VPLS address family.

Customizing the VPLS Autodiscovery Settings

Several commands allow you to customize the Virtual Private LAN Service (VPLS) environment. You can specify identifiers for the VPLS domain, the route distinguisher (RD), the route target (RT), and the provider edge (PE) device. Perform this task to customize these identifiers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *vfi-name* **autodiscovery**
4. **vpn id** *vpn-id*
5. **vpls-id** {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
6. **rd** {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
7. **route-target** [**import** | **export** | **both**] {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
8. **auto-route-target**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *vfi-name* **autodiscovery** **Example:** `Device(config)# l2 vfi vpls1 autodiscovery`	Enables VPLS Autodiscovery on the PE device and enters Layer 2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.
Step 5	**vpls-id** {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# vpls-id 5:300`	(Optional) Assigns an identifier to the VPLS domain. • This command is optional because VPLS Autodiscovery automatically generates a VPLS ID using the Border Gateway Protocol (BGP) autonomous system (AS) number and the configured VFI VPN ID. You can use this command to change the automatically generated VPLS ID. • There are two formats for configuring the VPLS ID argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*)

	Command or Action	Purpose
		format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 6	**rd** {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# rd 2:3`	(Optional) Specifies the RD to distribute endpoint information. • This command is optional because VPLS Autodiscovery automatically generates an RD using the BGP autonomous system number and the configured VFI VPN ID. You can use this command to change the automatically generated RD. • There are two formats for configuring the route distinguisher argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 7	**route-target** [**import** | **export** | **both**] {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# route-target 600:2222`	(Optional) Specifies the RT. • This command is optional because VPLS Autodiscovery automatically generates an RT using the lower 6 bytes of the RD and the VPLS ID. You can use this command to change the automatically generated RT. • There are two formats for configuring the route target argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 8	**auto-route-target** **Example:** `Device(config-vfi)# auto-route-target`	(Optional) Enables the automatic generation of a RT.
Step 9	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. • Commands take effect after the device exits Layer 2 VFI configuration mode.

Configuring BGP to Enable VPLS Autodiscovery using the commands associated with the L2VPN Protocol-Based CLIs feature

The BGP L2VPN address family supports a separate L2VPN Routing Information Base (RIB) that contains endpoint provisioning information for Virtual Private LAN Service (VPLS) Autodiscovery. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time a Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **bgp log-neighbor-changes**
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. Repeat Steps 6 and 7 to configure other BGP neighbors.
9. **address-family l2vpn** [**vpls**]
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **send-community** {**both** | **standard** | **extended**}
12. Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.
13. **exit-address-family**
14. **end**
15. **show l2vpn vfi**
16. **show ip bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured using the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast** router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.
Step 5	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 remote-as 65000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 update-source loopback1`	(Optional) Configures a device to select a specific source or interface to receive routing table updates. • This example uses a loopback interface. The advantage of this configuration is that the loopback interface is not affected by the effects of a flapping interface.
Step 8	Repeat Steps 6 and 7 to configure other BGP neighbors.	—
Step 9	**address-family l2vpn** [**vpls**] **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The optional **vpls** keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers.

	Command or Action	Purpose
		• In this example, an L2VPN VPLS address family session is created.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **send-community** {**both** | **standard** | **extended**} **Example:** `Device(config-router-af)# neighbor 10.10.10.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.
Step 12	Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.	—
Step 13	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 14	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 15	**show l2vpn vfi** **Example:** `Device# show l2vpn vfi`	Displays information about the Layer 2 VPN (L2VPN) virtual forwarding instances (VFI).
Step 16	**show ip bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*} **Example:** `Device# show ip bgp l2vpn vpls all`	Displays information about the L2VPN VPLS address family.

Customizing the VPLS Autodiscovery Settings using the commands associated with the L2VPN Protocol-Based CLIs feature

Several commands allow you to customize the Virtual Private LAN Service (VPLS) environment. You can specify identifiers for the VPLS domain, the route distinguisher (RD), the route target (RT), and the provider edge (PE) device. Perform this task to customize these identifiers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *vfi-name*
4. **vpn id** *vpn-id*
5. **autodiscovery bgp signaling** {**ldp** | **bgp**}
6. **vpls-id** {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
7. **rd** {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
8. **route-target** [**import** | **export** | **both**] {*autonomous-system-number*:*nn* | *ip-address*:*nn*}
9. **auto-route-target**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *vfi-name* **Example:** `Device(config)# l2vpn vfi context vpls1`	Establishes a L2VPN VFI context and enters L2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.
Step 5	**autodiscovery bgp signaling** {**ldp**	**bgp**} **Example:** `Device(config-vfi)# autodiscovery bgp signaling ldp`	Enables the VPLS Autodiscovery: BGP Based feature on the PE device.
Step 6	**vpls-id** {*autonomous-system-number*:*nn*	*ip-address*:*nn*} **Example:** `Device(config-vfi)# vpls-id 5:300`	(Optional) Assigns an identifier to the VPLS domain. • This command is optional because VPLS Autodiscovery automatically generates a VPLS ID using the Border Gateway Protocol (BGP) autonomous system (AS) number and the configured

	Command or Action	Purpose
		VFI VPN ID. You can use this command to change the automatically generated VPLS ID. • There are two formats for configuring the VPLS ID argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 7	**rd** {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# rd 2:3`	(Optional) Specifies the RD to distribute endpoint information. • This command is optional because VPLS Autodiscovery automatically generates an RD using the BGP autonomous system number and the configured VFI VPN ID. You can use this command to change the automatically generated RD. • There are two formats for configuring the route distinguisher argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 8	**route-target** [**import** | **export** | **both**] {*autonomous-system-number*:*nn* | *ip-address*:*nn*} **Example:** `Device(config-vfi)# route-target 600:2222`	(Optional) Specifies the RT. • This command is optional because VPLS Autodiscovery automatically generates an RT using the lower 6 bytes of the RD and the VPLS ID. You can use this command to change the automatically generated RT. • There are two formats for configuring the route target argument. It can be configured in the *autonomous-system-number:network number* (*ASN:nn*) format, as shown in the example, or it can be configured in the *IP-address:network number* format (*IP-address:nn*).
Step 9	**auto-route-target** **Example:** `Device(config-vfi)# auto-route-target`	(Optional) Enables the automatic generation of a RT.
Step 10	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. • Commands take effect after the device exits Layer 2 VFI configuration mode.

Configuring MST on VPLS N-PE Devices

A network-facing PE (N-PE) device is the root bridge for a Multiple Spanning Tree (MST) instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *vfi-name* **manual**
4. **vpn id** *vpn-id*
5. **forward permit l2protocol all**
6. **neighbor** *peer-N-PE-ip-address* **encapsulation mpls**
7. **exit**
8. **spanning-tree mode** [**mst** | **pvst** | **rapid-pvst**]
9. **spanning-tree mst configuration**
10. **name** *name*
11. **revision** *version*
12. **instance** *instance-id* **vlan** *vlan-range*
13. **end**
14. **show spanning-tree mst** [*instance-id* [**detail**] [*interface*] | **configuration** [**digest**] | **detail** | **interface** *type number* [**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2 vfi** *vfi-name* **manual** **Example:** `Device(config)# l2 vfi vpls-mst manual`	Creates a Layer 2 virtual forwarding instance (VFI) and enters Layer 2 VFI manual configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 4000`	Sets or updates the VPN ID on a VPN routing and forwarding (VRF) instance.
Step 5	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Defines the VPLS pseudowire that is used to transport the bridge protocol data unit (BPDU) information between two N-PE devices.

	Command or Action	Purpose
Step 6	**neighbor** *peer-N-PE-ip-address* **encapsulation mpls** **Example:** `Device(config-vfi)# neighbor 10.76.100.12 encapsulation mpls`	Specifies the type of tunnel signaling and encapsulation mechanism for each VPLS peer.
Step 7	**exit** **Example:** `Device(config-vfi)# exit`	Exits Layer 2 VFI manual configuration mode and returns to global configuration mode.
Step 8	**spanning-tree mode** [**mst** | **pvst** | **rapid-pvst**] **Example:** `Device(config)# spanning-tree mode mst`	Switches between MST, Per-VLAN Spanning Tree+ (PVST+), and Rapid-PVST+ modes.
Step 9	**spanning-tree mst configuration** **Example:** `Device(config)# spanning-tree mst configuration`	Enters MST configuration mode.
Step 10	**name** *name* **Example:** `Device(config-mst)# name cisco`	Sets the name for the MST region.
Step 11	**revision** *version* **Example:** `Device(config-mst)# revision 11`	Sets the revision number for the MST configuration.
Step 12	**instance** *instance-id* **vlan** *vlan-range* **Example:** `Device(config-mst)# instance 1 vlan 100`	Maps a VLAN or a group of VLANs to an MST instance.
Step 13	**end** **Example:** `Device(config-mst)# end`	Exits MST configuration mode and enters privileged EXEC mode.
Step 14	**show spanning-tree mst** [*instance-id* [**detail**] [*interface*] | **configuration** [**digest**] | **detail** | **interface** *type number* [**detail**]] **Example:** `Device# show spanning-tree mst 1`	Displays information about the MST configuration.

Configuring MST on VPLS N-PE Devices using the commands associated with the L2VPN Protocol-Based CLIs feature

A network-facing PE (N-PE) device is the root bridge for a Multiple Spanning Tree (MST) instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *vfi-name*
4. **vpn id** *vpn-id*
5. **forward permit l2protocol all**
6. **neighbor** *peer-N-PE-ip-address* **encapsulation mpls**
7. **exit**
8. **spanning-tree mode** [**mst** | **pvst** | **rapid-pvst**]
9. **spanning-tree mst configuration**
10. **name** *name*
11. **revision** *version*
12. **instance** *instance-id* **vlan** *vlan-range*
13. **end**
14. **show spanning-tree mst** [*instance-id* [**detail**] [*interface*] | **configuration** [**digest**] | **detail** | **interface** *type number* [**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *vfi-name* **Example:** `Device(config)# l2vpn vfi context vpls-mst`	Establishes an L2VPN VFI context and enters L2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 4000`	Sets or updates the VPN ID on a VPN routing and forwarding (VRF) instance.
Step 5	**forward permit l2protocol all** **Example:** `Device(config-vfi)# forward permit l2protocol all`	Defines the VPLS pseudowire that is used to transport the bridge protocol data unit (BPDU) information between two N-PE devices.
Step 6	**neighbor** *peer-N-PE-ip-address* **encapsulation mpls** **Example:**	Specifies the type of tunnel signaling and encapsulation mechanism for each VPLS peer.

	Command or Action	Purpose
	`Device(config-vfi)# neighbor 10.76.100.12 encapsulation mpls`	
Step 7	**exit** **Example:** `Device(config-vfi)# exit`	Exits Layer 2 VFI manual configuration mode and returns to global configuration mode.
Step 8	**spanning-tree mode** [**mst** | **pvst** | **rapid-pvst**] **Example:** `Device(config)# spanning-tree mode mst`	Switches between MST, Per-VLAN Spanning Tree+ (PVST+), and Rapid-PVST+ modes.
Step 9	**spanning-tree mst configuration** **Example:** `Device(config)# spanning-tree mst configuration`	Enters MST configuration mode.
Step 10	**name** *name* **Example:** `Device(config-mst)# name cisco`	Sets the name for the MST region.
Step 11	**revision** *version* **Example:** `Device(config-mst)# revision 11`	Sets the revision number for the MST configuration.
Step 12	**instance** *instance-id* **vlan** *vlan-range* **Example:** `Device(config-mst)# instance 1 vlan 100`	Maps a VLAN or a group of VLANs to an MST instance.
Step 13	**end** **Example:** `Device(config-mst)# end`	Exits MST configuration mode and enters privileged EXEC mode.
Step 14	**show spanning-tree mst** [*instance-id* [**detail**] [*interface*] | **configuration** [**digest**] | **detail** | **interface** *type number* [**detail**]] **Example:** `Device# show spanning-tree mst 1`	Displays information about the MST configuration.

Configuration Examples for VPLS Autodiscovery BGP Based

The following examples show the configuration of a network that uses VPLS Autodiscovery:

Example: Enabling VPLS Autodiscovery BGP Based

```
Device> enable
Device# configure terminal
Device(config)# l2 vfi vpls1 autodiscovery
Device(config-vfi)# vpn id 10
Device(config-vfi)# exit
```

Example: Enabling VPLS Autodiscovery BGP Based Using Commands Associated with L2VPN Protocol-Based Feature

```
Device> enable
Device# configure terminal
Device(config)# l2vpn vfi context vpls1
Device(config-vfi)# vpn id 10
Device(config-vfi)# autodiscovery bgp signaling ldp
Device(config-vfi)# exit
```

Example: Configuring BGP to Enable VPLS Autodiscovery

PE1

```
l2 router-id 10.1.1.1
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.1 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.1 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback1
```

```
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
```

PE2

```
l2 router-id 10.1.1.2
l2 vfi auto autodiscovery
 vpn id 100
!
 pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.2 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.2 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback1
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
```

PE3

```
l2 router-id 10.1.1.3
l2 vfi auto autodiscovery
 vpn id 100
```

```
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.3 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.3 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback1
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 exit-address-family
```

Example: Configuring BGP to Enable VPLS Autodiscovery Using Commands Associated with L2VPN Protocol-Based Feature

PE1

```
l2vpn
 router-id 10.1.1.1
l2vpn vfi context auto
 vpn id 100
 autodiscovery bgp signaling ldp
!
interface pseudowire 1
 encapsulation mpls
 neighbor 33.33.33.33 1
!
interface Loopback1
 ip address 10.1.1.1 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.1 255.255.255.0
 mpls ip
!
```

```
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback1
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
```

PE2

```
l2vpn
 router-id 10.1.1.2
l2vpn vfi context auto
 vpn id 100
 autodiscovery bgp signaling ldp

!
 interface pseudowire 1
 encapsulation mpls
 neighbor 33.33.33.33 1
!
interface Loopback1
 ip address 10.1.1.2 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.2 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback1
 neighbor 10.1.1.3 remote-as 1
 neighbor 10.1.1.3 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
```

```
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.3 activate
 neighbor 10.1.1.3 send-community extended
 exit-address-family
```

PE3

```
l2vpn
 router-id 10.1.1.3
l2vpn vfi context auto
 vpn id 100
 autodiscovery bgp signaling ldp

!
interface pseudowire 1
 encapsulation mpls
 neighbor 33.33.33.33 1
!
interface Loopback1
 ip address 10.1.1.3 255.255.255.255
!
interface GigabitEthernet 0/0/1
 description Backbone interface
 ip address 192.168.0.3 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.0.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback1
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 update-source Loopback1
!
 address-family ipv4
 no synchronization
 no auto-summary
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community extended
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 send-community extended
 exit-address-family
```

Example: Customizing VPLS Autodiscovery Settings

```
Device> enable
Device# configure terminal
Device(config)# l2 vfi vpls1 autodiscovery
```

```
Device(config-vfi)# vpn id 10
Device(config-vfi)# vpls-id 5:300
Device(config-vfi)# rd 2:3
Device(config-vfi)# route-target 600:2222
Device(config-vfi)# end
```

Example: Customizing VPLS Autodiscovery Settings using the commands associated with the L2VPN Protocol-Based CLIs feature

```
Device> enable
Device# configure terminal
Device(config)# l2vpn vfi context vpls1
Device(config-vfi)# vpn id 10
Device(config-vfi)# autodiscovery bgp signaling ldp
Device(config-vfi)# vpls-id 5:300
Device(config-vfi)# rd 2:3
Device(config-vfi)# route-target 600:2222
Device(config-vfi)# end
```

Example: Configuring MST on VPLS N-PE Devices

```
Device> enable
Device# configure terminal
Device(config)# l2 vfi vpls-mst manual
Device(config-vfi)# vpn id 4000
Device(config-vfi)# forward permit l2protocol all
Device(config-vfi)# neighbor 10.76.100.12 encapsulation mpls
Device(config-vfi)# exit
Device(config)# spanning-tree mode mst
Device(config)# spanning-tree mst configuration
Device(config-mst)# name cisco
Device(config-mst)# revision 11
Device(config-mst)# instance 1 vlan 100
Device(config-mst)# end
```

The following is sample output from the **show spanning-tree mst** command:

```
Device# show spanning-tree mst 1

##### MST1    vlans mapped:   100
Bridge        address 0023.3380.f8bb  priority      4097  (4096 sysid 1)
Root          this switch for MST1                              // Root for MST instance
 1 with VLAN 100
Interface                       Role Sts Cost      Prio.Nbr Type
-----------                  --- -- ------ ---- ------------------------------
Gi1/0/0                         Desg FWD 20000     128.18   P2p   // Access interface
VPLS-MST                        Desg FWD 1         128.28   Shr   // Forward VFI
```

The following is sample output from the **show spanning-tree mst detail** command:

```
Device# show spanning-tree mst 1 detail

##### MST1    vlans mapped:   100
Bridge        address 0023.3380.f8bb  priority      4097  (4096 sysid 1)
Root          this switch for MST1            // Root for MST instance 1 with VLAN 100
GigabitEthernet1/0/0 of MST1 is designated forwarding
Port info             port id         128.18  priority    128  cost       20000
```

```
Designated root       address 0023.3380.f8bb  priority   4097  cost           0
Designated bridge     address 0023.3380.f8bb  priority   4097  port id   128.18
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 40, received 5
VPLS-4000 of MST1 is designated forwarding
Port info             port id         128.28  priority    128  cost           1
Designated root       address 0023.3380.f8bb  priority   4097  cost           0
Designated bridge     address 0023.3380.f8bb  priority   4097  port id   128.28
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 28, received 26        // BPDU message exchange between N-PE devices
```

Example: Configuring MST on VPLS N-PE Devices using the commands associated with the L2VPN Protocol-Based CLIs feature

```
Device> enable
Device# configure terminal
Device(config)# l2vpn vfi context vpls-mst
Device(config-vfi)# vpn id 4000
Device(config-vfi)# forward permit l2protocol all
Device(config-vfi)# member 10.76.100.12 encapsulation mpls
Device(config-vfi)# exit
Device(config)# spanning-tree mode mst
Device(config)# spanning-tree mst configuration
Device(config-mst)# name cisco
Device(config-mst)# revision 11
Device(config-mst)# instance 1 vlan 100
Device(config-mst)# end
```

The following is sample output from the **show spanning-tree mst** command:

```
Device# show spanning-tree mst 1

##### MST1    vlans mapped:   100
Bridge        address 0023.3380.f8bb  priority      4097  (4096 sysid 1)
Root          this switch for MST1                            // Root for MST instance
 1 with VLAN 100
Interface                     Role Sts Cost      Prio.Nbr Type
------------                --- -- ------ ---- --------------------------------
Gi1/0/0                       Desg FWD 20000     128.18   P2p   // Access interface
VPLS-MST                      Desg FWD 1         128.28   Shr   // Forward VFI
```

The following is sample output from the **show spanning-tree mst detail** command:

```
Device# show spanning-tree mst 1 detail

##### MST1    vlans mapped:   100
Bridge        address 0023.3380.f8bb  priority      4097  (4096 sysid 1)
Root          this switch for MST1            // Root for MST instance 1 with VLAN 100
GigabitEthernet1/0/0 of MST1 is designated forwarding
Port info             port id         128.18  priority    128  cost       20000
Designated root       address 0023.3380.f8bb  priority   4097  cost           0
Designated bridge     address 0023.3380.f8bb  priority   4097  port id   128.18
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 40, received 5
VPLS-4000 of MST1 is designated forwarding
Port info             port id         128.28  priority    128  cost           1
Designated root       address 0023.3380.f8bb  priority   4097  cost           0
Designated bridge     address 0023.3380.f8bb  priority   4097  port id   128.28
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 28, received 26        // BPDU message exchange between N-PE devices
```

Example: BGP VPLS Autodiscovery Support on Route Reflector

In the following example, a host named PE-RR (indicating Provider Edge-Route Reflector) is configured as a route reflector that is capable of reflecting Virtual Private LAN Service (VPLS) prefixes. The VPLS address family is configured using the **address-family l2vpn vpls** command.

```
hostname PE-RR
!
router bgp 1
 bgp router-id 10.1.1.3
 no bgp default route-target filter
 bgp log-neighbor-changes
neighbor iBGP-PEERS peer-group
neighbor iBGP-PEERS remote-as 1
neighbor iBGP-PEERS update-source Loopback1
neighbor 10.1.1.1 peer-group iBGP-PEERS
neighbor 10.1.1.2 peer-group iBGP-PEERS
!
address-family l2vpn vpls
  neighbor iBGP-PEERS send-community extended
  neighbor iBGP-PEERS route-reflector-client
  neighbor 10.1.1.1 peer-group iBGP-PEERS
  neighbor 10.1.1.2 peer-group iBGP-PEERS
exit-address-family
```

Additional References for VPLS Autodiscovery BGP Based

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
MPLS commands	Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2vpn-signaling-08.txt	*Provisioning, Autodiscovery, and Signaling in L2VPNs*
draft-ietf-l2vpn-vpls-bgp-08.8	*Virtual Private LAN Service (VPLS) Using BGP for Autodiscovery and Signaling*
draft-ietf-mpls-lsp-ping-03.txt	*Detecting MPLS Data Plane Failures*
draft-ietf-pwe3-vccv-01.txt	*Pseudo-Wire (PW) Virtual Circuit Connection Verification (VCCV)*
RFC 3916	*Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3)*
RFC 3981	*Pseudo Wire Emulation Edge-to-Edge Architecture*
RFC 6074	Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)

Standard/RFC	Title
RFC 4761	Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling

MIBs

MIB	MIBs Link
• CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB) • CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB) • CISCO-IETF-PW-FR-MIB (PW-FR-MIB) • CISCO-IETF-PW-MIB (PW-MIB) • CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

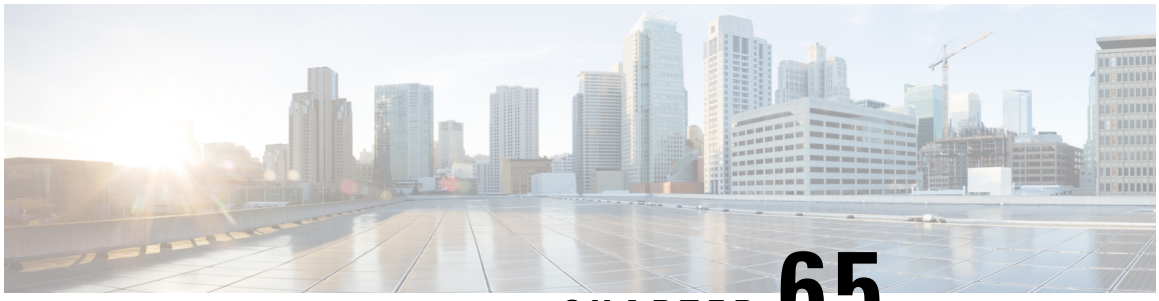
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for VPLS Autodiscovery BGP Based

Table 131: Feature Information for VPLS Autodiscovery BGP Based

Feature Name	Releases	Feature Information
VPLS Autodiscovery BGP Based	Cisco IOS XE Release 3.7S Cisco IOS Release 15.1(1)SY	VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain.



CHAPTER 65

N:1 PVC Mapping to PWE with Nonunique VPIs

The N:1 PVC Mapping to PseudoWire Emulation (PWE) with Nonunique virtual path identifiers (VPIs) feature maps one or more ATM permanent virtual circuits (PVCs) to a single pseudowire (PW). There are two modes of AAL0 encapsulation, N:1 and 1:1 mapping. In N:1 mapping, multiple unrelated virtual path identifier/virtual channel identifier (VPI/VCI) are carried over a single Multiprotocol Label Switching (MPLS) PW. This is an efficient mapping method because less resources are used from the MPLS network. In 1:1 mapping, a single VPI/VCI is carried over a single MPLS PW. Benefits of this feature include the following:

- Aggregate quality of service (QoS) can be applied to related PVCs.
- Bandwidth is conserved with the reduction in the number of pseudowires that are used.

Restrictions for N:1 PVC Mapping to PWE with Nonunique VPIs

- N:1 permanent virtual circuits (PVC) mapping configuration is supported only on multipoint subinterfaces; it is not supported on main interfaces or point-to-point subinterfaces.
- N:1 PVC mapping mode is not supported on Access Circuit Redundancy subinterfaces.
- Preconfigured PVCs cannot exist on the multipoint subinterface on which you want to configure N:1 PVC mapping.
- An attachment circuit that has been bound to a pseudowire cannot be removed unless all Layer 2 virtual circuits (VCs) have been removed.
- Layer 3 PVCs cannot be configured on N:1 subinterfaces.
- Cell packing values configured under a VC class attached to the PVC, main interface, or subinterface will not be inherited by N:1 PVCs.
- Operation, Administration, and Maintenance (OAM) functionality is not supported on N:1 Layer 2 PVCs. OAM cells coming from the customer edge (CE) network will be treated as normal data traffic and will traverse through the pseudowire.

- Only ATM adaptation layer type 0 (AAL0) encapsulation is supported for N:1 PVCs.
- The service policy configuration can be configured only at the subinterface level for N:1 PVCs.

Information About N:1 PVC Mapping to PWE with Nonunique VPIs

N:1 PVC Mapping to PWE with Nonunique VPIs Feature Description

To transport ATM cells over Multiprotocol Label Switching (MPLS), a VC is established between the provider edge (PE) routers on both ends of the MPLS backbone. With the N:1 permanent virtual circuit (PVC) Mapping to PseudoWire Emulation (PWE) with Nonunique VPIs feature, multiple PVCs irrespective of their Virtual Path Identifiers (VPIs), are transported over a single pseudowire configured on a subinterface. ("N:1" refers to the number of PVCs transported over one pseudowire). ATM cells are packed together in a single frame and sent over the single pseudowire. The ATM cell header information is packed together with the cell payload on a per-cell basis in the packets so that packets received at the egress end are unpacked and the ATM cells are mapped to the respective PVCs.

In N:1 PVC mapping mode, the device can pack cells only from a single PVC in an MPLS packet to transmit over a pseudowire; cells from multiple PVCs cannot be packed in a single MPLS packet and mapped to a single pseudowire for transmission. However, if a device receives an MPLS packet that is packed with cells from multiple PVCs, then those cells will be unpacked and sent to the respective PVCs.

How to Configure N:1 PVC Mapping to PWE with Nonunique VPIs

Configuring N:1 PVC Mapping to PWE with Nonunique VPIs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/subslot/port*
4. **atm mcpt-timers** *timer1 timer2 timer3*
5. **exit**
6. **configure terminal**
7. **interface atm** *slot/subslot/port.subslot* **multipoint**
8. **no ip address**
9. **atm enable-ilmi-trap**
10. **cell-packing** *maxcells* **mcpt-timer** *timer-number*
11. **xconnect** *peer-ipaddress vc-id* **encapsulation mpls**
12. **pvc** *vpi/vci* **l2transport**

13. Repeat Step 12 for the number of PVCs that you want to configure.
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot/subslot/port* **Example:** `Device(config)# interface atm 9/1/1`	Enables the ATM interface and enters interface configuration mode.
Step 4	**atm mcpt-timers** *timer1 timer2 timer3* **Example:** `Device(config-if)# atm mcpt-timers 100 200 300`	Sets the Maximum Cell Packing Timeout (MCPT) values in microseconds. • The MCPT timer sets the time for which the device waits for the raw cells (AAL0 encapsulation) to be packed into a single packet for punting to the pseudowire.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 7	**interface atm** *slot/subslot/port.subslot* **multipoint** **Example:** `Device(config)# interface atm 9/1/1.1 multipoint`	Enters subinterface configuration mode and creates a multipoint subinterface on the given port on the specified ATM Shared Port Adapter (SPA).
Step 8	**no ip address** **Example:** `Device(config-subif)# no ip address`	Removes the interface IP address.
Step 9	**atm enable-ilmi-trap** **Example:** `Device(config-subif)# atm enable-ilmi-trap`	Generates an Integrated Local Management Interface (ILMI) atmfVccChange trap when an ATM interface or subinterface is enabled or shut down.

	Command or Action	Purpose
Step 10	**cell-packing** *maxcells* **mcpt-timer** *timer-number* **Example:** `Device(config-subif)# cell-packing 20 mcpt-timer 2`	Enables ATM over MPLS to pack multiple ATM cells into each MPLS packet within the MCPT timing.
Step 11	**xconnect** *peer-ipaddress vc-id* **encapsulation mpls** **Example:** `Device(config-subif)# xconnect 10.1.1.1 100 encapsulation mpls`	(Optional) Enables the attachment circuit and specifies the IP address of the peer, a VC ID, and the data encapsulation method.
Step 12	**pvc** *vpi/vci* **l2transport** **Example:** `Device(config-subif)# pvc 10/100 l2transport`	Assigns a VPI and virtual channel identifier (VCI).
Step 13	Repeat Step 12 for the number of PVCs that you want to configure.	—
Step 14	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

Configuring N:1 PVC Mapping to PWE with Nonunique VPIs using the commands associated with the L2VPN Protocol-Based CLIs feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *slot/subslot/port*
4. **atm mcpt-timers** *timer1 timer2 timer3*
5. **exit**
6. **configure terminal**
7. **interface atm** *slot/subslot/portt.subslot* **multipoint**
8. **no ip address**
9. **atm enable-ilmi-trap**
10. **cell-packing** *maxcells* **mcpt-timer** *timer-number*
11. **end**
12. **interface pseudowire** *number*
13. **encapsulation mpls**
14. **neighbor** *peer-address vcid-value*
15. **exit**
16. **l2vpn xconnect context** *context-name*
17. **member pseudowire** *interface-number*

18. **member gigabitethernet** *interface-number*
19. **end**
20. **pvc** *vpi*/*vci* **l2transport**
21. Repeat Step 12 for the number of PVCs that you want to configure.
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *slot*/*subslot*/*port* **Example:** `Device(config)# interface atm 9/1/1`	Enables the ATM interface and enters interface configuration mode.
Step 4	**atm mcpt-timers** *timer1 timer2 timer3* **Example:** `Device(config-if)# atm mcpt-timers 100 200 300`	Sets the Maximum Cell Packing Timeout (MCPT) values in microseconds. • The MCPT timer sets the time for which the device waits for the raw cells (AAL0 encapsulation) to be packed into a single packet for punting to the pseudowire.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 7	**interface atm** *slot*/*subslot*/*portt.subslot* **multipoint** **Example:** `Device(config)# interface atm 9/1/1.1 multipoint`	Enters subinterface configuration mode and creates a multipoint subinterface on the given port on the specified ATM Shared Port Adapter (SPA).
Step 8	**no ip address** **Example:** `Device(config-subif)# no ip address`	Removes the interface IP address.

	Command or Action	Purpose
Step 9	**atm enable-ilmi-trap** **Example:** `Device(config-subif)# atm enable-ilmi-trap`	Generates an Integrated Local Management Interface (ILMI) atmfVccChange trap when an ATM interface or subinterface is enabled or shut down.
Step 10	**cell-packing** *maxcells* **mcpt-timer** *timer-number* **Example:** `Device(config-subif)# cell-packing 20 mcpt-timer 2`	Enables ATM over MPLS to pack multiple ATM cells into each MPLS packet within the MCPT timing.
Step 11	**end** **Example:** `Router(config-subif)# end`	Exits to privileged EXEC mode.
Step 12	**interface pseudowire** *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 13	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method.
Step 14	**neighbor** *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 10.1.1.1 100`	Specifies the peer IP address and virtual circuit (VC) ID value of the Layer 2 VPN (L2VPN) pseudowire.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 16	**l2vpn xconnect context** *context-name* **Example:** `Router(config)# l2vpn xconnect context con1`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 17	**member pseudowire** *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 18	**member gigabitethernet** *interface-number* **Example:**	Specifies the location of the Gigabit Ethernet member interface.

	Command or Action	Purpose
	`Router(config-xconnect)# member GigabitEthernet0/0/0.1`	
Step 19	**end** **Example:** `Router(config-xconnect)# end`	Exits to privileged EXEC mode.
Step 20	**pvc** *vpi*/*vci* **l2transport** **Example:** `Device(config-subif)# pvc 10/100 l2transport`	Assigns a VPI and virtual channel identifier (VCI).
Step 21	Repeat Step 12 for the number of PVCs that you want to configure.	—
Step 22	**end** **Example:** `Device(config-subif)# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

Configuration Examples for N:1 PVC Mapping to PWE with Nonunique VPIs

Example: Configuring N:1 PVC Mapping to PWE with Nonunique VPIs

The following example shows how to configure the N:1 ATM permanent virtual circuit (PVC) mapping to pseudowires with non unique virtual path identifiers (VPIs):

```
Device> enable
Device# configure terminal
Device(config)# interface atm 9/1/1
Device(config-if)# atm mcpt-timers 500 5000 50000
Device(config-if)# exit
Device# configure terminal
Device(config)# interface atm 9/1/1.1 multipoint
Device(config-subif)# no ip address
Device(config-subif)# atm enable-ilmi-trap
Device(config-subif)# cell packing 20 mcpt-timer 2
Device(config-subif)# xconnect 10.1.1.1 100 encapsulation mpls
Device(config-subif)# pvc 10/100 l2transport
Device(config-subif)# pvc 11/122 l2transport
Device(config-subif)# pvc 19/231 l2transport
Device(config-subif)# end
```

Example: Configuring N:1 PVC Mapping to PWE with Nonunique VPIs using the commands associated with the L2VPN Protocol-Based CLIs feature

The following example shows how to configure the N:1 ATM permanent virtual circuit (PVC) mapping to pseudowires with non unique virtual path identifiers (VPIs):

```
Device> enable
Device# configure terminal
Device(config)# interface atm 9/1/1
Device(config-if)# atm mcpt-timers 500 5000 50000
Device(config-if)# exit
Device(config)# configure terminal
Device(config)# interface atm 9/1/1.1 multipoint
Device(config-subif)# no ip address
Device(config-subif)# atm enable-ilmi-trap
Device(config-subif)# cell packing 20 mcpt-timer 2
Device(config-subif)# exit
Device(config)#interface pseudowire 100
Device(config-if)#encapsulation mpls
Device(config-if)#neighbor 10.1.1.1 100
Device(config-if)# pvc 10/100 l2transport
Device(config-if)# pvc 11/122 l2transport
Device(config-if)# pvc 19/231 l2transport
Device(config-if)# exit
Device(config)#l2vpn xconnect context A
Router(config-xconnect)#member pseudowire 100
Device(config-xconnect)#member atm 9/1/1
Device(config-xconnect)# end
```

Additional References

Related Documents

Related Topic	Document Title
ATM commands	Asynchronous Transfer Mode Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

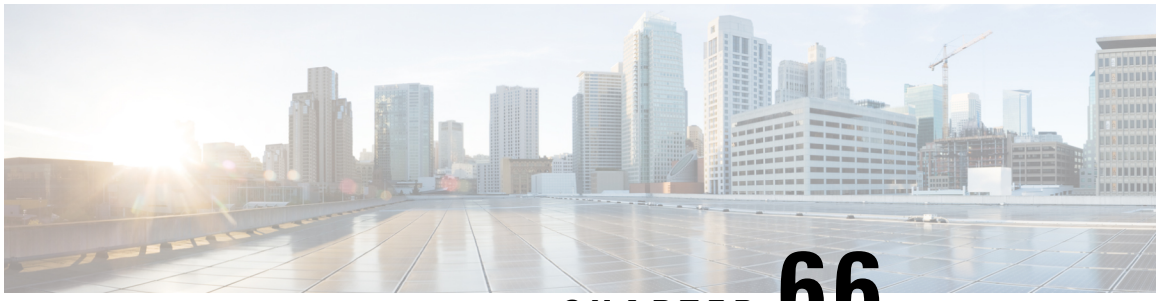
Feature Information for N:1 PVC Mapping to PWE with Nonunique VPIs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 132: Feature Information for N:1 PVC Mapping to PWE with Nonunique VPIs

Feature Name	Releases	Feature Information
N:1 PVC Mapping to PWE with Nonunique VPIs	Cisco IOS XE Release 3.7S	The N:1 PVC Mapping to PWE with Nonunique VPIs feature maps one or more ATM PVCs to a single pseudowire. In Cisco IOS XE Release 3.7S, support was added for Cisco ASR 903 Routers. The following command was introduced by this feature: **show atm cell-packaging** .



CHAPTER 66

QoS Policies for VFI Pseudowires

Restrictions for QoS Policies for VFI Pseudowires

- A maximum of 32K pseudowires.
- A maximum of 4K unique policy maps.
- A maximum of 128 neighbors per VFI context.

Information About QoS Policies for VFI Pseudowires

QoS Policies for VFI Pseudowires

QoS policies are specified on individual pseudowire interfaces and are applied only to the corresponding pseudowires. It is possible to specify different QoS policies on different pseudowire members of the same virtual forwarding interface (VFI) or on the subset of the pseudowires. There may be one or more pseudowires configured per VFI. Both manually configured and auto discovered pseudowire configurations are supported.

QoS policies are specified using a pseudowire template. The template can be applied on multiple pseudowires of the same, or different, VFIs. All those pseudowires get the same QoS policy applied as specified in the template. For auto-discovered pseudowires, QoS policies can only be specified using a pseudowire template.

The QoS Policies for VFI Pseudowires feature supports both ingress and egress policies and traffic classification can be done based on different match criteria.

How to Configure QoS Policies for VFI Pseudowires

Configuring QoS Policies for Pseudowires

Perform this task to configure QoS policies for pseudowires.

Before you begin

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **priority** *bandwidth-kbps*
6. **exit**
7. **class** *class-map-name*
8. **bandwidth percent** *percentage*
9. **exit**
10. **class** *class-map-name*
11. **police cir** *bps*
12. **exit**
13. **class** *class-map-name*
14. **shape average** *bps*
15. **queue-limit** *queue-limit size* **packets**
16. **random-detect**
17. **exit**
18. **exit**
19. **policy-map** *policy-map-name*
20. **class** *class-map-name*
21. **shape average** *bps*
22. **service-policy** *policy-map*
23. **exit**
24. **exit**
25. **policy-map** *policy-map-name*
26. **class** *class-map-name*
27. **shape average** *bps*
28. **exit**
29. **exit**
30. **policy-map** *policy-map-name*
31. **class** *class-map-name*
32. **shape average** *bps*
33. **exit**
34. **exit**

35. **exit policy-map** *policy-map-name*
36. **class** *class-map-name*
37. **shape average** *bps*
38. **exit**
39. **exit**
40. **policy-map** *policy-map-name*
41. **class** *class-map-name*
42. **police** *bps*
43. **interface pseudowire** *number*
44. **encap mpls**
45. **neighbor** *peer-address vcid-value*
46. **service-policy input** *policy-map-name*
47. **service-policy output** *policy-map-name*
48. **interface gigabit ethernet** *number*
49. **service-policy output** *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Device# policy-map gold-policy-child`	Creates a policy map to specify a service policy.
Step 4	**class** *class-map-name* **Example:** `Device(config-pmap)# class priority-class`	Specifies the name of the class map.
Step 5	**priority** *bandwidth-kbps* **Example:** `Device(config-pmap-c)# priority 100`	Gives priority to a class of traffic belonging to a policy map.
Step 6	**exit** **Example:**	Exits policy-map class configuration mode.

	Command or Action	Purpose
	`Device(config-pmap-c)# exit`	
Step 7	**class** *class-map-name* **Example:** `Device(config-pmap-c)# class guarantee-class`	Specifies the name of the class map.
Step 8	**bandwidth percent** *percentage* **Example:** `Device(config-pmap-c)# bandwidth percent 50`	Specifies or modifies the bandwidth allocated for a class belonging to a policy map.
Step 9	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 10	**class** *class-map-name* **Example:** `Device(config-pmap-c)# class limited-class`	Specifies the name of the class map.
Step 11	**police cir** *bps* **Example:** `Device(config-pmap-c)# police cir 8000`	Creates a per-interface policer and configures the policy-map class to use it.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 13	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 14	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 8000`	Shapes traffic to the indicated bit rate.
Step 15	**queue-limit** *queue-limit size* **packets** **Example:** `Device(config-pmap-c)# queue-limit 150 packets`	Specifies the queue limit size for a class.

	Command or Action	Purpose
Step 16	**random-detect** **Example:** `Device(config-pmap-c)# andom-detect`	Configures Weighted Random Early Detection (WRED) for a class in a policy map.
Step 17	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 18	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.
Step 19	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map gold-policy-hqos`	Creates a policy map to specify a service policy.
Step 20	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 21	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 10000`	Shapes traffic to the indicated bit rate.
Step 22	**service-policy** *policy-map* **Example:** `Device(config-pmap-c)# service-policy gold-policy-child`	Attaches a policy map to a class.
Step 23	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 24	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.

	Command or Action	Purpose
Step 25	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map pw-shaper`	Creates a policy map to specify a service policy.
Step 26	**class** *class-map-name* **Example:** `Device(config-pmap)#class class-default`	Specifies the name of the class map.
Step 27	**shape average** *bps* **Example:** `Device(config-pmap-c)#shape average 20000`	Shapes traffic to the indicated bit rate.
Step 28	**exit** **Example:** `Device(config-pmap-c)#exit`	Exits policy-map class configuration mode.
Step 29	**exit** **Example:** `Device(config-pmap)#exit`	Exits policy-map configuration mode.
Step 30	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map sub-ifc-shaper`	Creates a policy map to specify a service policy.
Step 31	**class** *class-map-name* **Example:** `Device(config-pmap)#class class-default`	Specifies the name of the class map.
Step 32	**shape average** *bps* **Example:** `Device(config-pmap-c)#shape average 40000`	Shapes traffic to the indicated bit rate.
Step 33	**exit** **Example:** `Device(config-pmap-c)#exit`	Exits policy-map class configuration mode.
Step 34	**exit** **Example:**	Exits policy-map configuration mode.

	Command or Action	Purpose
	`Device(config-pmap)#exit`	
Step 35	**exit policy-map** *policy-map-name* **Example:** `Device(config)# policy-map port-shaper`	Creates a policy map to specify a service policy.
Step 36	**class** *class-map-name* **Example:** `Device(config-pmap)#class class-default`	Specifies the name of the class map.
Step 37	**shape average** *bps* **Example:** `Device(config-pmap-c)#shape average 60000`	Shapes traffic to the indicated bit rate.
Step 38	**exit** **Example:** `Device(config-pmap-c)#exit`	Exits policy-map class configuration mode.
Step 39	**exit** **Example:** `Device(config-pmap)#exit`	Exits policy-map configuration mode.
Step 40	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map ingress-police`	Creates a policy map to specify a service policy.
Step 41	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	
Step 42	**police** *bps* **Example:** `Device(config-pmap-c)# police 10000`	Creates a per-interface policer and configures the policy-map class to use it.
Step 43	**interface pseudowire** *number* **Example:** `Device(config-pmap-c-police)# interface pseudowire 1`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 44	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 45	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)# neighbor 10.0.0.1 100`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 46	**service-policy input** *policy-map-name* **Example:** `Device(config-if)# service-policy input ingress-policy`	Attaches a policy map to an input interface.
Step 47	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output gold-policy-hqos`	Attaches a policy map to an output interface.
Step 48	**interface gigabit ethernet** *number* **Example:** `Device(config-if)# interface gigabitethernet 1/1/0`	Configures an interface type.
Step 49	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output port-shaper`	Attaches a policy map to an output interface.

Creating a Hierarchical Policy for VFI Pseudowires

Perform this task to create a hierarchical policy for VFI Pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **shape average** *bps*
6. **service-policy** *policy-map*
7. **exit**

8. **exit**
9. **policy-map** *policy-map-name*
10. **class** *class-map-name*
11. **shape average** *bps*
12. **exit**
13. **exit**
14. **policy-map** *policy-map-name*
15. **class** *class-map-name*
16. **shape average** *bps*
17. **exit**
18. **exit**
19. **exit policy-map** *policy-map-name*
20. **class** *class-map-name*
21. **shape average** *bps*
22. **exit**
23. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map gold-policy-hqos`	Creates a policy map to specify a service policy.
Step 4	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 5	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 10000`	Shapes traffic to the indicated bit rate.
Step 6	**service-policy** *policy-map* **Example:**	Attaches a policy map to a class.

	Command or Action	Purpose
	`Device(config-pmap-c)# service-policy gold-policy-child`	
Step 7	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 8	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.
Step 9	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map pw-shaper`	Creates a policy map to specify a service policy.
Step 10	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 11	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 20000`	Shapes traffic to the indicated bit rate.
Step 12	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 13	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.
Step 14	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map sub-ifc-shaper`	Creates a policy map to specify a service policy.
Step 15	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.

	Command or Action	Purpose
Step 16	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 40000`	Shapes traffic to the indicated bit rate.
Step 17	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 18	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.
Step 19	**exit policy-map** *policy-map-name* **Example:** `Device(config)# policy-map port-shaper`	Creates a policy map to specify a service policy.
Step 20	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 21	**shape average** *bps* **Example:** `Device(config-pmap-c)# shape average 60000`	Shapes traffic to the indicated bit rate.
Step 22	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 23	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode.

Attaching a Policy Map to a VFI Pseudowire

Perform this task to attach a policy map to a VFI Pseudowire.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **police** *bps*
6. **interface pseudowire** *number*
7. **encap mpls**
8. **neighbor** *peer-address vcid-value*
9. **service-policy input** *policy-map-name*
10. **service-policy output** *policy-map-name*
11. **interface gigabit ethernet** *number*
12. **service-policy output** *policy-map-name*
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Device# policy-map ingress-police`	Creates a policy map to specify a service policy.
Step 4	**class** *class-map-name* **Example:** `Device(config-pmap)# class class-default`	Specifies the name of the class map.
Step 5	**police** *bps* **Example:** `Device(config-pmap-c)# police 10000`	Creates a per-interface policer and configures the policy-map class to use it.
Step 6	**interface pseudowire** *number* **Example:** `Device(config-pmap-c-police)# interface pseudowire 1`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 8	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)# neighbor 10.0.0.1 100`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.
Step 9	**service-policy input** *policy-map-name* **Example:** `Device(config-if)# service-policy input ingress-policy`	Attaches a policy map to an input interface.
Step 10	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output gold-policy-hqos`	Attaches a policy map to an output interface.
Step 11	**interface gigabit ethernet** *number* **Example:** `Device(config-if)# interface gigabit ethernet 1/1/0`	Configures an interface type.
Step 12	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output port-shaper`	Attaches a policy map to an output interface.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Configuring VFI with Two Pseudowire Members with Different QoS Policies

Perform this task to configure VFI with two pseudowire members with different QoS policies.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface pseudowire** *number*
4. **encap mpls**
5. **neighbor** *peer-address vcid value*
6. **service-policy output** *policy-map-name*
7. **interface pseudowire** *number*
8. **encap mpls**
9. **neighbor** *peer-address vcid value*
10. **service-policy output** *policy-map-name*
11. **l2vpn vfi context** *name*
12. **vpn id** *vpn-id*
13. **member pseudowire** *pw-int-number*
14. **member pseudowire** *pw-int-number*
15. **bridge-domain** *bridge-domain-id*
16. **member** *interface-type-number*
17. **interface BDI** *number*
18. **ip vrf forwarding** *vrf-name*
19. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface pseudowire** *number* **Example:** `Device# interface pseudowire 1`	Configures an interface type and enters interface configuration mode.
Step 4	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 5	**neighbor** *peer-address vcid value* **Example:** `Device(config-if)# neighbor 10.0.0.1 100`	Specifies the peer IP address and virtual circuit (VC) ID value of an L2VPN pseudowire.

	Command or Action	Purpose
Step 6	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output gold-policy`	Attaches a policy map to an output interface.
Step 7	**interface pseudowire** *number* **Example:** `Device(config-if)# interface pseudowire 2`	Configures an interface type.
Step 8	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 9	**neighbor** *peer-address vcid value* **Example:** `Device(config-if)# neighbor 20.0.0.1 100`	Specifies the peer IP address and VCID of an L2VPN pseudowire.
Step 10	**service-policy output** *policy-map-name* **Example:** `Device(config-if)# service-policy output silver-policy`	Attaches a policy map to an output interface.
Step 11	**l2vpn vfi context** *name* **Example:** `Device(config-if)# l2vpn vfi context my-vfi`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks.
Step 12	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Sets a VPN ID on a Virtual Private LAN Services (VPLS) instance.
Step 13	**member pseudowire** *pw-int-number* **Example:** `Device(config-vfi)# member pseudowire 1`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.
Step 14	**member pseudowire** *pw-int-number* **Example:** `Device(config-vfi)# member pseudowire 2`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.

	Command or Action	Purpose
Step 15	**bridge-domain** *bridge-domain-id* **Example:** `Device(config-vfi)# bridge-domain 100`	Configures components on a bridge domain.
Step 16	**member** *interface-type-number* **Example:** `Device(config-bdomain)# member vfi my-vfi`	Binds a service instance to a bridge domain instance.
Step 17	**interface BDI** *number* **Example:** `Device(config-bdomain)# interface BDI 100`	Configures an interface type and enters interface configuration mode.
Step 18	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding MY-VRF`	Associates a Virtual Private Network (VPN) routing and forwarding (VRF) instance with an interface or subinterface.
Step 19	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 30.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

Configuring VFI with Two Pseudowire Members with the Same QoS Policy

Perform this task to configure VFI with two pseudowire members with the same QoS policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** *name*
4. **encap mpls**
5. **service-policy output** *policy-map-name*
6. **interface pseudowire** *number*
7. **encap mpls**
8. **neighbor** *peer-address vcid value*
9. **source template type pseudowire** *template-name*
10. **interface pseudowire** *number*
11. **encap mpls**
12. **neighbor** *peer-address vcid value*
13. **source template type pseudowire** *template-name*
14. **l2vpn vfi context** *name*
15. **vpn id** *vpn-id*

16. **member pseudowire** *pw-int-number*
17. **member pseudowire** *pw-int-number*
18. **bridge-domain** *bridge-domain-id*
19. **member** *interface-type-number*
20. **interface BDI** *number*
21. **ip vrf forwarding** *vrf-name*
22. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire my_template`	Configures a template.
Step 4	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 5	**service-policy output** *policy-map-name* **Example:** `Device(config-template)# service-policy output common-policy`	Attaches a policy map to a output interface.
Step 6	**interface pseudowire** *number* **Example:** `Device(config-if)# interface pseudowire 1`	Configures an interface type.
Step 7	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.

	Command or Action	Purpose
Step 8	**neighbor** *peer-address vcid value* **Example:** `Device(config-if)# neighbor 10.0.0.1 100`	Specifies the peer IP address and VCID of an L2VPN pseudowire.
Step 9	**source template type pseudowire** *template-name* **Example:** `Device(config-if)# source template type pseudowire my_template`	Configures the name of a source template of type pseudowire.
Step 10	**interface pseudowire** *number* **Example:** `Device(config-if)# interface pseudowire 2`	Configures an interface type.
Step 11	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 12	**neighbor** *peer-address vcid value* **Example:** `Device(config-if)# neighbor 20.0.0.1 100`	Specifies the peer IP address and VCID of an L2VPN pseudowire.
Step 13	**source template type pseudowire** *template-name* **Example:** `Device(config-if)# source template type pseudowire my_template`	Configures the name of a source template of type pseudowire.
Step 14	**l2vpn vfi context** *name* **Example:** `Device(config-if)# l2vpn vfi context my-vfi`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks.
Step 15	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Sets a VPN ID on a Virtual Private LAN Services (VPLS) instance.
Step 16	**member pseudowire** *pw-int-number* **Example:** `Device(config-vfi)# member pseudowire 1`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.

	Command or Action	Purpose
Step 17	**member pseudowire** *pw-int-number* **Example:** `Device(config-vfi)# member pseudowire 2`	Specifies the devices that form a point-to-point Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) connection.
Step 18	**bridge-domain** *bridge-domain-id* **Example:** `Device(config-vfi)# bridge-domain 100`	Configures components on a bridge domain.
Step 19	**member** *interface-type-number* **Example:** `Device(config-bdomain)# member vfi my-vfi`	Binds a service instance to a bridge domain instance.
Step 20	**interface BDI** *number* **Example:** `Device(config-bdomain)# interface BDI 100`	Configures an interface type and enters interface configuration mode.
Step 21	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding MY-VRF`	Associates a Virtual Private Network (VPN) routing and forwarding (VRF) instance with an interface or subinterface.
Step 22	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 30.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

Configuring VFI with Auto Discovered Pseudowires

Perform this task to configure VFI with auto discovered pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template type pseudowire** *name*
4. **encap mpls**
5. **service-policy output** *policy-map-name*
6. **l2vpn vfi context** *name*
7. **vpn id** *vpn-id*
8. **autodiscovery bgp signaling ldp template** *template-name*
9. **bridge-domain** *bridge-domain-id*
10. **member** *interface-type-number*

11. **interface BDI** *number*
12. **ip vrf forwarding** *vrf-name*
13. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. **Note** Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire my_template`	Configures a template.
Step 4	**encap mpls** **Example:** `Device(config-if)# encap mpls`	Configures MPLS encapsulation.
Step 5	**service-policy output** *policy-map-name* **Example:** `Device(config-template)# service-policy output common-policy`	Attaches a policy map to a output interface.
Step 6	**l2vpn vfi context** *name* **Example:** `Device(config-if)# l2vpn vfi context my-vfi`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) between two or more separate networks.
Step 7	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Sets a VPN ID on a Virtual Private LAN Services (VPLS) instance.
Step 8	**autodiscovery bgp signaling ldp template** *template-name* **Example:**	Designates a Layer 2 virtual forwarding interface (VFI) as having Label Distribution Protocol (LDP) autodiscovered pseudowire members.

	Command or Action	Purpose
	`Device(config-vfi)# autodiscovery bgp signaling ldp template my_template`	
Step 9	**bridge-domain** *bridge-domain-id* **Example:** `Device(config-vfi)# bridge-domain 100`	Configures components on a bridge domain.
Step 10	**member** *interface-type-number* **Example:** `Device(config-bdomain)# member vfi my-vfi`	Binds a service instance to a bridge domain instance.
Step 11	**interface BDI** *number* **Example:** `Device(config-bdomain)# interface BDI 100`	Configures an interface type and enters interface configuration mode.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding MY-VRF`	Associates a Virtual Private Network (VPN) routing and forwarding (VRF) instance with an interface or subinterface.
Step 13	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 30.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.

Configuration Examples for QoS Policies for VFI Pseudowires

Example: Configuring QoS Policies for Pseudowires

The following example shows how to QoS policies for pseudowires:

```
Device(config)# policy-map GOLD-POLICY-CHILD
Device(config-pmap)# class PRIORITY-CLASS
Device(config-pmap-c)# priority 100
Device(config-pmap-c)# exit
Device(config-pmap)# class GUARANTEE-CLASS
Device(config-pmap-c)# bandwidth 1000
Device(config-pmap-c)# exit
Device(config-pmap)# class LIMITED-CLASS
Device(config-pmap-c)# police cir 8000
Device(config-pmap-c-police)# class class-default
Device(config-pmap-c)# shape average 8000
Device(config-pmap-c)# queue-limit 150
Device(config-pmap-c)# random-detect
```

```
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map GOLD-POLICY-HQOS
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 8000
Device(config-pmap-c)# service-policy GOLD-POLICY-CHILD
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map PW-SHAPER
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 8000
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map SUB-IFC-SHAPER
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 10000
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map PORT-SHAPER
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average 20000
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# policy-map INGRESS-POLICE
Device(config-pmap)# class class-default
Device(config-pmap-c)# police 10000
Device(config-pmap-c-police)# interface pseudowire 1
Line protocol on Interface pseudowire0, changed state to up
Device(config-if)# encap mpls
Device(config-if)# neighbor 10.0.0.1 100
Device(config-if)# service-policy input INGRESS-POLICY
Device(config-if)# service-policy output GOLD-POLICY-HQOS
Device(config-if)# interface GigabitEthernet 1/1/0
--- Pseudowire is going out through this interface
Device(config-if)# service-policy output PORT-SHAPER
```

Example: Configuring VFI with Two Pseudowire Members with Different QoS Policies

The following example shows how to configure VFI with two pseudowire members with different QoS policies:

```
Device(config)# interface pseudowire1
Line protocol on Interface pseudowire0, changed state to up
Device(config-if)# encap mpls
Device(config-if)# neighbor 10.0.0.1 100
Device(config-if)# service-policy output GOLD-POLICY
Device(config-if)# interface pseudowire2
Device(config-if)# encap mpls
Device(config-if)# neighbor 20.0.0.1 100
Device(config-if)# service-policy output SILVER-POLICY
Device(config-if)# l2vpn vfi context MY-VFI
Device(config-vfi)# vpn id 100
Device(config-vfi)# member pseudowire1
Device(config-vfi)# member pseudowire2
Device(config-vfi)# bridge-domain 100
Device(config-bdomain)# member vfi MY-VFI
```

```
STATUS_CHANGED: Status of VFI my-vfi changed from DOWN to UP
Device(config-bdomain)# interface BDI 100
Device(config-if)# ip vrf forwarding MY-VRF
Device(config-if)# ip address 30.0.0.1 255.255.255.0
```

Example: Configuring VFI with Two Pseudowire Members with the Same QoS Policy

The following example shows how to configure VFI with two pseudowire members with the same QoS policy:

```
Device(config)# template type pseudowire MY_TEMPLATE
Device(config-template)# encapsulation mpls
Device(config-template)# service-policy output COMMON-POLICY
Device(config-template)# interface pseudowire1
Line protocol on Interface pseudowire0, changed state to up
Device(config-if)# encap mpls
Device(config-if)# neighbor 10.0.0.1 100
Device(config-if)# source template type pseudowire MY_TEMPLATE
Device(config-if)# interface pseudowire2
Device(config-if)# encap mpls
Device(config-if)# neighbor 20.0.0.1 100
Device(config-if)# source template type pseudowire MY_TEMPLATE
Device(config-if)# l2vpn vfi context MY-VFI
Device(config-vfi)# vpn id 100
Device(config-vfi)# member pseudowire1
Device(config-vfi)# member pseudowire2
Device(config-vfi)# bridge-domain 100
Device(config-bdomain)# member vfi MY-VFI
Status of VFI my-vfi changed from DOWN to UP
Device(config-bdomain)# interface BDI 100
Device(config-if)# ip vrf forwarding MY-VRF
Device(config-if)# ip address 30.0.0.1 255.255.255.0
```

Example: Configuring VFI with Auto Discovered Pseudowires

The following example shows how to configure VFI with auto discovered pseudowires:

```
Device(config)# template type pseudowire MY_TEMPLATE
Device(config-template)# encapsulation mpls
Device(config-template)# service-policy output COMMON-POLICY
Device(config-template)# l2vpn vfi context MY-VFI
Device(config-vfi)# vpn id 100
Line protocol on Interface pseudowire0, changed state to up
Device(config-vfi)# autodiscovery bgp signaling ldp template MY_TEMPLATE
Device(config-vfi-autodiscovery)# bridge-domain 100
Device(config-bdomain)# member vfi MY-VFI
Status of VFI my-vfi changed from DOWN to UP
Device(config-bdomain)# interface BDI 100
Device(config-if)# ip vrf forwarding MY-VRF
Device(config-if)# ip address 30.0.0.1 255.255.255.0
```

Example: Displaying Pseudowire Policy Map Information

The following is sample output from the **show policy-map interface** command which shows class maps and policy maps configured for the pseudowire 2 interface:

```
Device#show policy-map interface pseudowire2
 pseudowire2

 Service-policy output: pw_brr

    Class-map: prec1 (match-all)
      0 packets, 0 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 1
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 1

    Class-map: prec2 (match-all)
      0 packets, 0 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 2
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 2

    Class-map: prec3 (match-all)
      0 packets, 0 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 3
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 3

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: any
      Queueing
      queue limit 4166 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth remaining ratio 4
Device#
```

Additional References for QoS Policies for VFI Pseudowires

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*
Configuring the pseudowire class	"Any Transport over MPLS"
Layer 2 VPN	• Any Transport over MPLS • L2VPN Pseudowire Switching • MPLS LSP Ping/Traceroute for LDP/TE, and LSP Ping for VCCV
L2VPN pseudowires	• L2VPN Pseudowire Redundancy • MPLS Pseudowire Status Signaling

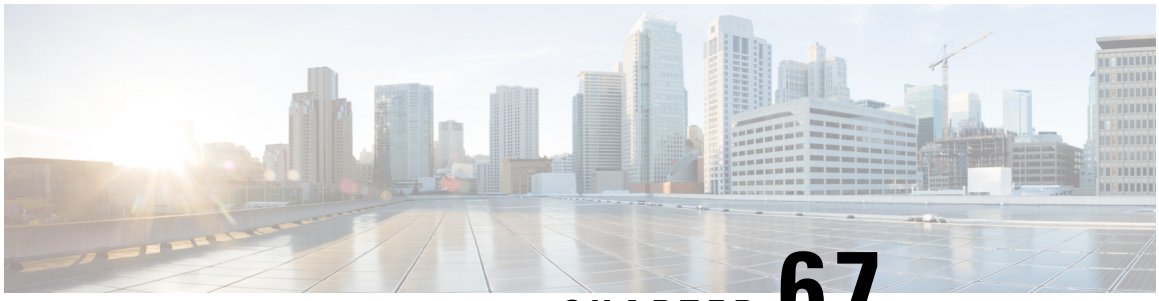
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information For QoS Policies for VFI Pseudowires

Table 133: Feature Information for QoS Policies for VFI Pseudowire

Feature Name	Releases	Feature Information
QoS Policies for VFI Pseudowires	Cisco IOS XE 3.8S	This features allows you to configure QoS classes and policies for use on VFI pseudowire members. The following commands were introduced or modified: **show policy-map interface**.



CHAPTER 67

VPLS BGP Signaling L2VPN Inter-AS Option A

The Virtual Private LAN Switching (VPLS) Border Gateway Protocol (BGP) Signaling Layer 2 Virtual Private Network (L2VPN) feature simplifies the auto-discovery and signaling of all known PE devices in a VPLS instance by using BGP.

Prerequisites for VPLS BGP Signaling L2VPN Inter-AS Option A

- The Control word must turned off for VPLS BGP signaling by using the **no control-word** command under a pseudowire class. For example:

```
Router> enable
Router# configure terminal
Router(config)# pseudowire-class my_pw_class
Router(config-pw-class)# no control-word
```

- The Route Distinguisher (RD) must match for all the virtual forwarding instances (VFIs) in a VPLS domain.

Information About VPLS BGP Signaling L2VPN Inter-AS Option A

BGP Auto-discovery and Signaling for VPLS

The Virtual Private LAN Switching (VPLS) control plane is used for auto-discovery and signaling. Auto-discovery involves locating all provider edge (PE) devices that participate in a particular VPLS instance.

Signaling is accomplished by configuring pseudowires for a VPLS instance. Prior to the introduction of the VPLS BGP Signaling L2VPN Inter-AS Option B feature, Label Distribution Protocol (LDP) was used for signaling and Border Gateway Protocol (BGP) was used for auto-discovery, as specified in RFC 6074. With the introduction of the VPLS BGP Signaling L2VPN Inter-AS Option B feature, the VPLS BGP Signaling L2VPN feature supports RFC 4761 by simplifying the auto-discovery and signaling of all known PE devices in a VPLS instance by using BGP for both functions. Auto-discovery is defined per VPLS instance.

Internal BGP (IBGP) peers exchange update messages of the L2VPN Address Family Identifier (AFI) and the Subsequent Address Family Identifier (SAFI) numbers with L2VPN information to perform both auto-discovery and signaling, which includes the Network Layer Reachability Information (NLRI).

Both BGP standards (RFC 6074 and RFC 4761) for the auto-discovery protocol for VPLS use the same BGP AFI (25) and SAFI (65) but they have different Network Layer Reachability Information (NLRI) encoding, which makes them incompatible with each other. CLI configuration is needed to distinguish the two encoding types as they are mutually exclusive per neighbor. The difference between the two BGP standards is:

- RFC 6074 provides guidelines for specifying length encoding as bits.
- RFC 4761 provides guidelines for specifying length encoding as bytes.

To detect which NLRI encoding standard is supported, the length encoding needs to be determined.

BGP L2VPN Signaling with NLRI

Network Layer Reachability Information (NLRI) enables Border Gateway Protocol (BGP) to carry supernetting information, as well as perform aggregation. Each NLRI consists of block labels that follow the structure LB, LB+1, ... , LB+VBS-1. The NLRI is exchanged between BGP devices for BGP auto-discovery with BGP signaling. The following fields are configured or auto-generated for each Virtual Private LAN Switching (VPLS) instance:

- Length (2 Octets)
- Route distinguisher (RD) is usually an auto-generated 8-byte VPN ID that can also be configured. This value must be unique for a VPLS bridge-domain (or instance).
- VPLS Endpoint ID (VEID) (2 Octets). Each PE device is configured with a VEID value.
- VPLS Endpoint Block Offset (VBO) (2 Octets).
- VPLS Endpoint Block Size (VBS) (2 Octets).
- Label Base (LB) (3 Octets).
- Extended Community Type (2 Octets) - 0x800A attributes. The Route Target (RT) specified for a VPLS instance, next-hop and other Layer 2 information is carried in this encoding. An RT-based import and export mechanism similar to L3VPN is performed by BGP to perform filtering on the L2VPN NLRIs of a particular VPLS instance.
- Encapsulation Type (1 Octet) - VPLS = 19
- Control Flags (1 Octet)
- Layer 2 Maximum Transmission Unit (MTU) (2 Octets)
- Reserved (2 Octets)

How to Configure VPLS BGP Signaling L2VPN Inter-AS Option A

Enabling BGP Auto-discovery and BGP Signaling

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices by BGP auto-discovery and BGP signaling functions announced through IBGP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *vfi-context-name*
4. **vpn id** *vpn-id*
5. **autodiscovery bgp signaling bgp**
6. **ve id** *ve-ID-number*
7. **ve range** *ve-range-number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *vfi-context-name* **Example:** `Device(config)# l2vpn vfi context vfi1`	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) for specifying core-facing pseudowires in a Virtual Private LAN Services (VPLS) and enters L2VFI configuration mode. • The VFI represents an emulated LAN or a VPLS forwarder from the VPLS architectural model when using an emulated LAN interface.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.
Step 5	**autodiscovery bgp signaling bgp** **Example:**	Enables BGP auto-discovery and BGP signaling on the device.

	Command or Action	Purpose
	`Device(config-vfi)# autodiscovery bgp signaling bgp`	
Step 6	**ve id** *ve-ID-number* **Example:** `Device(config-vfi)# ve id 1`	Configures a VPLS Endpoint ID (VEID) for the NLRI exchanged between BGP devices for BGP auto-discovery with BGP signaling. • For example, VEID numbering sequences such as 1,2,3 or 501, 502, 503 are preferred because the VEIDs are contiguous. • Avoid a non-contiguous numbering scheme such as 100, 200, 300. Repeat this step to add more VEIDs. The VEID must be unique within the same VPLS domain for all PE devices. **Note** If you change the VEID, then the virtual circuit (VC) reprovisions and traffic is impacted as a result.
Step 7	**ve range** *ve-range-number* **Example:** `Device(config-vfi)# ve range 10`	Overrides the minimum size of VPLS edge (VE) blocks. • The VE range value should be approximately the same as the number of neighbors (up to 100). • The VE range can be configured based on the number of neighboring PE devices in the network. • For example, if 50 PE devices are in a VPLS domain, then a VE range of 50 is better than 10 because the number of NLRIs exchanged are less and the convergence time is reduced. **Note** If no VE range is configured or an existing VE range value is removed, then the default VE range of 10 is applied. The default VE range should not be used if the device has many PE neighbors. **Note** If you change the VE range, then the VC reprovisions and traffic is impacted as a result.
Step 8	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. **Note** Commands take effect after the device exits L2VFI configuration mode.

Configuring BGP Signaling for VPLS Autodiscovery

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **address-family l2vpn vpls**
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
9. **neighbor** {*ip-address* | *peer-group-name*} **suppress-signaling-protocol ldp**
10. **exit-address-family**
11. Repeat steps 1 to 10 to configure and activate other BGP neighbors in an L2VPN address family.
12. **end**
13. **show l2vpn vfi**
14. **show ip bgp l2vpn vpls** {**all** [**summary**] | **rd** *route-distinguisher*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode for the specified routing process.
Step 4	**bgp graceful-restart** **Example:** `Device(config-router)# bgp graceful-restart`	Enables the Border Gateway Protocol (BGP) graceful restart capability globally for all BGP neighbors.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 198.51.100.1 remote-as 65000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor.

	Command or Action	Purpose
		• If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.
Step 6	**address-family l2vpn vpls** **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The **vpls** keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers and a L2VPN VPLS address family session is created.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 198.51.100.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Device(config-router-af)# neighbor 198.51.100.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **suppress-signaling-protocol ldp** **Example:** `Device(config-router-af)# neighbor 198.51.100.1 suppress-signaling protocol ldp`	Suppresses LDP signaling for a BGP neighbor so that BGP signaling for VPLS auto-discovery is used instead. • In this example, LDP signaling is suppressed for the neighbor at 10.10.10.1.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 11	Repeat steps 1 to 10 to configure and activate other BGP neighbors in an L2VPN address family.	
Step 12	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Step 13	**show l2vpn vfi**
	Example:
	`Device# show l2vpn vfi`
	`PE1-standby#sh l2vpn vfi`
		Displays information about the configured VFI instances.

	Command or Action	Purpose
	Load for five secs: 0%/0%; one minute: 0%; five minutes: 0% Time source is hardware calendar, *20:50:52.526 GMT Wed Aug 29 2012 Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No VFI name: VFI1, state: up, type: multipoint, signaling: BGP VPN ID: 1, VE-ID: 10, VE-SIZE: 10 RD: 1:1, RT: 1:1 Bridge-Domain 100 attachment circuits: Pseudo-port interface: pseudowire100001 Interface Peer Address VE-ID Local Label Remote Label S pseudowire100003 198.51.100.2 11 1003 2002 Y pseudowire100005 198.51.100.3 12 1004 2002 Y VFI name: VFI2, state: up, type: multipoint, signaling: BGP VPN ID: 2, VE-ID: 20, VE-SIZE: 12 RD: 1:2, RT: 1:2, import 3:3, export 4:4 Bridge-Domain 200 attachment circuits: Pseudo-port interface: pseudowire100002 Interface Peer Address VE-ID Local Label Remote Label S pseudowire100004 198.51.100.2 21 1021 2020 Y pseudowire100006 198.51.100.3 22 1022 2020 Y	
Step 14	**show ip bgp l2vpn vpls** {**all** [**summary**] | **rd** *route-distinguisher*} **Example:** Device# **show ip bgp l2vpn vpls all summary** BGP router identifier 198.51.100.1, local AS number 65000 BGP table version is 14743, main routing table version 14743 6552 network entries using 1677312 bytes of memory 6552 path entries using 838656 bytes of memory 3276/3276 BGP path/bestpath attribute entries using 760032 bytes of memory 1638 BGP extended community entries using 65520 bytes of memory 0 BGP route-map cache entries using 0 bytes of memory 0 BGP filter-list cache entries using 0 bytes of memory BGP using 3341520 total bytes of memory BGP activity 9828/3276 prefixes, 9828/3276 paths, scan interval 60 secs Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd 198.51.101.1 4 65000 90518 90507	Displays information about the L2VPN VPLS address family.

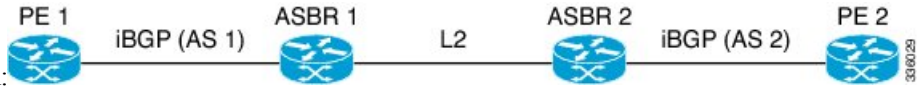
	Command or Action	Purpose
	14743 0 0 8w0d 1638 198.51.102.2 4 65000 4901 4895 14743 0 0 2d01h 1638 198.51.103.3 4 65000 4903 4895 14743 0 0 2d01h 1638	

VPLS BGP Signaling L2VPN Inter-AS Option A: Example

The following example configuration describes Inter-AS Option A for VPLS BGP signaling in an L2VPN. The Autonomous System Boundary Router (ASBR) 1 acts as the Provider Edge (PE) for all VPLS instances that span over Autonomous System (AS) 1 and ASBR 2 are viewed as the CE device. And for the other way around, for AS 2, ASBR 2 acts as the PE and ASBR 1 is viewed as the CE. MPLS is not required between ASBR 1 and ASBR 2 because VPLS is used for layer 2 linking. Each VPLS instance needs to be segregated so that it can be sent in the proper VPLS domain in ASBRs (for example, a switchport interface or Ethernet sub-interface).

Note From a BGP signaling perspective, there is no specific change within the AS. From the VPLS perspective, there is no BGP peering between ASBR1 and ASBR2.

The following figure shows a network diagram for the BGP signaling Inter-AS option A BGP configuration:



The following example shows the PE 1 BGP configuration for Inter-AS Option A:

```
router bgp 100
  neighbor 10.0.0.2 remote-as 100
   address-family l2vpn vpls
    neighbor 10.0.0.2 activate
    neighbor 10.0.0.2 send-community extended
    neighbor 10.0.0.2 suppress-signaling-protocol ldp
   exit-address-family
```

The following example shows the ASBR 1 BGP configuration for Inter-AS Option A:

```
router bgp 100
  neighbor 10.0.0.1 remote-as 100
   address-family l2vpn vpls
    neighbor 10.0.0.1 activate
    neighbor 10.0.0.1 send-community extended
    neighbor 10.0.0.1 suppress-signaling-protocol ldp
   exit-address-family
```

The following example shows the ASBR 2 BGP configuration for Inter-AS Option A:

```
router bgp 200
  neighbor 10.0.1.1 remote-as 100
   address-family l2vpn vpls
    neighbor 10.0.1.1 activate
    neighbor 10.0.1.1 send-community extended
    neighbor 10.0.1.1 suppress-signaling-protocol ldp
   exit-address-family
```

The following example shows the PE 2 BGP configuration for Inter-AS Option A:

```
router bgp 200
  neighbor 10.0.1.2 remote-as 100
   address-family l2vpn vpls
    neighbor 10.0.1.2 activate
    neighbor 10.0.1.2 send-community extended
    neighbor 10.0.1.2 suppress-signaling-protocol ldp
   exit-address-family
```

Additional References for VPLS Autodiscovery BGP Based

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
MPLS commands	Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
draft-ietf-l2vpn-signaling-08.txt	*Provisioning, Autodiscovery, and Signaling in L2VPNs*
draft-ietf-l2vpn-vpls-bgp-08.8	*Virtual Private LAN Service (VPLS) Using BGP for Autodiscovery and Signaling*
draft-ietf-mpls-lsp-ping-03.txt	*Detecting MPLS Data Plane Failures*
draft-ietf-pwe3-vccv-01.txt	*Pseudo-Wire (PW) Virtual Circuit Connection Verification (VCCV)*
RFC 3916	*Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3)*
RFC 3981	*Pseudo Wire Emulation Edge-to-Edge Architecture*
RFC 6074	Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)
RFC 4761	Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling

MIBs

MIB	MIBs Link
• CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB) • CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB) • CISCO-IETF-PW-FR-MIB (PW-FR-MIB) • CISCO-IETF-PW-MIB (PW-MIB) • CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

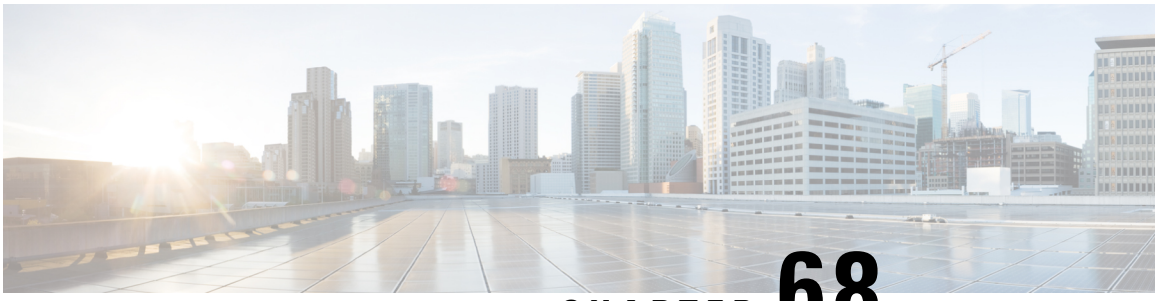
Feature Information for VPLS BGP Signaling L2VPN Inter-AS Option A

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 134: Feature Information for VPLS BGP Signaling L2VPN

Feature Name	Releases	Feature Information
VPLS BGP Signaling L2VPN	Cisco IOS XE Release 3.8S	This feature simplifies the auto-discovery and signaling of all known provider edge (PE) devices in a VPLS instance by using BGP for both functions. The following commands were introduced or modified: **autodiscovery bgp signaling bgp**, **debug bgp l2vpn vpls updates**, **neighbor suppress-signaling-protocol ldp**, **ve id**, **ve range**, **show bgp l2vpn vpls**.



CHAPTER 68

VPLS BGP Signaling L2VPN Inter-AS Option B

The VPLS BGP Signaling L2VPN Inter-AS Option B feature simplifies the auto-discovery and signaling of all known provider edge (PE) devices in a Virtual Private LAN Switching (VPLS) instance by using Border Gateway Protocol (BGP). This document describes how to configure the VPLS BGP Signaling L2VPN Inter-AS Option B feature.

Prerequisites for VPLS BGP Signaling L2VPN Inter-AS Option B

- Disable control word for Virtual Private LAN Switching (VPLS) Border Gateway Protocol (BGP) signaling by using the **no control-word** command under a pseudowire class. For example:

```
Device> enable
Device# configure terminal
Device(config)# pseudowire-class my-pw-class
Device(config-pw-class)# no control-word
```

- The route distinguisher (RD) must match for all the virtual forwarding instances (VFIs) in a VPLS domain.
- Ensure that the L2VPN VPLS Inter-AS Option B feature is configured on Autonomous System Boundary Routers (ASBRs) and PE devices.

Information About VPLS BGP Signaling L2VPN Inter-AS Option B

BGP Auto-discovery and Signaling for VPLS

The Virtual Private LAN Switching (VPLS) control plane is used for auto-discovery and signaling. Auto-discovery involves locating all provider edge (PE) devices that participate in a particular VPLS instance. Signaling is accomplished by configuring pseudowires for a VPLS instance. Prior to the introduction of the VPLS BGP Signaling L2VPN Inter-AS Option B feature, Label Distribution Protocol (LDP) was used for signaling and Border Gateway Protocol (BGP) was used for auto-discovery, as specified in RFC 6074. With the introduction of the VPLS BGP Signaling L2VPN Inter-AS Option B feature, the VPLS BGP Signaling L2VPN feature supports RFC 4761 by simplifying the auto-discovery and signaling of all known PE devices in a VPLS instance by using BGP for both functions. Auto-discovery is defined per VPLS instance.

Internal BGP (IBGP) peers exchange update messages of the L2VPN Address Family Identifier (AFI) and the Subsequent Address Family Identifier (SAFI) numbers with L2VPN information to perform both auto-discovery and signaling, which includes the Network Layer Reachability Information (NLRI).

Both BGP standards (RFC 6074 and RFC 4761) for the auto-discovery protocol for VPLS use the same BGP AFI (25) and SAFI (65) but they have different Network Layer Reachability Information (NLRI) encoding, which makes them incompatible with each other. CLI configuration is needed to distinguish the two encoding types as they are mutually exclusive per neighbor. The difference between the two BGP standards is:

- RFC 6074 provides guidelines for specifying length encoding as bits.
- RFC 4761 provides guidelines for specifying length encoding as bytes.

To detect which NLRI encoding standard is supported, the length encoding needs to be determined.

BGP L2VPN Signaling with NLRI

Network Layer Reachability Information (NLRI) enables Border Gateway Protocol (BGP) to carry supernetting information, as well as perform aggregation. Each NLRI consists of block labels that follow the structure LB, LB+1, ... , LB+VBS-1. The NLRI is exchanged between BGP devices for BGP auto-discovery with BGP signaling. The following fields are configured or auto-generated for each Virtual Private LAN Switching (VPLS) instance:

- Length (2 Octets)
- Route distinguisher (RD) is usually an auto-generated 8-byte VPN ID that can also be configured. This value must be unique for a VPLS bridge-domain (or instance).
- VPLS Endpoint ID (VEID) (2 Octets). Each PE device is configured with a VEID value.
- VPLS Endpoint Block Offset (VBO) (2 Octets).
- VPLS Endpoint Block Size (VBS) (2 Octets).
- Label Base (LB) (3 Octets).

- Extended Community Type (2 Octets) - 0x800A attributes. The Route Target (RT) specified for a VPLS instance, next-hop and other Layer 2 information is carried in this encoding. An RT-based import and export mechanism similar to L3VPN is performed by BGP to perform filtering on the L2VPN NLRIs of a particular VPLS instance.
- Encapsulation Type (1 Octet) - VPLS = 19
- Control Flags (1 Octet)
- Layer 2 Maximum Transmission Unit (MTU) (2 Octets)
- Reserved (2 Octets)

How to Configure VPLS BGP Signaling L2VPN Inter-AS Option B

Enabling BGP Auto-discovery and BGP Signaling

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices by BGP auto-discovery and BGP signaling functions announced through IBGP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *vfi-context-name*
4. **vpn id** *vpn-id*
5. **autodiscovery bgp signaling bgp**
6. **ve id** *ve-ID-number*
7. **ve range** *ve-range-number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *vfi-context-name* **Example:**	Establishes a Layer 2 VPN (L2VPN) virtual forwarding interface (VFI) for specifying core-facing pseudowires in

	Command or Action	Purpose
	`Device(config)# l2vpn vfi context vfi1`	a Virtual Private LAN Services (VPLS) and enters L2VFI configuration mode. • The VFI represents an emulated LAN or a VPLS forwarder from the VPLS architectural model when using an emulated LAN interface.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 10`	Configures a VPN ID for the VPLS domain.
Step 5	**autodiscovery bgp signaling bgp** **Example:** `Device(config-vfi)# autodiscovery bgp signaling bgp`	Enables BGP auto-discovery and BGP signaling on the device.
Step 6	**ve id** *ve-ID-number* **Example:** `Device(config-vfi)# ve id 1`	Configures a VPLS Endpoint ID (VEID) for the NLRI exchanged between BGP devices for BGP auto-discovery with BGP signaling. • For example, VEID numbering sequences such as 1,2,3 or 501, 502, 503 are preferred because the VEIDs are contiguous. • Avoid a non-contiguous numbering scheme such as 100, 200, 300. Repeat this step to add more VEIDs. The VEID must be unique within the same VPLS domain for all PE devices. **Note** If you change the VEID, then the virtual circuit (VC) reprovisions and traffic is impacted as a result.
Step 7	**ve range** *ve-range-number* **Example:** `Device(config-vfi)# ve range 10`	Overrides the minimum size of VPLS edge (VE) blocks. • The VE range value should be approximately the same as the number of neighbors (up to 100). • The VE range can be configured based on the number of neighboring PE devices in the network. • For example, if 50 PE devices are in a VPLS domain, then a VE range of 50 is better than 10 because the number of NLRIs exchanged are less and the convergence time is reduced. **Note** If no VE range is configured or an existing VE range value is removed, then the default VE range of 10 is applied. The default VE range should not be used if the device has many PE neighbors. **Note** If you change the VE range, then the VC reprovisions and traffic is impacted as a result.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-vfi)# end`	Exits L2 VFI configuration mode and returns to privileged EXEC mode. **Note** Commands take effect after the device exits L2VFI configuration mode.

Configuring BGP Signaling for VPLS Autodiscovery

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **address-family l2vpn vpls**
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
9. **neighbor** {*ip-address* | *peer-group-name*} **suppress-signaling-protocol ldp**
10. **exit-address-family**
11. Repeat steps 1 to 10 to configure and activate other BGP neighbors in an L2VPN address family.
12. **end**
13. **show l2vpn vfi**
14. **show ip bgp l2vpn vpls** {**all** [**summary**] | **rd** *route-distinguisher*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode for the specified routing process.
Step 4	**bgp graceful-restart** **Example:** `Device(config-router)# bgp graceful-restart`	Enables the Border Gateway Protocol (BGP) graceful restart capability globally for all BGP neighbors.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 198.51.100.1 remote-as 65000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.
Step 6	**address-family l2vpn vpls** **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The **vpls** keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers and a L2VPN VPLS address family session is created.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 198.51.100.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Device(config-router-af)# neighbor 198.51.100.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **suppress-signaling-protocol ldp** **Example:** `Device(config-router-af)# neighbor 198.51.100.1 suppress-signaling protocol ldp`	Suppresses LDP signaling for a BGP neighbor so that BGP signaling for VPLS auto-discovery is used instead. • In this example, LDP signaling is suppressed for the neighbor at 10.10.10.1.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 11	Repeat steps 1 to 10 to configure and activate other BGP neighbors in an L2VPN address family.	

	Command or Action	Purpose
Step 12	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Step 13	**show l2vpn vfi**
	Example:
	`Device# show l2vpn vfi`
	
	`PE1-standby#sh l2vpn vfi`
	`Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%`
	`Time source is hardware calendar, *20:50:52.526 GMT Wed Aug 29 2012`
	
	`Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No`
	
	`VFI name: VFI1, state: up, type: multipoint, signaling: BGP`
	`VPN ID: 1, VE-ID: 10, VE-SIZE: 10`
	`RD: 1:1, RT: 1:1`
	`Bridge-Domain 100 attachment circuits:`
	`Pseudo-port interface: pseudowire100001`
	`Interface Peer Address VE-ID Local Label Remote Label S`
	`pseudowire100003 198.51.100.2 11 1003 2002 Y`
	`pseudowire100005 198.51.100.3 12 1004 2002 Y`
	
	`VFI name: VFI2, state: up, type: multipoint, signaling: BGP`
	`VPN ID: 2, VE-ID: 20, VE-SIZE: 12`
	`RD: 1:2, RT: 1:2, import 3:3, export 4:4`
	`Bridge-Domain 200 attachment circuits:`
	`Pseudo-port interface: pseudowire100002`
	`Interface Peer Address VE-ID Local Label Remote Label S`
	`pseudowire100004 198.51.100.2 21 1021 2020 Y`
	`pseudowire100006 198.51.100.3 22 1022 2020 Y`
		Displays information about the configured VFI instances.

Step 14	**show ip bgp l2vpn vpls** {**all** [**summary**] | **rd** *route-distinguisher*}
	Example:
	`Device# show ip bgp l2vpn vpls all summary`
	
	`BGP router identifier 198.51.100.1, local AS number 65000`
	`BGP table version is 14743, main routing table version 14743`
	`6552 network entries using 1677312 bytes of memory`
	`6552 path entries using 838656 bytes of memory`
	`3276/3276 BGP path/bestpath attribute entries using 760032 bytes of memory`
	`1638 BGP extended community entries using 65520`
		Displays information about the L2VPN VPLS address family.

	Command or Action	Purpose
	bytes of memory 0 BGP route-map cache entries using 0 bytes of memory 0 BGP filter-list cache entries using 0 bytes of memory BGP using 3341520 total bytes of memory BGP activity 9828/3276 prefixes, 9828/3276 paths, scan interval 60 secs Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd 198.51.101.1 4 65000 90518 90507 14743 0 0 8w0d 1638 198.51.102.2 4 65000 4901 4895 14743 0 0 2d01h 1638 198.51.103.3 4 65000 4903 4895 14743 0 0 2d01h 1638	

Configuration Examples for L2VPN VPLS Inter-AS Option B

Example: VPLS BGP Signaling L2VPN Inter-AS Option B

The following example configuration describes Inter-AS Option B for VPLS BGP signaling in a Layer 2 VPN. BGP MPLS forwarding is required between ASBR 1 and ASBR 2.

Note From a BGP signaling perspective, there is no specific change within the autonomous system. From the VPLS perspective, there is EBGP peering between ASBR1 and ASBR2.

The following figure shows a network diagram for the BGP signaling Inter-AS option B BGP configuration:

Figure 93: VPLS BGP Signaling L2VPN Inter-AS Option B Sample Topology

The following example shows the PE 1 BGP configuration for Inter-AS Option B:

```
l2vpn vfi context TEST101
 vpn id 1
 autodiscovery bgp signaling bgp
  ve id 1
  route-target import 22:22
  route-target export 11:11
  no auto-route-target
!
mpls ldp graceful-restart
!
bridge-domain 1
 member GigabitEthernet0/0/7 service-instance 101
```

```
 member vfi TEST101
!
interface Loopback0
 ip address 198.51.101.2 255.255.255.255
!
interface GigabitEthernet0/0/1
 description - connects to RR1
 ip address 200.1.1.1 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet0/0/7
 description - connects to CE1
 no ip address
 negotiation auto
 service instance 101 ethernet
  encapsulation dot1q 101
  rewrite ingress tag pop 1 symmetric
 !
!
router ospf 10
 nsf
 network 200.1.1.0 0.0.0.255 area 0
 network 198.51.101.2 0.0.0.0 area 0
!
router bgp 10
 bgp log-neighbor-changes
 bgp update-delay 1
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 200.1.1.1 remote-as 10
 neighbor 200.1.1.1 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 200.1.1.1 activate
  neighbor 200.1.1.1 send-community extended
  neighbor 200.1.1.1 suppress-signaling-protocol ldp
 exit-address-family
!
```

The following example shows the ASBR 1 BGP configuration for Inter-AS Option B:

```
router bgp 10
bgp log-neighbor-changes
bgp update-delay 1
bgp graceful-restart restart-time 120
bgp graceful-restart stalepath-time 360
bgp graceful-restart
no bgp default ipv4-unicast
no bgp default route-target filter
 neighbor 192.0.2.1 remote-as 10
 neighbor 192.0.2.1 update-source Loopback0
 neighbor 203.0.203.1 remote-as 20
 neighbor 203.0.203.1 ebgp-multihop 255
 neighbor 203.0.203.1 update-source Loopback0
!
address-family ipv4
 exit-address-family
!
address-family l2vpn vpls
```

```
 neighbor 192.0.2.1 activate
 neighbor 192.0.2.1 send-community extended
 neighbor 192.0.2.1 next-hop-self
 neighbor 192.0.2.1 suppress-signaling-protocol ldp
 neighbor 203.0.203.1 activate
 neighbor 203.0.203.1 send-community extended
 neighbor 203.0.203.1 next-hop-self
 neighbor 203.0.203.1 suppress-signaling-protocol ldp
 exit-address-family
```

The following example shows the ASBR 2 BGP configuration for Inter-AS Option B:

```
mpls ldp graceful-restart
!
interface Loopback0
 ip address 203.0.203.1 255.255.255.255
!
interface GigabitEthernet0/0/1
 description - connects to RR1
 ip address 192.0.2.2 255.255.255.0
 negotiation auto
 mpls ip
 mpls bgp forwarding
!
interface GigabitEthernet0/2/1
 description - connects to ASBR3
 ip address 192.0.2.200 255.255.255.0
 negotiation auto
 mpls ip
 mpls bgp forwarding
!
router ospf 10
 nsf
 network 192.0.2.0 0.0.0.255 area 0
 network 203.0.203.1 0.0.0.0 area 0
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 10
 bgp log-neighbor-changes
 bgp update-delay 1
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 no bgp default route-target filter
 neighbor 203.0.203.3 remote-as 20
 neighbor 203.0.203.3 ebgp-multihop 255
 neighbor 203.0.203.3 update-source Loopback0
 neighbor 203.0.203.2 remote-as 10
 neighbor 203.0.203.2 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 203.0.203.3 activate
  neighbor 203.0.203.3 send-community extended
  neighbor 203.0.203.3 next-hop-self
  neighbor 203.0.203.3 suppress-signaling-protocol ldp
  neighbor 203.0.203.2 activate
  neighbor 203.0.203.2 send-community extended
  neighbor 203.0.203.2 next-hop-self
  neighbor 203.0.203.2 suppress-signaling-protocol ldp
 exit-address-family
```

The following example shows the PE 2 BGP configuration for Inter-AS Option B:

```
l2vpn vfi context TEST101
 vpn id 1
 autodiscovery bgp signaling bgp
  ve id 2
  route-target import 22:22
  route-target export 11:11
  no auto-route-target
!
mpls ldp graceful-restart
!
bridge-domain 1
 member GigabitEthernet0/0/7 service-instance 101
 member vfi TEST101
!
interface Loopback0
 ip address 192.0.2.3 255.255.255.255
!
interface GigabitEthernet0/0/1
 description - connects to RR1
 ip address 192.0.2.1 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet0/0/7
 description - connects to CE2
 no ip address
 negotiation auto
 service instance 101 ethernet
  encapsulation dot1q 101
  rewrite ingress tag pop 1 symmetric
 !
!
router ospf 10
 nsf
 network 192.0.2.0 0.0.0.255 area 0
 network 192.0.2.3 0.0.0.0 area 0
!
router bgp 10
 bgp log-neighbor-changes
 bgp update-delay 1
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 211.1.1.1 remote-as 10
 neighbor 211.1.1.1 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 211.1.1.1 activate
  neighbor 211.1.1.1 send-community extended
  neighbor 211.1.1.1 suppress-signaling-protocol ldp
 exit-address-family
```

The following example shows the route reflector device BGP configuration for Inter-AS Option B:

```
mpls ldp graceful-restart
!
interface Loopback0
 ip address 203.0.203.1 255.255.255.255
!
```

```
interface GigabitEthernet1/1
 description - connects to PE1
 ip address 203.0.203.2 255.255.255.0
 mpls ip
!
interface GigabitEthernet1/2
 description - connects to PE2
 ip address 203.0.203.3 255.255.255.0
 mpls ip
!
interface GigabitEthernet1/5
 description - connects to ASBR1
 ip address 203.0.203.4 255.255.255.0
 mpls ip
 mpls bgp forwarding
!
interface GigabitEthernet1/6
 description - connects to ASBR2
 ip address 203.0.203.5 255.255.255.0
 mpls ip
 mpls bgp forwarding
!
router ospf 10
 nsf
 network 203.0.203.6 0.0.0.255 area 0
 network 203.0.203.7 0.0.0.255 area 0
 network 203.0.203.8 0.0.0.255 area 0
 network 203.0.203.9 0.0.0.255 area 0
 network 203.0.203.1 0.0.0.0 area 0
!
router bgp 10
 bgp log-neighbor-changes
 bgp update-delay 1
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 neighbor 203.0.203.11 remote-as 10
 neighbor 203.0.203.11 update-source Loopback0
 neighbor 203.0.203.12 remote-as 10
 neighbor 203.0.203.12 update-source Loopback0
 neighbor 203.0.203.13 remote-as 10
 neighbor 203.0.203.13 update-source Loopback0
 neighbor 203.0.203.14 remote-as 10
 neighbor 203.0.203.14 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family l2vpn vpls
  neighbor 203.0.203.11 activate
  neighbor 203.0.203.11 send-community extended
  neighbor 203.0.203.11 route-reflector-client
  neighbor 203.0.203.11 suppress-signaling-protocol ldp
  neighbor 203.0.203.12 activate
  neighbor 203.0.203.12 send-community extended
  neighbor 203.0.203.12 route-reflector-client
  neighbor 203.0.203.12 suppress-signaling-protocol ldp
  neighbor 203.0.203.13 activate
  neighbor 203.0.203.13 send-community extended
  neighbor 203.0.203.13 route-reflector-client
  neighbor 203.0.203.13 suppress-signaling-protocol ldp
  neighbor 203.0.203.14 activate
  neighbor 203.0.203.14 send-community extended
```

```
  neighbor 203.0.203.14 route-reflector-client
  neighbor 203.0.203.14 suppress-signaling-protocol ldp
 exit-address-family
!
```

Additional References for VPLS BGP Signaling L2VPN Inter-AS Option B

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Multiprotocol Label Switching Command Reference
L2VPN VPLS Inter-AS Option B	*L2VPN VPLS Inter-AS Option B*
VPLS Autodiscovery: BGP Based	*VPLS Autodiscovery BGP Based*
VPLS BGP Signaling L2VPN Inter-AS Option A	*VPLS BGP Signaling L2VPN Inter-AS Option A*

Standards and RFCs

Standard and RFC	Title
draft-kothari-l2vpn-auto-site-id-01.txt	*Automatic Generation of Site IDs for Virtual Private LAN Service*
draft-ietf-l2vpn-vpls-multihoming-03.txt	*BGP based Multi-homing in Virtual Private LAN Service*
RFC 6074	*Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*
RFC 4761	*Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

MIBs

MIB	MIBs Link
• CISCO-IETF-PW-ATM-MIB (PW-ATM-MIB) • CISCO-IETF-PW-ENET-MIB (PW-ENET-MIB) • CISCO-IETF-PW-FR-MIB (PW-FR-MIB) • CISCO-IETF-PW-MIB (PW-MIB) • CISCO-IETF-PW-MPLS-MIB (PW-MPLS-MIB)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

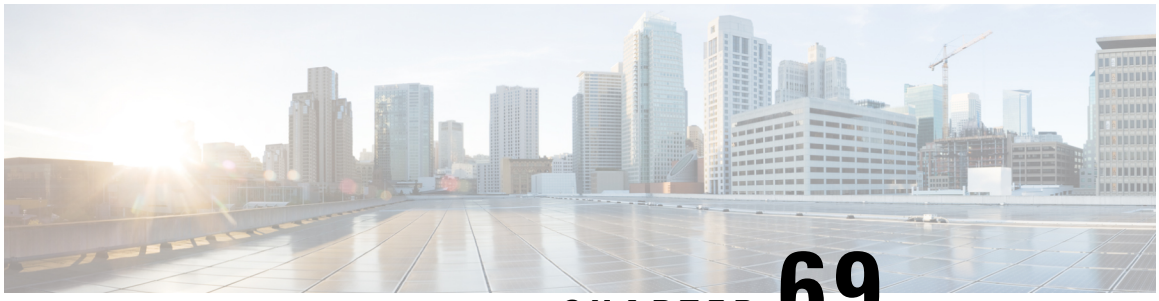
Feature Information for VPLS BGP Signaling L2VPN Inter-AS Option B

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 135: Feature Information for VPLS BGP Signaling L2VPN Inter-AS Option B

Feature Name	Releases	Feature Information
VPLS BGP Signaling L2VPN Inter-AS Option B	Cisco IOS XE Release 3.12S	This feature simplifies the auto-discovery and signaling of all known provider edge (PE) devices in a VPLS instance by using BGP for both functions. The following command was modified: **show mpls forwarding**



CHAPTER 69

Frame Relay over L2TPv3

The Frame Relay over L2TPv3 (FRoL2TPv3) feature enables Frame Relay switching over Layer 2 Tunnel Protocol Version 3 (L2TPv3). The feature works with like interfaces and disparate interfaces (L2VPN interworking).

Prerequisites for Configuring Frame Relay over L2TPv3

Before configuring Frame Relay over L2TPv3, you should understand how to configure Layer 2 VPNs and Frame Relay. See the “Additional References” section in this chapter for pointers to the feature modules that explain how to configure and use Layer 2 VPNs and Frame Relay.

Restrictions for Configuring Frame Relay over L2TPv3

The following functionalities are not supported:

- Frame Relay to 802.1Q/QinQ VLAN interworking
- Frame Relay-to-Ethernet routed interworking
- Frame Relay port-to-port switching
- L2TPv3 pseudowire redundancy for Frame Relay

Information About Configuring Frame Relay over L2TPv3

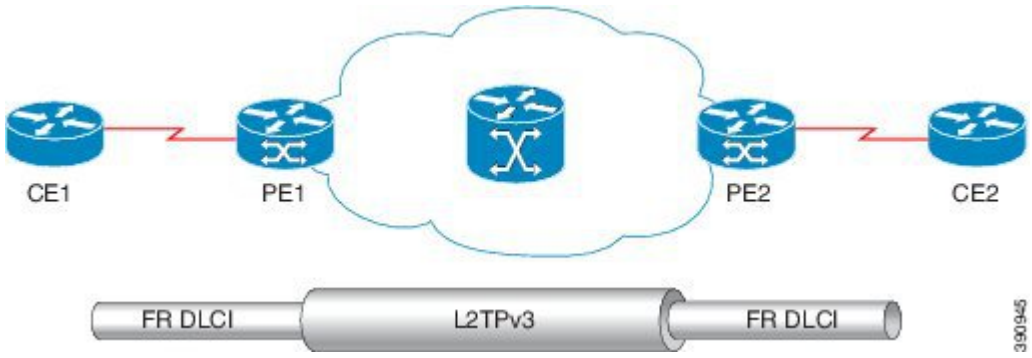
Frame Relay over L2TPv3 Overview

Frame Relay over L2TPv3 enables provider edge (PE) devices to forward Frame Relay frames to pseudowires based on the receiving interface and the Data-Link Connection Identifier (DLCI) number. PE devices also provide Local Management Interface (LMI)-based signaling to customer edge (CE) devices, emulating Frame Relay switches.

In Frame Relay over L2TPv3, the Frame Relay header is retained at the ingress PE device. The device does not reconstruct the Frame Relay header before forwarding packets to the CE device.

The figure below shows a Frame Relay over L2TPv3 topology.

Figure 94: Frame Relay over L2TPv3



Frame Relay over L2TPv3 supports the following functionalities:

- Frame Relay data-link connection identifier (DLCI)-to-Frame Relay DLCI
- Frame Relay DLCI-to-Ethernet port / 802.1Q / QinQ bridged interworking
- Local Management Interface (LMI)
- L2TPv3 sequencing
- L2TPv3 tunnel marking

How to Configure Frame Relay over L2TPv3

Configuring Frame Relay over L2TPv3 without LMI

This section explains how to configure Frame Relay over L2TPv3 without enabling Local Management Interface (LMI).

On CE1

The CE1 device receives the Frame Relay frames forwarded by the PE1 device over the Frame Relay link. On CE1, configure an interface and a DLCI number based on which the PE1 device forwards traffic to the appropriate pseudowire.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **encapsulation frame-relay** [**cisco** | **ietf**]
6. **no keepalive**
7. **frame-relay intf-type dce**
8. **exit**
9. **interface** *type number* **point-to-point**
10. **ip address** *ip-address mask*
11. **frame-relay interface-dlci** *dlci*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 5	**encapsulation frame-relay** [**cisco**	**ietf**] **Example:**	Specifies Frame Relay encapsulation for the interface. • You can specify different types of encapsulations. • You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.

	Command or Action	Purpose
	`Device(config-if)# encapsulation frame-relay ietf`	
Step 6	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables the keepalive configuration.
Step 7	**frame-relay intf-type dce** **Example:** `Device(config-if)# frame-relay intf-type dce`	Specifies that the interface is a DCE switch. • You can also specify the interface to support Network-to-Network Interface (NNI) and DTE connections.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **point-to-point** **Example:** `Device(config)# interface serial 3/1/0.1 point-to-point`	Specifies a serial interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 198.51.100.2 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 11	**frame-relay interface-dlci** *dlci* **Example:** `Device(config-if)# frame-relay interface-dlci 25`	Assigns a data-link connection identifier (DLCI) to the Frame Relay interface.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode. After configuring CE1, you can configure CE2 in a similar manner.

On PE1

The PE1 device forwards Frame Relay frames to the appropriate pseudowire, based on the receiving interface and DLCI number configured on the CE1 device.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number*
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **encapsulation frame-relay** [**cisco** | **ietf**]
6. **no keepalive**
7. **pseudowire-class** [*pw-class-name*]
8. **encapsulation l2tpv3**
9. **ip local interface loopback** *loopback id*
10. **connect** *connection-name interface dlci* **l2transport**
11. **xconnect** *peer-router-id vcid* **encapsulation l2tpv3 pw-class l2tpv3**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 5	**encapsulation frame-relay** [**cisco**	**ietf**] **Example:** `Device(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. • You can specify different types of encapsulations. • You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 6	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables the keepalive configuration.

	Command or Action	Purpose
Step 7	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class l2tpv3`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 8	**encapsulation l2tpv3** **Example:** `Device(config-pw)# encapsulation l2tpv3`	Specifies the tunneling encapsulation as L2TPv3.
Step 9	**ip local interface loopback** *loopback id* **Example:** `Device(config-pw)# ip local interface Loopback0`	Specifies the local loopback interface on PE1 for the L2TPv3 tunnel.
Step 10	**connect** *connection-name interface dlci* **l2transport** **Example:** `Device(config)# connect fr1 serial5/0 1000 l2transport`	Defines connections between Frame Relay Permanent Virtual Circuits (PVCs) and enters connect configuration mode. • Using the **l2transport** keyword specifies that the PVC is not a locally switched PVC, but is tunneled over the backbone network. • The *connection-name* argument is a text string that you provide. • The *interface* argument is the interface on which a PVC connection is defined. • The *dlci* argument is the DLCI number of the PVC that is connected.
Step 11	**xconnect** *peer-router-id vcid* **encapsulation l2tpv3 pw-class l2tpv3** **Example:** `Device(config-xconnect-conn-config)# xconnect 198.51.100.2 123 encapsulation l2tpv3 pw-class l2tpv3`	Creates the VC to transport the Layer 2 packets. • In a DLCI-to DLCI connection type, Frame Relay over L2TPv3 uses the **xconnect** command in connect configuration mode. • The *vcid* or identifier of the virtual circuit (VC) between the PE devices should be the same on both devices that are being connected.
Step 12	**end** **Example:** `Device(config-xconnect-conn-config)# end`	Exits connect configuration mode and returns to privileged EXEC mode. After configuring PE1, you can configure PE2 in a similar manner.

Configuring Frame Relay over L2TPv3 with LMI

This section explains how to configure Frame Relay over L2TPv3 with Local Management Interface (LMI) enabled.

On CE1

The CE1 device receives the Frame Relay frames forwarded by the PE1 device over the Frame Relay link. On CE1, configure an interface and a DLCI number based on which the PE1 device forwards traffic to the appropriate pseudowire. Local Management Interface (LMI) is also tunneled over the pseudowire. Therefore, you need to properly configure the customer edge (CE) device for LMI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot/subslot /port* [**.** *subinterface*]
4. **no ip address** [*ip-address mask*] [**secondary**]
5. **encapsulation frame-relay** [**cisco** | **ietf**]
6. **frame-relay intf-type dce**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot/subslot /port* [**.** *subinterface*] **Example:** `Device(config)# interface serial3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 4	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 5	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Device(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. • You can specify different types of encapsulations. • You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 6	**frame-relay intf-type dce** **Example:**	Specifies that the interface is a Data Communications Equipment (DCE) switch.

	Command or Action	Purpose
	`Device(config-if)# frame-relay intf-type dce`	• You can also specify the interface to support Network-to-Network Interface (NNI) and Data Transmission Equipment (DTE) connections.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode. After configuring CE1, you can configure CE2 in a similar manner.

On PE1

The PE1 device forwards Frame Relay frames to the CE1 device over the Frame Relay link. The PE1 device also provides Local Management Interface (LMI) signaling to the CE1 device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot/subslot/port* [**.** *subinterface*]
4. **encapsulation frame-relay** [**cisco** | **ietf**]
5. **pseudowire-class** [*pw-class-name*]
6. **encapsulation l2tpv3**
7. **ip local interface loopback** *loopback id*
8. **connect** *connection-name interface dlci* **l2transport**
9. **xconnect** *peer-router-id vcid* **encapsulation l2tpv3 pw-class l2tpv3**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot/subslot/port* [**.** *subinterface*] **Example:** `Device(config)# interface serial3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 4	**encapsulation frame-relay** [**cisco**	**ietf**]	Specifies Frame Relay encapsulation for the interface.

	Command or Action	Purpose
	Example: `Device(config-if)# encapsulation frame-relay ietf`	• You can specify different types of encapsulations. • You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 5	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class l2tpv3`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 6	**encapsulation l2tpv3** **Example:** `Device(config-pw)# encapsulation l2tpv3`	Specifies the tunneling encapsulation as L2TPv3.
Step 7	**ip local interface loopback** *loopback id* **Example:** `Device(config-pw)# ip local interface Loopback0`	Specifies the local loopback interface.
Step 8	**connect** *connection-name interface dlci* **l2transport** **Example:** `Device(config)# connect fr1 serial5/0 1000 l2transport`	Defines connections between Frame Relay Permanent Virtual Circuits (PVCs) and enters connect configuration mode. • Using the **l2transport** keyword specifies that the PVC is not a locally switched PVC, but is tunneled over the backbone network. • The *connection-name* argument is a text string that you provide. • The *interface* argument is the interface on which a PVC connection is defined. • The *dlci* argument is the DLCI number of the PVC that is connected.
Step 9	**xconnect** *peer-router-id vcid* **encapsulation l2tpv3 pw-class l2tpv3** **Example:** `Device(config-fr-pw-switching)# xconnect 198.51.100.2 123 encapsulation l2tpv3 pw-class l2tpv3`	Creates the virtual circuit (VC) to transport the Layer 2 packets. • In a DLCI-to-DLCI connection type, Frame Relay over L2TPv3 uses the **xconnect** command in connect configuration mode.
Step 10	**end** **Example:** `Device(config-fr-pw-switching)# end`	Exits connect configuration mode and returns to privileged EXEC mode. After configuring PE1, you can configure PE2 in a similar manner.

Configuring Frame Relay L2TPv3 Tunnel Marking

L2TPv3 Tunnel Marking introduces the capability to define and control the quality of service (QoS) for incoming customer traffic on the provider edge (PE) device in a service provider network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-name*
4. **match fr-dlci** *dlci-number*
5. **policy-map dlci** *dlci-number*
6. **class** *class-name*
7. **set ip precedence tunnel** *precedence-value*
8. **interface serial** *slot/subslot/port* [**.** *subinterface*]
9. **no ip address** [*ip-address mask*] [**secondary**]
10. **encapsulation frame-relay** [**cisco** | **ietf**]
11. **no keepalive**
12. **service-policy input** *policy-name*
13. **end**
14. **pseudowire-class** [*pw-class-name*]
15. **encapsulation l2tpv3**
16. **ip local interface loopback** *loopback id*
17. **connect** *connection-name interface dlci* **l2transport**
18. **xconnect** *peer-router-id vcid* **encapsulation l2tpv3 pw-class l2tpv3**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-name* **Example:** `Device(config)# class-map class1`	Specifies the user-defined name of the traffic class and enters class map configuration mode.

	Command or Action	Purpose
Step 4	**match fr-dlci** *dlci-number* **Example:** `Device(config-cmap)# match fr-dlci 50`	Specifies the number of the Data-Link Connection Identifier (DLCI) associated with the packet as a match criterion in the class map.
Step 5	**policy-map dlci** *dlci-number* **Example:** `Device(config-cmap)# policy-map dlci 50`	Specifies the type of policy map as DLCI and enters policy map configuration mode.
Step 6	**class** *class-name* **Example:** `Device(config-pmap)# class class1`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy and enters policy-map class configuration mode.
Step 7	**set ip precedence tunnel** *precedence-value* **Example:** `Device(config-pmap-c)# set ip precedence tunnel 2`	Sets the precedence value in the header of the L2TPv3 tunneled packet for tunnel marking.
Step 8	**interface serial** *slot/subslot/port* [**.** *subinterface*] **Example:** `Device(config-pmap-c)# interface serial3/1/0`	Specifies a serial interface and enters interface configuration mode.
Step 9	**no ip address** [*ip-address mask*] [**secondary**] **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 10	**encapsulation frame-relay** [**cisco** | **ietf**] **Example:** `Device(config-if)# encapsulation frame-relay ietf`	Specifies Frame Relay encapsulation for the interface. • You can specify different types of encapsulations. • You can set one interface to Cisco encapsulation and the other interface to IETF encapsulation.
Step 11	**no keepalive** **Example:** `Device(config-if)# no keepalive`	Disables the keepalive configuration.
Step 12	**service-policy input** *policy-name* **Example:** `Device(config-if)# service-policy input policy1`	Attaches a traffic policy to the interface.

	Command or Action	Purpose
Step 13	**end** **Example:** `Device(config-if)# end`	Exits connect configuration mode and returns to privileged EXEC mode.
Step 14	**pseudowire-class** [*pw-class-name*] **Example:** `Device(config)# pseudowire-class l2tpv3`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 15	**encapsulation l2tpv3** **Example:** `Device(config-pw)# encapsulation l2tpv3`	Specifies the tunneling encapsulation as L2TPv3.
Step 16	**ip local interface loopback** *loopback id* **Example:** `Device(config-pw)# ip local interface Loopback0`	Specifies the local loopback interface.
Step 17	**connect** *connection-name interface dlci* **l2transport** **Example:** `Device(config-pw)# connect fr1 serial5/0 1000 l2transport`	Defines connections between Frame Relay Permanent Virtual Circuits (PVCs) and enters connect configuration mode. • Using the **l2transport** keyword specifies that the PVC is not a locally switched PVC, but is tunneled over the backbone network. • The *connection-name* argument is a text string that you provide. • The *interface* argument is the interface on which a PVC connection is defined. • The *dlci* argument is the DLCI number of the PVC that is connected.
Step 18	**xconnect** *peer-router-id vcid* **encapsulation l2tpv3 pw-class l2tpv3** **Example:** `Device(config-xconnect-conn-config)# xconnect 198.51.100.2 123 encapsulation l2tpv3 pw-class l2tpv3`	Creates the VC to transport the Layer 2 packets. • In a DLCI-to-DLCI connection type, Frame Relay over L2TPv3 uses the **xconnect** command in connect configuration mode.
Step 19	**end** **Example:** `Device(config-xconnect-conn-config)# end`	Exits connect configuration mode and returns to privileged EXEC mode.

Verifying Frame Relay over L2TPv3 Configuration

You can use **show** commands to view information about a Frame Relay over L2TPv3 configuration.

SUMMARY STEPS

1. **show xconnect all detail**
2. **show frame-relay pvc**
3. **show connection**

DETAILED STEPS

Step 1 show xconnect all detail

The following example is sample output of the **show xconnect all detail** command:

Example:

```
Device# show xconnect all detail

Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                          S2
------+---------------------------------+--+---------------------------------+--
UP pri   ac Se0/2/0:0:16(FR DLCI)        UP l2tp 22.2.2.2:100                  UP
            Interworking: L2L                    Session ID: 306532470
                                                 Tunnel ID: 1381396806
                                                 Protocol State: UP
                                                 Remote Circuit State: UP
                                                 pw-class: fr_fr
UP pri   ac Se0/2/0:0:17(FR DLCI)        UP l2tp 22.2.2.2:101                  UP
            Interworking: Eth                    Session ID: 1373339282
                                                 Tunnel ID: 1381396806
                                                 Protocol State: UP
                                                 Remote Circuit State: UP
                                                 pw-class: fr_eth
```

Step 2 show frame-relay pvc

The following example is sample output of the **show frame-relay pvc** command:

Example:

```
Device# show frame-relay pvc

pseudowire100001 is up, VC status is up PW type: Ethernet
  Create time: 5d20h, last status change time: 5d20h
    Last label FSM state change time: 5d20h
  Destination address: 2.1.1.2 VC ID: 1234000
    Output interface: Et0/0, imposed label stack {2001}
  Preferred path: not configured
  Default path: active
  Next hop: 20.0.0.2
Member of xconnect service Et1/0.1-1001, group right
  Associated member Et1/0.1 is up, status is up
  Interworking type is Ethernet
  Service id: 0x6d000002
Signaling protocol: LDP, peer 2.1.1.2:0 up
```

```
  Targeted Hello: 1.1.1.1(LDP Id) -> 2.1.1.2, LDP is UP
  Graceful restart: not configured and not enabled
  Non stop routing: not configured and not enabled
  PWid FEC (128), VC ID: 1234000
  Status TLV support (local/remote) : enabled/supported
    LDP route watch : enabled
    Label/status state machine : established, LruRru
    Local dataplane status received : No fault
    BFD dataplane status received : Not sent
    BFD peer monitor status received : No fault
    Status received from access circuit : No fault
    Status sent to access circuit : No fault
    Status received from pseudowire i/f : No fault
    Status sent to network peer : No fault
    Status received from network peer : No fault
    Adjacency status of remote peer : No fault
Sequencing: receive disabled, send disabled
Bindings
Parameter    Local                          Remote
------------ ------------------------------ ------------------------------
Label        2007                           2001
Group ID     0                              6
Interface
MTU          1500                           1500
Control word on (configured: autosense)     on
PW type      Ethernet                       Ethernet
VCCV CV type 0x12                           0x12
               LSPV [2], BFD/Raw [5]          LSPV [2], BFD/Raw [5]
VCCV CC type 0x07                           0x07
             CW [1], RA [2], TTL [3]        CW [1], RA [2], TTL [3]
Status TLV   enabled                        supported
Dataplane:
  SSM segment/switch IDs: 12309/4115 (used), PWID: 1
Rx Counters
  106563 input transit packets, 9803650 bytes
  0 drops, 0 seq err
Tx Counters
  0 output transit packets, 0 bytes
  0 drops
```

Step 3 show connection

The following example is sample output of the **show connection** command:

Example:

```
Device# show connection

ID   Name            Segment 1              Segment 2                  State
================================================================================
1    fr_fr           Se0/2/0:0 16           22.2.2.2 100               UP
2    fr_eth          Se0/2/0:0 17           22.2.2.2 101               UP

--------------------------
```

Configuration Examples for Frame Relay over L2TPv3

Example: Frame Relay over L2TPv3 with LMI

The following example shows how to configure Frame Relay over L2TPv3 with Local Management Interface (LMI) enabled:

PE1 device	CE1 device

configure terminal
 interface Serial 0/2/0:0
  no ip address
  encapsulation frame-relay
!
keepalive 15
frame-relay lmi-type cisco
	configure terminal
	 interface Serial 1/0:0
	  no ip address
	  encapsulation frame-relay
	  frame-relay intf-type dce
	!
	keepalive 15
	frame-relay lmi-type cisco
	interface Serial 1/0:0.100 point-to-point
	
	 ip address 198.51.100.33 255.255.255.0
	 frame-relay interface-dlci 16

Examples: Frame Relay over L2TPv3 without LMI

The following example shows how to configure Frame Relay DLCI-to-Frame Relay DLCI over L2TPv3 without Local Management Interface (LMI) enabled:

PE1 device	CE1 device

configure terminal
 interface Serial 0/1/0
  encapsulation frame-relay
!
pseudowire-class fr_l2tpv3
 encapsulation l2tpv3
 ip local interface Loopback0
!
connect FR Serial 0/1/0 100 l2transport
 xconnect 198.51.100.2 100 encapsulation l2tpv3
pw-class fr_l2tpv3
	configure terminal
	 interface Serial 0/0/0
	  encapsulation frame-relay
	  exit
	!
	interface Serial 0/0/0.100 point-to-point
	
	 ip address 198.51.100.22 255.255.255.0
	 frame-relay interface-dlci 100

The following example shows how to configure Frame Relay DLCI-to-Ethernet Interworking over L2TPv3 without LMI enabled:

PE1 device	CE1 device
configure terminal pseudowire-class fr_eth encapsulation l2tpv3 interworking ethernet ip local interface Loopback0 ! connect FR-Eth Serial 0/1/0 500 l2transport xconnect 198.51.100.27 500 encapsulation l2tpv3 pw-class fr_eth	configure terminal interface Serial 0/0/0.500 point-to-point frame-relay interface-dlci 500 ! interface BVI 200 ip address 198.51.100.29 255.255.255.0

Additional References for Frame Relay over L2TPv3

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Multiprotocol Label Switching Command Reference
Configuring Frame Relay over MPLS	*Configuring Frame Relay over MPLS*
MPLS Layer 2 VPNs Configuration Guide	*MPLS Layer 2 VPNs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2427	*Multiprotocol Interconnect over Frame Relay*
RFC 4591	*Frame Relay over Layer 2 Tunneling Protocol Version 3 (L2TPv3)*

MIBs

MIB	MIBs Link
• Cisco Frame Relay MIB (CISCO-FRAME-RELAY-MIB.my) • Interfaces MIB (IF-MIB.my)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/support

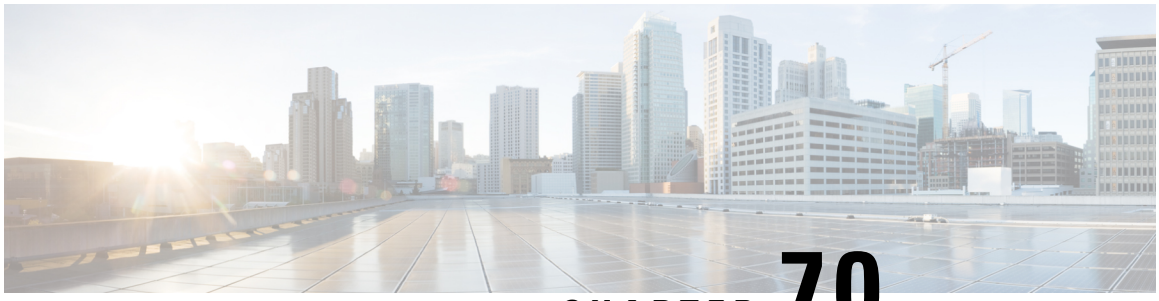
Feature Information for Frame Relay over L2TPv3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 136: Feature Information for Frame Relay over L2TPv3

Feature Name	Releases	Feature Information
Frame Relay over L2TPv3	Cisco IOS XE Release 3.12S	This feature enables Frame Relay switching over Layer 2 Tunnel Protocol Version 3 (L2TPv3). The feature works with like interfaces and disparate interfaces (L2VPN interworking).



CHAPTER 70

Loop-Free Alternate Fast Reroute with L2VPN

The Loop-Free Alternate (LFA) Fast Reroute (FRR) with Layer 2 Virtual Private Network (L2VPN) feature minimizes packet loss due to link or node failure.

Restrictions for Loop-Free Alternate Fast Reroute with L2VPN

- Load balancing is not supported
- Time-division multiplexing (TDM) pseudowire is not supported
- Virtual Private LAN Services (VPLS) is not supported
- The Virtual Private Wire Services (VPWS) scale number might change

Information About Loop-Free Alternate Fast Reroute with L2VPN

L2VPN Over Loop-Free Alternate Fast Reroute

The Loop-Free Alternate (LFA) Fast Reroute (FRR) feature offers an alternative to the MPLS Traffic Engineering Fast Reroute feature to minimize packet loss due to link or node failure. It introduces LFA FRR support for L2VPNs and Virtual Private Wire Services (VPWS), providing the following benefits:

- Same level of protection from traffic loss
- Simplified configuration
- Link and node protection
- Link and path protection

- LFA (loop-free alternate) paths
- Support for both IP and Label Distribution Protocol (LDP) core

LFA FRR enables a backup route to avoid traffic loss if a network fails. The backup routes (repair paths) are precomputed and installed in the router as the backup for the primary paths. After the router detects a link or adjacent node failure, it switches to the backup path to avoid traffic loss.

How to Configure Loop-Free Alternate Fast Reroute with L2VPN

To enable loop-free alternate fast reroute support for L2VPNs and VPWS, you must configure LFA FRR for the routing protocol. No additional configuration tasks are necessary. See one of the following documents, depending on the routing protocol:

- IS-IS Remote Loop-Free Alternate Fast Reroute in the *IP Routing: ISIS Configuration Guide*
- OSPFv2 Loop-Free Alternate Fast Reroute in the *IP Routing: OSPF Configuration Guide*
- OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute in the *IP Routing: OSPF Configuration Guide*

Verifying Loop-Free Alternate Fast Reroute with L2VPN

Use one or more of the following commands to verify the LFA FRR configuration:

SUMMARY STEPS

1. **show ip cef** *network-prefix* **internal**
2. **show mpls infrastructure lfd pseudowire internal**
3. **show platform hardware pp active feature cef database ipv4** *network-prefix*

DETAILED STEPS

Step 1 **show ip cef** *network-prefix* **internal**

Example:

```
show ip cef 16.16.16.16 internal
```

Displays entries in the Cisco Express Forwarding (CEF) Forwarding Information Base (FIB).

Step 2 **show mpls infrastructure lfd pseudowire internal**

Example:

```
show mpls infrastructure lfd pseudowire internal
```

Displays information about the Label Forwarding Database (LFD) and pseudowires.

Step 3 **show platform hardware pp active feature cef database ipv4** *network-prefix*

Example:

```
show platform hardware pp active feature cef database ipv4 16.16.16.16/32
```

Displays information about the CEF database.

Configuration Examples for Loop-Free Alternate Fast Reroute with L2VPN

Example: Verifying LFA FRR with L2VPN

show ip cef internal

The following example shows the configuration of LFA FRR for OSPF:

```
router ospf 1
 router-id 17.17.17.17
 fast-reroute per-prefix enable prefix-priority low
 network 3.3.3.0 0.0.0.255 area 1
 network 6.6.6.0 0.0.0.255 area 1
 network 7.7.7.0 0.0.0.255 area 1
 network 17.17.17.17 0.0.0.0 area 1
```

show ip cef internal

The following is sample output from the **show ip cef internal** command:

```
Device# show ip cef 16.16.16.16 internal
16.16.16.16/32, epoch 2, RIB[I], refcount 7, per-destination sharing
  sources: RIB, RR, LTE
  feature space:
   IPRM: 0x00028000
   Broker: linked, distributed at 1st priority
   LFD: 16.16.16.16/32 1 local label
   local label info: global/17
        contains path extension list
        disposition chain 0x3A3C1DF0
        label switch chain 0x3A3C1DF0
  subblocks:
   1 RR source [no flags]
    non-eos chain [16|44]
  ifnums:
   GigabitEthernet0/0/2(9): 7.7.7.2
   GigabitEthernet0/0/7(14): 7.7.17.9
  path 35D61070, path list 3A388FA8, share 1/1, type attached nexthop, for IPv4, flags
has-repair
    MPLS short path extensions: MOI flags = 0x20 label 16
  nexthop 7.7.7.2 GigabitEthernet0/0/2 label [16|44], adjacency IP adj out of
GigabitEthernet0/0/2, addr 7.7.7.2 35E88520
    repair: attached-nexthop 7.7.17.9 GigabitEthernet0/0/7 (35D610E0)
  path 35D610E0, path list 3A388FA8, share 1/1, type attached nexthop, for IPv4, flags
repair, repair-only
  nexthop 7.7.17.9 GigabitEthernet0/0/7, repair, adjacency IP adj out of GigabitEthernet0/0/7,
 addr 7.7.17.9 3A48A4E0
  output chain: label [16|44]
  FRR Primary (0x35D10F60)
```

```
  <primary:  TAG adj out of GigabitEthernet0/0/2, addr 7.7.7.2 35E88380>
  <repair:  TAG adj out of GigabitEthernet0/0/7, addr 7.7.17.9 3A48A340>
Rudy17#show mpls infrastructure lfd pseudowire internal
PW ID: 1VC ID: 4, Nexthop address: 16.16.16.16
SSM Class: SSS HW
Segment Count: 1
VCCV Types Supported:  cw ra ttl
Imposition details:
 Label stack {22 16}, Output interface: Gi0/0/2
 Preferred path: not configured
 Control Word: enabled, Sequencing: disabled
 FIB Non IP entry: 0x35D6CEEC
 Output chain:  AToM Imp (locks 4) label 22 label [16|44]
  FRR Primary (0x35D10F60)
  <primary:  TAG adj out of GigabitEthernet0/0/2, addr 7.7.7.2 35E88380>
Disposition details:
 Local label: 16
 Control Word: enabled, Sequencing: disabled
 SSS Switch: 3976200193
 Output chain:  mpls_eos( connid router-alert AToM Disp (locks 5)/ drop)
```

show mpls infrastructure lfd pseudowire internal

The following is sample output from the **show mpls infrastructure lfd pseudowire internal** command:

```
Device# show mpls infrastructure lfd pseudowire internal
PW ID: 1VC ID: 4, Nexthop address: 16.16.16.16
SSM Class: SSS HW
Segment Count: 1
VCCV Types Supported:  cw ra ttl
Imposition details:
 Label stack {22 16}, Output interface: Gi0/0/2
 Preferred path: not configured
 Control Word: enabled, Sequencing: disabled
 FIB Non IP entry: 0x35D6CEEC
 Output chain:  AToM Imp (locks 4) label 22 label [16|44]
  FRR Primary (0x35D10F60)
  <primary:  TAG adj out of GigabitEthernet0/0/2, addr 7.7.7.2 35E88380>
Disposition details:
 Local label: 16
 Control Word: enabled, Sequencing: disabled
 SSS Switch: 3976200193
 Output chain:  mpls_eos( connid router-alert AToM Disp (locks 5)/ drop)
```

show platform hardware pp active feature cef database

The following is sample output from the **show platform hardware pp active feature cef database** command:

```
Device# show platform hardware pp active feature cef database ipv4 16.16.16.16/32
=== CEF Prefix ===
16.16.16.16/32 -- next hop: UEA Label OCE (PI:0x104abee0, PD:0x10e6b9c8)
                Route Flags: (0)
                Handles (PI:0x104ab6e0) (PD:0x10e68140)

  HW Info:
    TCAM handle: 0x0000023f    TCAM index: 0x0000000d
    FID index  : 0x0000f804    EAID      : 0x0000808a
    MET        : 0x0000400c    FID Count : 0x00000000
```

```
=== Label OCE ===
  Label flags: 4
  Num Labels: 1
  Num Bk Labels: 1
  Out Labels: 16
  Out Backup Labels: 44
  Next OCE Type: Fast ReRoute OCE; Next OCE handle: 0x10e6f428

=== FRR OCE ===
  FRR type         : IP FRR
  FRR state        : Primary
  Primary IF's gid : 3
  Primary FID      : 0x0000f801
  FIFC entries     : 32
  PPO handle       : 0x00000000
  Next OCE         : Adjacency (0x10e63b38)
  Bkup OCE         : Adjacency (0x10e6e590)

=== Adjacency OCE ===
  Adj State: COMPLETE(0)   Address: 7.7.7.2
  Interface: GigabitEthernet0/0/2   Protocol: TAG
  mtu:1500, flags:0x0, fixups:0x0, encap_len:14
  Handles (adj_id:0x00000039) (PI:0x1041d410) (PD:0x10e63b38)
  Rewrite Str: d0:c2:82:17:8a:82:d0:c2:82:17:f2:02:88:47

  HW Info:
    FID index: 0x0000f486    EL3 index: 0x00001003    EL2 index: 0x00000000
    El2RW    : 0x00000107    MET index: 0x0000400c    EAID     : 0x00008060
    HW ADJ FLAGS: 0x40
    Hardware MAC Rewrite Str: d0:c2:82:17:8a:82:08:00:40:00:0d:02

=== Adjacency OCE ===
  Adj State: COMPLETE(0)   Address: 7.7.17.9
  Interface: GigabitEthernet0/0/7   Protocol: TAG
  mtu:1500, flags:0x0, fixups:0x0, encap_len:14
  Handles (adj_id:0x00000012) (PI:0x104acbd0) (PD:0x10e6e590)
  Rewrite Str: d0:c2:82:17:c9:83:d0:c2:82:17:f2:07:88:47

  HW Info:
    FID index: 0x0000f49d    EL3 index: 0x00001008    EL2 index: 0x00000000
    El2RW    : 0x00000111    MET index: 0x00004017    EAID     : 0x0000807d
    HW ADJ FLAGS: 0x40
    Hardware MAC Rewrite Str: d0:c2:82:17:c9:83:08:00:40:00:0d:07
```

Example: Configuring Remote LFA FRR with VPLS

Example: Configuration of Remote LFA FRR with Interior Gateway Protocol (IGP)

```
router isis hp
 net 49.0101.0000.0000.0802.00
 is-type level-2-only
 ispf level-2
 metric-style wide
 fast-flood
 set-overload-bit on-startup 180
 max-lsp-lifetime 65535
 lsp-refresh-interval 65000
 spf-interval 5 50 200
 prc-interval 5 50 200
```

```
 lsp-gen-interval 5 5 200
 no hello padding
 log-adjacency-changes
 nsf cisco
 fast-reroute per-prefix level-1 all
 fast-reroute per-prefix level-2 all
 fast-reroute remote-lfa level-1 mpls-ldp
 fast-reroute remote-lfa level-2 mpls-ldp
 passive-interface Loopback0
 mpls ldp sync
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-2
```

Example: Configuration of Remote LFA FRR with VPLS at the interface level.

```
!
interface GigabitEthernet0/3/3
 ip address 198.51.100.1 255.255.255.0
 ip router isis hp
 logging event link-status
 load-interval 30
 negotiation auto
 mpls ip
 mpls traffic-eng tunnels
 isis network point-to-point
end
!
```

Example: Configuration of remote LFA FRR with VPLS at the global level.

```
!
l2 vfi Test-2000 manual
 vpn id 2010
 bridge-domain 2010
 neighbor 192.0.2.1 encapsulation mpls
!
```

Example: Configuration of remote LFA FRR with VPLS at Access side.

```
!
interface TenGigabitEthernet0/2/0
 no ip address
 service instance trunk 1 ethernet
  encapsulation dot1q 12-2012
  rewrite ingress tag pop 1 symmetric
  bridge-domain from-encapsulation
 !
```

Example: Verifying Remote LFA FRR with VPLS

show ip cef internal

The following is sample output from the **show ip cef internal** command:

```
Router# show ip cef 198.51.100.2/32 internal

198.51.100.2/32, epoch 2, RIB[I], refcount 7, per-destination sharing
```

```
  sources: RIB, RR, LTE
  feature space:
   IPRM: 0x00028000
   Broker: linked, distributed at 1st priority
   LFD: 198.51.100.2/32 1 local label
   local label info: global/2033
        contains path extension list
        disposition chain 0x46764E68
        label switch chain 0x46764E68
  subblocks:
   1 RR source [heavily shared]
    non-eos chain [explicit-null|70]
  ifnums:
   TenGigabitEthernet0/1/0(15): 192.0.2.10
   MPLS-Remote-Lfa2(46)
  path 44CE1290, path list 433CF8C0, share 1/1, type attached nexthop, for IPv4, flags
has-repair
    MPLS short path extensions: MOI flags = 0x21 label explicit-null
  nexthop 192.0.2.10 TenGigabitEthernet0/1/0 label [explicit-null|70], adjacency IP adj out
 of TenGigabitEthernet0/1/0, addr 192.0.2.10 404B3960
    repair: attached-nexthop 192.0.2.1 MPLS-Remote-Lfa2 (44CE1300)
  path 44CE1300, path list 433CF8C0, share 1/1, type attached nexthop, for IPv4, flags
repair, repair-only
  nexthop 192.0.2.1 MPLS-Remote-Lfa2, repair, adjacency IP midchain out of MPLS-Remote-Lfa2
 404B3B00
  output chain: label [explicit-null|70]
  FRR Primary (0x3E25CA00)
  <primary:  TAG adj out of TenGigabitEthernet0/1/0, addr 192.168.101.22 404B3CA0>
  <repair:  TAG midchain out of MPLS-Remote-Lfa2 404B37C0 label 37 TAG adj out of
GigabitEthernet0/3/3, addr 192.0.2.14 461B2F20>
```

show ip cef detail

The following is sample output from the **show ip cef detail** command:

```
Router# show ip cef 198.51.100.2/32 detail

198.51.100.2/32, epoch 2
  local label info: global/2033
  1 RR source [heavily shared]
  nexthop 192.0.2.14 TenGigabitEthernet0/1/0 label [explicit-null|70]
    repair: attached-nexthop 192.0.2.1 MPLS-Remote-Lfa2
  nexthop 192.0.2.1 MPLS-Remote-Lfa2, repair
!
```

show platform hardware pp active feature cef databas

The following is sample output from the **show platform hardware pp active feature cef database** command:

```
Router# show platform hardware pp active feature cef database ipv4 198.51.100.2/32

=== CEF Prefix ===
198.51.100.2/32 -- next hop: UEA Label OCE (PI:0x10936770, PD:0x12dd1cd8)
               Route Flags: (0)
               Handles (PI:0x109099c8) (PD:0x12945968)

  HW Info:
    TCAM handle: 0x00000266    TCAM index: 0x00000015
    FID index  : 0x00008e7f    EAID      : 0x0001d7c4
    MET        : 0x0000401c    FID Count : 0x00000000
```

```
=== Label OCE ===
  Label flags: 4
  Num Labels: 1
  Num Bk Labels: 1
  Out Labels: 0
  Out Backup Labels: 70
=== FRR OCE ===
  FRR type         : IP FRR
  FRR state        : Primary
  Primary IF's gid : 52
  Primary FID      : 0x00008cb6
  FIFC entries     : 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0
  PPO handle       : 0x00000000
  Next OCE         : Adjacency (0x130e0df0)
  Bkup OCE         : Adjacency (0x130de608)

=== Adjacency OCE ===
  Adj State: COMPLETE(0)   Address: 192.168.101.22
  Interface: TenGigabitEthernet0/1/0   Protocol: TAG
  mtu:1500, flags:0x0, fixups:0x0, encap_len:14
  Handles (adj_id:0x000016ac) (PI:0x1090cc10) (PD:0x130e0df0)
  Rewrite Str: 18:33:9d:3d:83:10:c8:f9:f9:8d:04:10:88:47
HW Info:
    FID index: 0x00008e7e    EL3 index: 0x00001034    EL2 index: 0x00000000
    El2RW    : 0x0000010d    MET index: 0x00004012    EAID     : 0x0001d7c1
    HW ADJ FLAGS: 0x40
    Hardware MAC Rewrite Str: 18:33:9d:3d:83:10:08:00:40:00:0d:10
=== Adjacency OCE ===
  Adj State: COMPLETE(0)   Address: 0
  Interface: MPLS-Remote-Lfa2   Protocol: TAG
  mtu:17940, flags:0x40, fixups:0x0, encap_len:0
  Handles (adj_id:0xf80002e8) (PI:0x10da2150) (PD:0x130de608)
  Rewrite Str:

  HW Info:
    FID index: 0x00008ca8    EL3 index: 0x0000101c    EL2 index: 0x00000000
    El2RW    : 0x00000003    MET index: 0x00004024    EAID     : 0x0001d7cb
    HW ADJ FLAGS: 0x40
    Hardware MAC Rewrite Str: 00:00:00:00:00:00:00:00:00:00:00:00

=== Label OCE ===
  Label flags: 4
  Num Labels: 1
Num Bk Labels: 1
  Out Labels: 37
  Out Backup Labels: 37
  Next OCE Type: Adjacency; Next OCE handle: 0x12943a00
=== Adjacency OCE ===
  Adj State: COMPLETE(0)   Address: 30.1.1.1
  Interface: GigabitEthernet0/3/3   Protocol: TAG
  mtu:1500, flags:0x0, fixups:0x0, encap_len:14
  Handles (adj_id:0x0000378e) (PI:0x10909738) (PD:0x12943a00)
  Rewrite Str: c8:f9:f9:8d:01:b3:c8:f9:f9:8d:04:33:88:47

  HW Info:
    FID index: 0x00008c78    EL3 index: 0x0000101c    EL2 index: 0x00000000
    El2RW    : 0x00000109    MET index: 0x0000400e    EAID     : 0x0001cf4b
    HW ADJ FLAGS: 0x40
    Hardware MAC Rewrite Str: c8:f9:f9:8d:01:b3:08:00:40:00:0d:33
```

show mpls l2transport detail

The following is sample output from the **show mpls l2transport detail** command:

```
Router# show mpls l2transport vc 2000 detail

Local interface: VFI Test-1990 vfi up
  Interworking type is Ethernet
  Destination address: 192.0.2.1, VC ID: 2000, VC status: up
    Output interface: Te0/1/0, imposed label stack {0 2217}
    Preferred path: not configured
    Default path: active
    Next hop: 192.51.100.22
  Create time: 1d08h, last status change time: 1d08h
    Last label FSM state change time: 1d08h
  Signaling protocol: LDP, peer 192.0.51.1:0 up
    Targeted Hello: 192.51.100.2(LDP Id) -> 192.51.100.200, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Multiprotocol Label Switching Command Reference

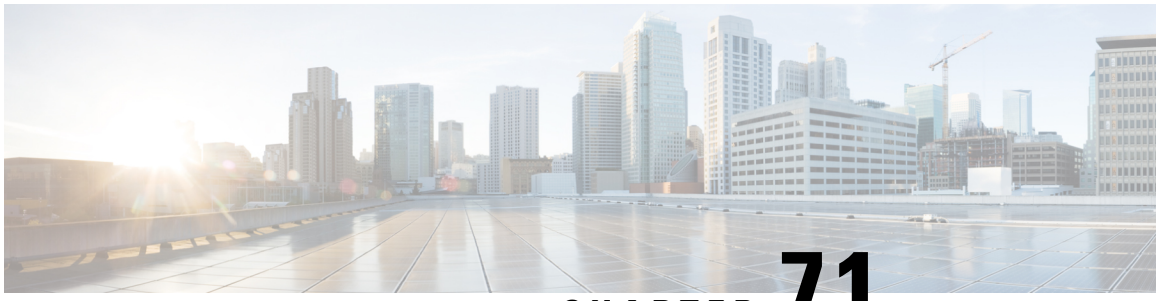
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Loop-Free Alternate Fast Reroute with L2VPN

Table 137: Feature Information for Loop-Free Alternate Fast Reroute with L2VPN

Feature Name	Releases	Feature Information
Loop-Free Alternate Fast Reroute with L2VPN	15.3(2)S Cisco IOS XE Release 3.9S Cisco IOS XE Release 3.10 S	This feature introduces loop-free alternate (LFA) fast reroute (FRR) support for Layer 2 VPN (L2VPN) and Virtual Private Wire Services (VPWS) to minimize packet loss due to link or node failure. No commands were introduced or modified. In Cisco IOS XE Release 3.9S, support was added for the Cisco ASR 903 Router. In Cisco IOS XE Release 3.10S, Remote LFA FRR is supported on ATM (IMA) and TDM pseudowires for the Cisco ASR 903 Router. In Cisco IOS XE Release 3.10S, Remote LFA FRR is supported over VPLS for Cisco ASR 903 Router.



CHAPTER 71

EVPN Single-Homing

The EVPN Single-Homing feature utilizes the functionality defined in RFC 7432 (BGP MPLS-based Ethernet VPN), to achieve single-homing between a Provider Edge (PE) and a Customer Edge (CE) device.

Information about EVPN Single-Homing

Ethernet Multipoint Connectivity

To achieve Ethernet multipoint connectivity, MPLS deployments traditionally rely on Virtual Private LAN Services (VPLS). A VPLS service is built with a full-mesh of pseudowires between PE devices that are part of a Layer 2 broadcast domain. A VPLS PE device performs data-plane MAC learning. For MAC learning, the VPLS PE device uses local interfaces for traffic coming from the access network and uses pseudowires for the traffic coming from the core network.

EVPN Multipoint Solution

EVPN is the next generation of multipoint L2VPN solution that aligns operation principles of L3VPN with Ethernet services. Instead of relying solely on data plane for MAC Address learning, EVPN PE devices signal and learn MAC addresses over the core network using BGP, while still using data plane MAC-learning on the access side. Providers can configure BGP as a common VPN control plane for their ethernet offerings and leverage the advantages of Layer 3 VPN over VPLS.

EVPN Building Blocks

There are three fundamental building blocks for EVPN technology, EVPN Instance (EVI), Ethernet Segment (ES), EVPN BGP routes and extended communities:

- EVI is a VPN connection on a PE router. It is the equivalent of IP VPN Routing and Forwarding (VRF) in Layer 3 VPN. It is also known as MAC-VRF.
- ES is a connection with a customer site (device or network) and is associated with access-facing interfaces. Access-facing interfaces are assigned unique IDs that are referred to as Ethernet Segment Identifiers (ESI). A site can be connected to one or more PEs. The ES connection has the same ESI in each PE connected to the site.
- RFC 7432 defines routes and extended communities to enable EPVN support. In Cisco IOS XE Fuji 16.8.x Software Release, Route Type 2 and Route Type 3 are supported.

  In BGP MPLS-based EVPN, an EVI is configured for every PE device for each customer associated with the PE device. In this case, a customer is any customer edge device that is attached to the PE device. The CE device can be a host, a switch or a router. Each EVI has a unique Route Distinguisher (RD) and one or more Route Targets (RT).

  For EVPN Single-Homing feature, a CE device is attached to a single PE device and has an Ethernet Segment with ESI=0.

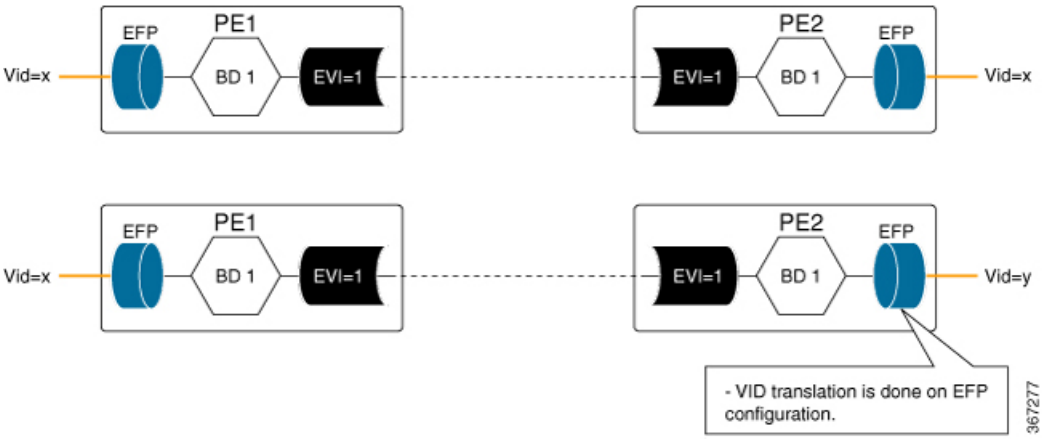
Service Interfaces

The following are types of EVPN VLAN service interfaces:

VLAN-based Service Interface

In VLAN-based service interface, each VLAN is associated to one bridge domain and one EVI.

Figure 95: VLAN-Based Service Interface



For VLAN-based Service Interface, Type 1 Route Distinguisher, a unique number used to distinguish identical routes in different VRFs, is used for EVIs as recommended by the RFC 7432. The Route Distinguishers and Router Targets, which are used to share routes between different VRFs, are autogenerated to ensure unique Route Distinguisher numbers across EVIs.

VLAN Bundle Service Interface

In VLAN Bundle Service Interface, multiple VLANs share the same bridge table.

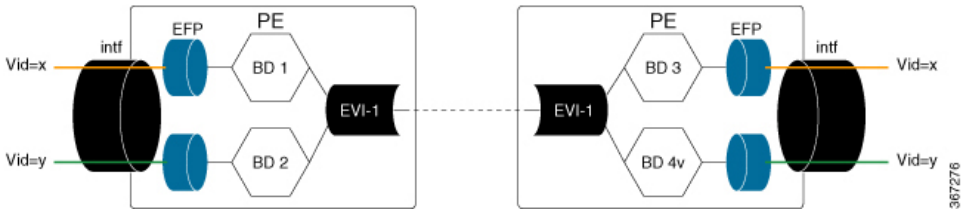
Figure 96: VLAN Bundle Service Interface

Each EVPN instance corresponds to multiple broadcast domains maintained in a single bridge table per MAC-VRF. For VLAN Bundle Service Interface service to work, MAC addresses must be unique across all VLANs for an EVI.

VLAN-Aware Bundle Service Interface

For VLAN-aware Bundle Service Interface, each VLAN is associated with one bridge domain, but there can be multiple bridge domains associated with one EVI.

Figure 97: VLAN-Aware Bundle Service Interface



An EVPN instance consists of multiple broadcast domains where each VLAN has one bridge table. Multiple bridge tables (one per VLAN) are maintained by a single MAC-VRF that corresponds to the EVPN instance.

Route Types

For EVPN Single-Homing feature, Route Type 2 and Route Type 3 are supported, as defined by RFC 7432.

Route Type 2 — MAC and IP Advertisement Route

Type 2 Routes are used to advertise MAC addresses and their associated IP addresses. When a PE router learns the MAC address of a CE device that is connected to it locally, or a MAC address of a device behind the CE device, a MAC and an IP advertisement route is created.

The following table describes the header format for the MAC and IP Advertisement Route packet:

Table 138: Header format for the MAC and IP Advertisement Route packet

Field	Value	Length (Octets)
Route Type	0x02	1
Length	Variable	1
EVI RD	Type 1 (IPv4 address) RD unique across all EVIs on the PE	8
ESI	Ethernet Segment Identifier	10
Ethernet Tag	0 or valid Ethernet Tag	4

Field	Value	Length (Octets)
MAC Addr Len	48	1
MAC Address	Valid MAC address	6
IP Addr Length	IP address length in bits: 0, 32 or 128	1
IP Address	Optional IP address	0 or 4 or 16
Label1	Valid downstream assigned label to perform forwarding to a CE device based on the destination MAC address	3
Label2	Specifies a second label	0-3
EVI RT	Type 0 (2byteAS) route target	8

Note

- MAC Address field is populated with the CE address.
- IP address field is optional with IP Address length set to 0 bits.
- For EVPN Single-Homing feature, ESI value is always set to 0.
- In the Label field (Label1, Label2), Per-BD or Per-CE labels can be assigned.
  - Per-BD is used when PE advertises a single label for all MAC addresses learned in a given bridge domain.
  - Per-CE label assigns a separate label to each access port in the bridge domain.

Route Type 3 — Inclusive Multicast Ethernet Tag Route

Type 3 routes are used for transporting Broadcast, Unknown Unicast, and Multicast (BUM) traffic to other PE devices across a given EVPN network instance.

The following tables describes the header format for Type 3 routes:

Table 139: Header Format for Type 3 Route Packets

Field	Value	Length (Octets)
Route Type	0x03	1
Length	26 or 38	1
EVI RD	Type 1 (IPv4Addr) RD unique across all EVIs on the PE	8
Ethernet Tag	0 or valid Ethernet Tag	4

Field	Value	Length (Octets)
IP Addr Length	IP Address Length - 32 bits or 128 bits	1
IP Address	IP Address common for all EVIs (for example, loopback address)	4 or 16
PMSI Tunnel Attr	{1 byte flags = 0}:{1 byte Tunnel Type}:{3 byte label}:{variable length Tunnel Identifier}	Variable
EVI RT	Type 0 (2byteAS) route target	8

The PE devices advertise an Inclusive Multicast Ethernet Tag (IMET) Route for every EVI-Ethernet Tag sequence. The Ethernet Tag is set to 0 for VLAN-based and VLAN-bundling service interfaces. The Ethernet Tag is set to a valid VLAN ID for VLAN-aware bundling service interface.

Type 3 route also carries a Provider Multicast Service Interface (PMSI) Tunnel attribute as specified in RFC 6514 (BGP Encodings and Procedures for MVPNs).

For Ingress Replication, the IMET route is used to advertise the label (in the PMSI Tunnel Attribute) that the other PEs can use to send BUM traffic to the originating PE device.

Prerequisites for EVPN Single-Homing

- EVI and Bridge domains must be in established state with associated MPLS labels.

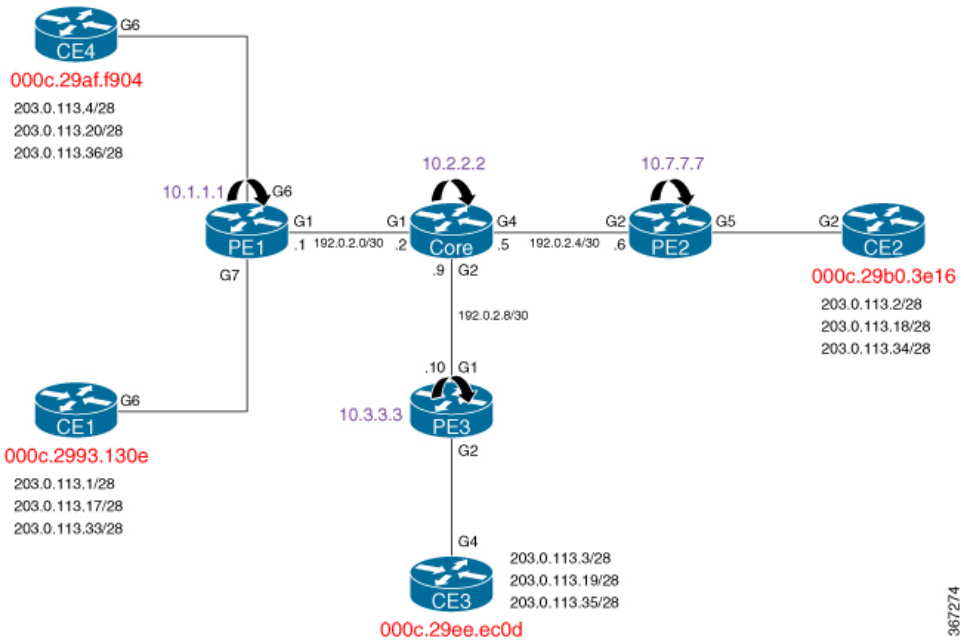
Restrictions for EVPN Single-Homing

- Route Type 1 and Route Type 4 are not supported.
- Per-EVI-based labelling is not supported.
- Maximum number of supported bridge domains is 1600.
- Maximum number of supported EEPs or service instances is 8000.
- Stateful Switchover is not supported.
- Single-Homing feature is not supported with port channel interface between Provider Edge and Customer Edge devices.
- If want to create a VLAN-bundle or VLAN-aware EVI's, they must be configured before adding to a bridge domain (or VLAN).
- MAC mobility with duplicate MAC detection is not supported.
- ESI must be all 0s.

How to Configure EVPN Single Homing

Configuring EVPN

Figure 98: EVPN Single Homing



The above figure represents a simple EVPN network. Use the following steps to configure EVPN:

EVPN Configuration

```
enable
    configure terminal
        l2vpn evpn
          replication-type ingress
          router-id Loopback1
          mpls label mode per-ce
         !
        l2vpn evpn instance 10 vlan-based
          route-distinguisher 10.1.1.1:10
          route-target both 10:10
          no auto-route-target
         !
        bridge-domain 10
        member evpn-instance 10
          member GigabitEthernet 0/0/1 service-instance 10
        !
        interface GigabitEthernet 0/0/1
          no ip address
          service instance 10 ethernet
           encapsulation dot1q 200
           ¡
          ¡
        !
```

Configuring L2VPN EVPN Globally and EVI on IOS-XE Router

```
l2vpn evpn
 replication-type ingress    ----> Enables ingress replication label
!
l2vpn evpn instance 10 vlan-based ---> Configures Vlan-based EVI 10
!
l2vpn evpn instance 20 vlan-bundle ---→ Configures Vlan-bundled EVI 20
!
l2vpn evpn instance 30 vlan-aware   ---→ Configures Vlan-aware EVI 30
```

Configuring Bridge Domains on IOS-XE Router

```
bridge-domain 10
 mac aging-time 30
 member GigabitEthernet6 service-instance 10  --→ Links SI 10 on interface with Bridge-domain
 10
 member evpn-instance 10 --→ Links EVI 10 with Bridge-domain 10
!
bridge-domain 20
 mac aging-time 30
 member GigabitEthernet6 service-instance 20 --→ Links SI 20 on interface with Bridge-domain
 20
 member evpn-instance 20 --> Links EVI 20 with Bridge-domain 20
!
bridge-domain 30
 mac aging-time 30
 member GigabitEthernet6 service-instance 30 --→ Links SI 30 on interface with Bridge-domain
 30
 member evpn-instance 30 ethernet-tag 30 ---> Links EVI 30 with Bridge-domain 30
```

Configuring Access Interface on a Provider Edge

```
interface GigabitEthernet6
 no ip address
 negotiation auto
 service instance 10 ethernet ----> Enables service instance 10 under the physical interface

  encapsulation dot1q 10
 !
 service instance 20 ethernet ----> Enables service instance 20 under the physical interface

  encapsulation dot1q 20-21
 !
 service instance 30 ethernet ----> Enables service instance 30 under the physical interface

  encapsulation dot1q 30
```

Configuring EVPN Single-Homing

Use the following steps to configure EVPN Single-Homing:

Configuring BGP on Provider Edge Device, PE1

```
enable
configure terminal
 router bgp 100
  bgp router-id 10.1.1.1
  bgp log-neighbor-changes
```

```
 bgp graceful-restart
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 update-source Loopback0
!
address-family ipv4
 neighbor 10.2.2.2 activate
exit-address-family
!
address-family l2vpn evpn     ----> Enables L2VPN EVPN address family
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community both
 neighbor 10.2.2.2 soft-reconfiguration inbound
exit-address-family
```

Configuring BGP on Route Reflector

```
router bgp 100
 bgp router-id 10.2.2.2
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 update-source Loopback0
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.3.3.3 update-source Loopback0
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.7.7.7 update-source Loopback0
 !
 address-family ipv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 route-reflector-client
  neighbor 10.7.7.7 activate
  neighbor 10.7.7.7 route-reflector-client
 exit-address-family
 !
 address-family l2vpn evpn      ----> Enables L2vpn evpn address family
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 send-community both
  neighbor 10.1.1.1 route-reflector-client
  neighbor 10.1.1.1 soft-reconfiguration inbound
  neighbor 10.3.3.3 activate
  neighbor 10.3.3.3 send-community both
  neighbor 10.3.3.3 route-reflector-client
  neighbor 10.3.3.3 soft-reconfiguration inbound
  neighbor 10.7.7.7 activate
  neighbor 10.7.7.7 send-community both
  neighbor 10.7.7.7 route-reflector-client
  neighbor 10.7.7.7 soft-reconfiguration inbound
 exit-address-family
```

Configuring Customer Edge and Provider Edge Interfaces

CE1 configuration

```
interface GigabitEthernet6.10
 encapsulation dot1Q 10
 ip address 203.0.113.1 255.255.255.240
interface GigabitEthernet6.20
 encapsulation dot1Q 20
 ip address 203.0.113.17 255.255.255.240
interface GigabitEthernet6.30
 encapsulation dot1Q 30
 ip address 203.0.113.33 255.255.255.240
```

PE1 Configuration

```
interface GigabitEthernet6
 no ip address
 negotiation auto
 service instance 10 ethernet
  encapsulation dot1q 10
 !
 service instance 20 ethernet
  encapsulation dot1q 20-21
 !
 service instance 30 ethernet
  encapsulation dot1q 30
```

Verification Examples for EVPN Single-Homing

Use the following command to verify that EVI and Bridge domains are in established state and to display associated MPLS labels:

```
show l2vpn evpn evi detail
EVPN instance:    10 (VLAN Based)    ----> VLAN Based EVI
  RD:             10.1.1.1:10 (auto) ----> RD derived from Loopback0 of PE1
  Import-RTs:     100:10
  Export-RTs:     100:10
  Per-EVI Label:  none
  State:          Established    ----> EVI state
  Encapsulation:  mpls
  Bridge Domain:  10
    Ethernet-Tag: 0
    BUM Label:    23             ----> Broadcast/Unknown unicast/Multicast traffic label
    Per-BD Label: 22
    State:        Established     ----> Bridge-domain state
    Pseudoports:
      GigabitEthernet6 service instance 10 ----> Local interface part of bridge-domain
      GigabitEthernet7 service instance 10   ---→ Local interface part of bridge-domain

EVPN instance:    20 (VLAN Bundle) ----> VLAN Bundled EVI
  RD:             10.1.1.1:20 (auto)
  Import-RTs:     100:20
  Export-RTs:     100:20
  Per-EVI Label:  none
  State:          Established
  Encapsulation:  mpls
  Bridge Domain:  20
    Ethernet-Tag: 0
    BUM Label:    20
    Per-BD Label: 21
    State:        Established
    Pseudoports:
      GigabitEthernet6 service instance 20
      GigabitEthernet7 service instance 20

EVPN instance:    30 (VLAN Aware)  ----> VLAN-Aware EVI
  RD:             10.1.1.1:30 (auto)
  Import-RTs:     100:30
  Export-RTs:     100:30
  Per-EVI Label:  none
  State:          Established
  Encapsulation:  mpls
  Bridge Domain:  30
```

```
    Ethernet-Tag: 30
    BUM Label:    18
    Per-BD Label: 19
    State:        Established
    Pseudoports:
      GigabitEthernet6 service instance 30
      GigabitEthernet7 service instance 30
```

Use the following command to verify that the bridge domain has learnt the local and remote MAC addresses:

```
PE1#show bridge-domain 10
Bridge-domain 10 (3 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 30 second(s)  ----> MAC aging timer for bridge-domain
    GigabitEthernet6 service instance 10
    GigabitEthernet7 service instance 10
    EVPN Instance 10
   AED MAC address    Policy  Tag       Age  Pseudoport
   -   000C.29B0.3E16 forward static_r  0    OCE_PTR:0xe8eb04a0 ---→ Remotely learnt MAC
   -   000C.29AF.F904 forward dynamic_c 29   GigabitEthernet6.EFP10 --→ MAC locally learnt

   -   000C.2993.130E forward dynamic_c 26   GigabitEthernet7.EFP10
   -   000C.29EE.EC0D forward static_r  0    OCE_PTR:0xe8eb0500
```

Note In the above output, MAC addresses with forward dynamic_c tags are locally learned addresses and MAC addresses with forward static_r tags are remote addresses learned through EVPN.

Use the following command to verify that EVPN manager has received the local MACs learned by the bridge domain:

```
PE1# show l2vpn evpn mac
MAC Address    EVI   BD    ESI                       Ether Tag  Next Hop
-------------- ----- ----- ------------------------ ---------- ---------------
000c.2993.130e 10    10    0000.0000.0000.0000.0000  0          Gi7:10
000c.29af.f904 10    10    0000.0000.0000.0000.0000  0          Gi6:10
000c.29b0.3e16 10    10    0000.0000.0000.0000.0000  0          10.7.7.7
000c.29ee.ec0d 10    10    0000.0000.0000.0000.0000  0          10.3.3.3

PE1# show l2vpn evpn mac detail
MAC Address:               000c.2993.130e
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:          0000.0000.0000.0000.0000
Ethernet Tag ID:           0
Next Hop(s):               GigabitEthernet7 service instance 10
Label:                     22
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running

MAC Address:               000c.29ee.ec0d
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:          0000.0000.0000.0000.0000
Ethernet Tag ID:           0
Next Hop(s):               10.3.3.3
Local Address:             10.1.1.1
```

```
Label:                     19
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running
```

Note In the above output, the next hop address of the remote MAC is the address of the provider edge device, if it is learnt remotely or the local interface if MAC address is learnt locally.

Use the following command to verify that Layer 2 Routing Information Base (RIB) has the required the MAC info:

```
PE1# show l2vpn l2route evpn mac
  EVI       ETag  Prod    Mac Address                    Next Hop(s) Seq Number
----- ---------- ----- -------------- ---------------------------- ----------
   10          0 L2VPN 000C.2993.130E                       Gi7:10          0
   10          0 L2VPN 000C.29AF.F904                       Gi6:10          0
   10          0   BGP 000C.29B0.3E16             L:19 IP:10.7.7.7          0
   10          0   BGP 000C.29EE.EC0D             L:19 IP:10.3.3.3          0
```

Note Remote MACs are learnt through BGP. In the above command output, the producer is BGP and local MACs are learned through Layer 2 VPN.

Use the following command to verify that Layer 2 FIB has received the MAC information from Layer 2 RIB, and bridge-domain and MFI are configured.

```
PE1# show l2fib bridge-domain 10 detail
Bridge Domain : 10
  Reference Count : 18
  Replication ports count : 4
  Unicast Address table size : 4
  IP Multicast Prefix table size : 4

  Flood List Information :
    Olist: Id 9225, Port Count 4

  Port Information :
    Serv Inst: Gi6:10
    Serv Inst: Gi7:10
    EVPN MPLS Encap: pathlist 107
    EVPN MPLS Encap: pathlist 101

  Unicast Address table information :
    Mac: 000c.2993.130e, Adjacency: Serv Inst: Gi7:10
    Mac: 000c.29af.f904, Adjacency: Serv Inst: Gi6:10
    Mac: 000c.29b0.3e16, Adjacency: EVPN MPLS Encap: pathlist 98
    Mac: 000c.29ee.ec0d, Adjacency: EVPN MPLS Encap: pathlist 104

  IP Multicast Prefix table information :
    Source: *, Group: 224.0.0.0/4, IIF:  , Adjacency: Olist: 9226, Ports: 0
    Source: *, Group: 224.0.0.0/24, IIF:  , Adjacency: Olist: 9225, Ports: 4
    Source: *, Group: 224.0.1.39, IIF:  , Adjacency: Olist: 9225, Ports: 4
    Source: *, Group: 224.0.1.40, IIF:  , Adjacency: Olist: 9225, Ports:
```

Use the following command to verify that the information on BGP route type 3 is sent to L2RIB:

```
PE1# show l2vpn l2route evpn imet
  EVI       ETAG  Prod  Router IP Addr  Type   Label       Tunnel ID
----- ---------- ----- --------------- ----- ------- ---------------
   10          0   BGP         10.3.3.3     6      18         10.3.3.3
   10          0   BGP         10.7.7.7     6      18         10.7.7.7
   10          0 L2VPN         10.1.1.1     6      23         10.1.1.1
```

Use the following command to verify MPLS forwarding:

```
PE1#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
18         No Label   evpn(mc:bd 30)   305042        none       point2point
19         No Label   evpn(uc:bd 30)   7684          none       point2point
20         No Label   evpn(mc:bd 20)   542588        none       point2point
21         No Label   evpn(uc:bd 20)   13786         none       point2point
22         No Label   evpn(uc:bd 10)   6638          none       point2point
23         No Label   evpn(mc:bd 10)   277740        none       point2point
24         Pop Label  192.0.2.2-A      0             Gi1        192.0.2.2
25         Pop Label  192.0.2.2-A      0             Gi1        192.0.2.2
16001      16001      10.3.3.3/32       0             Gi1         192.0.2.2
16002      Pop Label  10.2.2.2/32       0             Gi1         192.0.2.2
16004      16004      10.7.7.7/32       0             Gi1         192.0.2.2

PE1# show ip bgp l2vpn evpn route-type 2
BGP routing table entry for [2][10.1.1.1:10][0][48][000C2993130E][0][*]/20, version 43
Paths: (1 available, best #1, table evi_10)
  Advertised to update-groups:
     2
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (10.1.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      EVPN ESI: 00000000000000000000, Label1 22
      Extended Community: RT:100:10
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [2][10.1.1.1:10][0][48][000C29B03E16][0][*]/20, version 116
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [2][10.7.7.7:10][0][48][000C29B03E16][0][*]/20
 (global)
    10.7.7.7 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 19
      Extended Community: RT:100:10
      Originator: 10.7.7.7, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [2][10.1.1.1:10][0][48][000C29B03E16][0][*]/20, version 116
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [2][10.7.7.7:10][0][48][000C29B03E16][0][*]/20
 (global)
    10.7.7.7 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 19
      Extended Community: RT:100:10
      Originator: 10.7.7.7, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [2][10.1.1.1:10][0][48][000C29EEEC0D][0][*]/20, version 134
```

```
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [2][10.3.3.3:10][0][48][000C29EEEC0D][0][*]/20
 (global)
    10.3.3.3 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 19
      Extended Community: RT:100:10
      Originator: 10.3.3.3, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0


PE1# show ip bgp l2vpn evpn route-type 3
BGP routing table entry for [3][10.1.1.1:10][0][32][10.1.1.1]/17, version 41
Paths: (1 available, best #1, table evi_10)
  Advertised to update-groups:
     2
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (10.1.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:100:10
      PMSI Attribute: for EVPN, Flags: 0x0, Tunnel type: 6, length 4, label: 23 (vni 368)
tunnel parameters: 0101 0101
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [3][10.1.1.1:10][0][32][10.3.3.3]/17, version 137
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [3][10.3.3.3:10][0][32][10.3.3.3]/17 (global)

    10.3.3.3 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:10
      Originator: 10.3.3.3, Cluster list: 10.2.2.2
      PMSI Attribute: for EVPN, Flags: 0x0, Tunnel type: 6, length 4, label: 18 (vni 288)
tunnel parameters: 0303 0303
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [3][10.1.1.1:10][0][32][10.7.7.7]/17, version 122
Paths: (1 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 3
  Local, (received & used), imported path from [3][10.7.7.7:10][0][32][10.7.7.7]/17 (global)

    10.7.7.7 (metric 30) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:10
      Originator: 10.7.7.7, Cluster list: 10.2.2.2
      PMSI Attribute: for EVPN, Flags: 0x0, Tunnel type: 6, length 4, label: 18 (vni 288)
tunnel parameters: 0707 0707
      rx pathid: 0, tx pathid: 0x0
```

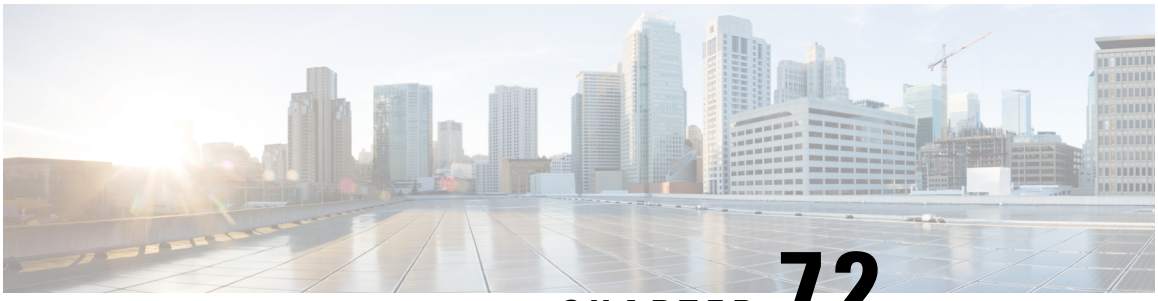
Additional References for EVPN Single-Homing

Standards and RFCs

Standard	Title
RFC 7432	BGP MPLS-Based Ethernet VPN

Feature Information for EVPN Single-Homing

Feature Name	Releases	Feature Information
EVPN Single-Homing	Cisco IOS XE Fuji 16.8.x	The EVPN Single-Homing feature utilizes the BGP MPLS-based Ethernet VPN (EVPN) functionality to achieve single-homing between a Provider Edge and a Customer Edge device. The following command was introduced or modified: address-family l2vpn, l2vpn evpn, member (bridge-domain), show ip bgp l2vpn evpn, show l2vpn evpn, show l2vpn l2route



CHAPTER 72

EVPN Multihoming

The EVPN Multihoming feature utilizes the functionality defined in RFC 7432 (BGP MPLS-based Ethernet VPN) to achieve multihoming between Provider Edge (PE) and Customer Edge (CE) devices.

Information about EVPN Multihoming

BGP MPLS-based EVPN

Ethernet VPN (EVPN) is an evolution of the L2VPN VPLS solution that addresses the following requirements:

- PE node redundancy with load-balancing based on Layer 2, Layer 3, or Layer 4 flows from CE to PE.
- Flow-based multi-pathing of traffic from local PE to remote PEs across core and vice-versa.
- Geographically redundant PE nodes with optimum unicast forwarding.
- Flexible redundancy grouping, where a PE can be a member of multiple redundancy groups each containing a different set of CEs.

There are three fundamental building blocks for EVPN technology - EVPN Instance (EVI), Ethernet Segment (ES), and EVPN BGP routes and extended communities. For more information, refer to EVPN Building Blocks section.

In BGP MPLS-based EVPN, an EVI is configured for every PE device for each customer associated with the PE device. An example of a customer is the CE device that is attached to the PE device. Each EVI has a unique Route Distinguisher (RD) and one or more Route Targets (RT). The CE device can be a host, a switch or a router.

For any port involved in a multihoming CE configuration, an ESI must be defined and associated with it. In Cisco IOS XE Fuji 16.9.x software release, only type 3 ESI is supported as defined in section 5 of RFC7432. Type 3 ESI consists of PE System MAC address and local discriminator.

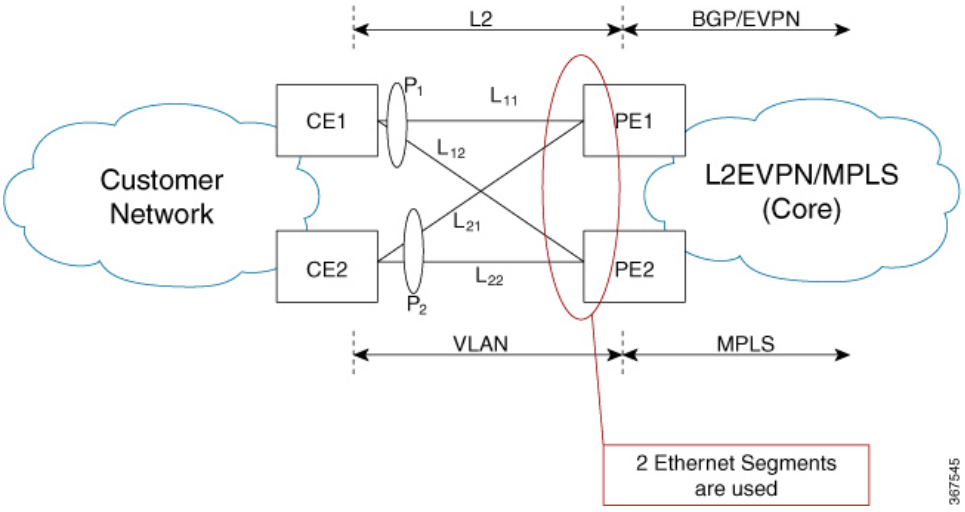
In EVPN multihoming, a customer site is connected to multiple PE devices and can have an Ethernet Segment with ESI value greater than one.

RFC7432 defines four new routes and four new extended communities to enable EPVN support. From Cisco IOS XE Fuji 16.9.x Software Release, all four route types are supported.

EVPN Multihoming Topology

The following figure shows a typical deployment involving two CE devices, where each CE device connects to multiple PE devices, to mitigate any single-point failures:

Figure 99: EVPN Multihoming Topology



- CE1 uses a port channel consisting of links, L11 and L12, to connect to PE1 and PE2, respectively.
- CE2 uses a port channel consisting of links L21 and L22, to connect to PE1 and PE2, respectively.
- On PE1 and PE2, ESI-1 is used to identify Ethernet Flow Points (EFPs) corresponding to links from P1, and ESI-2 is used to identify EFPs corresponding to links from P2.

Note Since CE1 and CE2 are port channels, each port channel can support flow-based load balancing for traffic egress towards PE1 and PE2.

Note For each PE, ESI is a property associated to a port.

All-Active Multihoming

EVPN Multihoming access gateway enables redundant network connectivity by allowing a CE device to connect to more than one PE device. Disruptions to the network connectivity are prevented by allowing a CE device to be connected to a PE device or several PE devices through multihoming. Ethernet segment is the group of ethernet links through which a CE device is connected to more than one PE devices. The All-Active Link Aggregation Group (LAG) bundle operates as an ethernet segment.

In all-active multihoming scenario, when multihop is configured to the same destination, the access side device load balances traffic on the access side and the PEs load balance traffic to remote PEs on the core side.

Route Types

RFC7432 introduces four new BGP route types (1–4) and communities.

- In EVPN multihoming scenarios, route types 1 and 4 are advertised to discover other PEs and their redundancy modes.
- Route type 2 is used for MAC learning. EVPN introduces the concept of BGP MAC routing and uses Multiprotocol-BGP (mBGP) for learning the MAC addresses between the PEs.

Route Type 1 - Ethernet Auto-Discovery Route

The route type value for EAD routes is 0x01. This route is originated when a PE is connected to a CE for which multihoming is configured. Two types of EAD routes are supported in this feature: Per-EVI (EVPN Instance) EAD routes and Per-ES (Ethernet Segment) EAD Routes.

Route Type-1 advertisement is used for achieving split-horizon, fast convergence, and aliasing. EAD-ES and EAD-EVI are used to achieve these functionalities. Fast convergence allows PEs to change the next-hop adjacencies for all MACs associated with an ES and aliasing allows balancing traffic across multiple egress points. Route Type 1 is advertised only if ES is set to a non-zero value, that is, type 1 routes are originated only for sites where multihoming is configured. These routes are sent per-ES and carry the combined set of route targets of all of the EVIs that belong to that ES.

The per-ES EAD route includes the ESI label extended community which indicates if it is an all-active or a single-active configuration. The ESI label extended community also carries the ESI label that is used for split horizon configuration. The per-ES EAD route is also used for fast convergence when failure occurs at the ES on the access side.

The per-EVI EAD and per-ES EAD routes are used for aliasing, and fast convergence and providing the split horizon label, respectively. In a multihoming group, each PE associated with a CE may learn only a subset of MAC addresses on traffic ingress from CE. The MAC addresses learned by the PEs may not overlap with each other. Aliasing is the ability of a PE to signal that it has reachability to an EVPN instance on a given ES, even when the PE has not learned MAC addresses from that EVI or ES. In an all-active multi-homing configuration, a remote PE that receives a MAC advertisement route considers the advertised MAC address to be reachable through all PEs that have advertised reachability to EVI or ES of the MAC address.

Table 140: Per-EVI Ethernet Auto-Discovery Route

Field	Value	Length (Octets)
Route Type	0x01	1
Length	25	1

Field	Value	Length (Octets)
EVI RD	Type 1 (IPv4Addr) RD unique across all EVIs on the PE.	8
ESI	Ethernet Segment Identifier	10
Ethernet Tag	0 or valid Ethernet Tag	4
Label	Valid MPLS label allocated per [EVI, ESI, EtherTag] tuple	3
EVI RT	Type 0 (2byteAS) route target	8

The route target is specific to the EVI. It can be automatically derived from EVI and AS numbers, or explicitly configured. As in L2VPN and L3VPN, multiple route targets can be configured for an EVPN instance (EVI) and in this case multiple route target extended communities are attached to the per-EVI EAD route.

Following is the header format of the Per-ES EAD route:

Table 141: Per-ES Ethernet Auto-Discovery Route

Field	Value	Length (Octets)
Route Type	0x01	1
Length	25	1
ES RD	Type 1 (IPv4Addr) RD.	8
ESI	Ethernet Segment Identifier	10
Ethernet Tag	0xFFFFFFFF	4
Label	0	3
ESI Label	0x0601:{1 byte single-active bit}:0x0000:{Split-Horizon label}	8
EVI-1 RT	Type 0 (2byteAS) route target for EVI-1	8
EVI-2 RT	Type 0 (2byteAS) route target for EVI-2	8
...	...	...
EVI-n RT	Type 0 (2byteAS) route target for EVI-n	8

One per-ES-EAD route is sourced per Ethernet Segment. Per-ES-EAD route carries the route targets of all EVIs the Ethernet Segment belongs to. If the number of EVI route targets is too large to be carried in one per-ES-EAD route, then multiple routes are advertised. Each route is assigned a different Ethernet Segment Route Distinguisher (ES-RD). The per-EVI-EAD route is used along with the per-ES-EAD route for aliasing and backup path. The per-ES-EAD is also used for fast convergence in case of failure in the Ethernet Segment.

Route Type 2 - MAC and IP Advertisement Route

Type 2 routes are used to advertise MAC addresses and their associated IP addresses. When a PE router learns the MAC address of a CE device that is connected to it locally, or a MAC address of a device behind the CE device, a MAC and IP advertisement route is created.

Following is the header format for the MAC and IP Advertisement Route packet:

Table 142: Header format for the MAC and IP Advertisement Route packet

Field	Value	Length (Octets)
Route Type	0x02	1
Length	Variable	1
EVI RD	Type 1 (IPv4 address) RD unique across all EVIs on the PE.	8
ESI	Ethernet Segment Identifier	10
Ethernet Tag	0 or valid Ethernet Tag	4
MAC Addr Len	48	1
MAC Address	Valid MAC address	6
IP Addr Length	IP address length in bits: 0 or 32 or 128	1
IP Address	Optional IP address	0 or 4 or 16
Label1	Valid downstream assigned label to perform forwarding to CE based on the destination MAC address	3
Label2	Specifies a second label	0-3
EVI RT	Type 0 (2byteAS) route target	8
MAC Mobility	0x0600:{1 byte Sticky bit}:0x00:{4 byte sequence number}	8

- MAC Address field is populated with the CE address.
- IP address field is optional with IP Address length set to 0 bits.

Note IP learning is not supported on Cisco ASR 1000 Series Aggregation Services Routers.

- In the Label field, Per-BD or Per-CE labels can be assigned.

- Per-BD is used when PE advertises a single label for all MAC addresses learned in a given bridge domain.
- Per-CE label assigns a separate label to each access port in the bridge domain.

Route Type 3 - Inclusive Multicast Ethernet Tag Route

Type 3 routes are used for transporting Broadcast, Unknown Unicast and Multicast (BUM) traffic to other PE devices across a given EVPN network instance.

The following is the header format for Type 3 routes:

Table 143: Route Type 3 - Inclusive Multicast Ethernet Tag Route Header

Field	Value	Length (Octets)
Route Type	0x03	1
Length	26 or 38	1
EVI RD	Type 1 (IPv4Addr) RD unique across all EVIs on the PE.	8
Ethernet Tag	0 or valid Ethernet Tag	4
IP Addr Length	IP Address Length - 32 bits or 128 bits	1
IP Address	IP Address common for all EVIs (for example, loopback address)	4 or 16
PMSI Tunnel Attr	{1 byte flags = 0}:{1 byte Tunnel Type}:{3 byte label}:{variable length Tunnel Identifier}	Variable
EVI RT	Type 0 (2byteAS) route target	8

The PE devices advertises an Inclusive Multicast Ethernet Tag (IMET) Route for every EVI-Ethernet Tag sequence. The Ethernet Tag is set to 0 for VLAN-based and VLAN-bundling service interfaces. The Ethernet Tag is set to a valid VLAN ID for VLAN-aware bundling service interface.

Type 3 route also carries a Provider Multicast Service Interface (PMSI) Tunnel attribute as specified in RFC 6514 (BGP Encodings and Procedures for MVPNs).

For Ingress Replication, the IMET route is used to advertise the label (in the PMSI Tunnel Attribute) that the other PEs can use to send BUM traffic to the originating PE device.

Route Type 4 - Ethernet Segment Route

Ethernet segment routes are needed in multihomed scenarios to enable the discovery of PE devices connected to the same Ethernet segment. Ethernet segment routes are also used electing the designated forwarder (DF) for BUM traffic to the CE, on a particular Ethernet segment. Once an ESI has been assigned for the Ethernet segment for a multihomed CE, the ESI is advertised to the ES-Import extended community by the PE as BGP route type 4. The PEs where the import community matches with the ESI import community, imports ES route to auto-discover each other.

The route type value for Ethernet Segment Route is 0x04. It is originated only by PEs connected to multihomed CEs. It is imported only by PEs connected to the same Ethernet Segment. This route has the following format:

Table 144: Route Type 4 - Ethernet Segment Route

Field	Value	Length (Octets)
Route Type	0x04	1
Length	23	1
ES RD	Type 1 (IPv4Addr) RD unique across all Ethernet Segments on the PE.	8
ESI	Ethernet Segment Identifier	10
IP Addr Length	IP Address Length - 32 bits or 128 bits	1
IP Address	IP Address of the originating PE	4 or 16
ES-Import RT	0x0602:{high order 6-octet portion of the 9-octect ESI value}	8

Core Isolation

In scenarios where a PE loses connectivity to the core network, either the core-facing interface on the PE goes to DOWN state, or an upstream event results in BGP peering loss. All the BGP routes types 1, 2, 3, and 4 are withdrawn after the timers expire. All other PEs in the same Ethernet segment are alerted and a new DF is elected by the remaining PEs. However, the access side switch or node is not aware of this event since the multihomed access interface on the PE is still in the UP state. This results in traffic being blackholed, since the access side device continues to forward traffic to the PE.

To remedy this scenario, the core isolation solution is implemented in Cisco IOS-XE software. In the event of BGP peering loss on the PE or the core facing interface goes to DOWN state, the multihomed access interfaces on the PE are placed in err-disabled state. There are no configuration changes made on these access interfaces. Since the access port is in DOWN state, the link partner on the access switch is also in DOWN state and the corresponding port-channel, on the switch, detects that this member interface has gone DOWN. Therefore, the switch stops forwarding traffic on this interface and load balances the traffic amongst the remaining member interfaces. Once the BGP peering is restored the error-disabled states are removed from the multi-homed access interfaces.

Prerequisites for EVPN Multihoming

- EVI and Bridge domains must be in established state with associated MPLS labels.

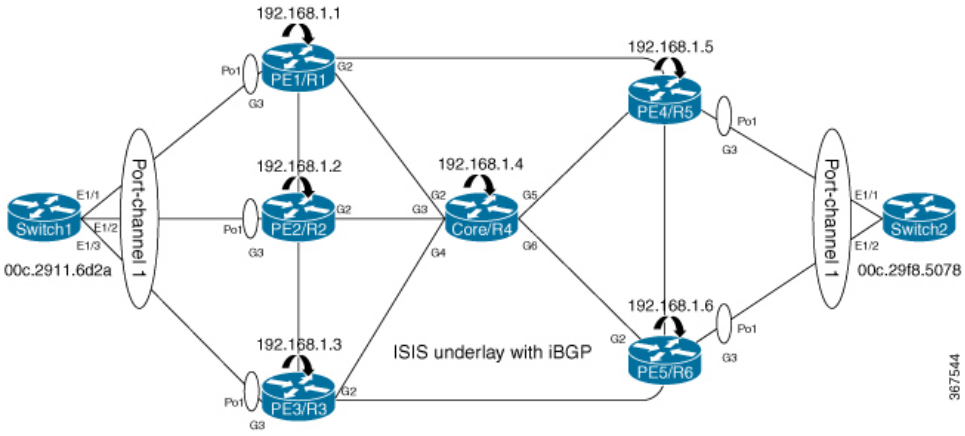
Restrictions for EVPN Multihoming

- The number of bridge domains that are supported are 16000.
- The number of EFPs or service instances that are supported per physical interface are 8000.
- Stateful Switchover is not supported.
- IP learning is not supported on Cisco ASR 1000 Series Aggregation Services Routers.
- Only all-active redundancy mode (2 or 2+ PEs in the same redundancy group sharing the same ESI and all forwarding traffic) is supported.
- Single-active mode is not supported.
- Only access-side flow-based load balancing with multihoming LAG ON mode is supported. Any ether-channel signaling (LACP or PAgP) is not supported.
- MAC mobility and duplication detection is not supported.
- Per-EVI and per-MAC labeling is not supported. Only per-BD and per-CE labeling is supported.
- Only type 3 ESI is supported as defined in section 5 of RFC7432. Type 3 ESI consists of PE System MAC address and local discriminator.
- Port-channel signaling is not supported.
- The port-channel should be configured in ON mode only.

How to Configure EVPN Multioming

Configuring EVPN Multihoming

Figure 100: All-Active Multihoming Topology



The above figure represents L2VPN All-Active Multihoming network. Use the following steps to configure Multihoming:

Configuring L2VPN EVPN Globally and EVI on IOS-XE Router

```
enable
configure terminal
 l2vpn evpn
  replication-type ingress  -> Enables ingress replication label
  router-id Loopback0   -> Configures L2VPN EVPN Router-ID
!
 l2vpn evpn instance 10 vlan-based -> Configures Vlan-based EVI 10
!
 l2vpn evpn instance 20 vlan-bundle -> Configures Vlan-bundled EVI 20
!
 l2vpn evpn instance 30 vlan-aware -> Configures Vlan-aware EVI 30
```

Configuring access interface on PE for EVPN Multi-homing all-active

```
enable
    configure terminal
        interface Port-channel1
           no ip address
           no negotiation auto
           evpn ethernet-segment 1                      -> Configures Ethernet Segment ID
              identifier type 3  system-mac abcd.abcd.abc1    -> Configures system MAC
              redundancy all-active          -> Configures redundancy mode
(all-active/single-active)
            service instance 10 ethernet   -> Enables service instance 10 under the physical
 interface
              encapsulation dot1q 10
              !
            service instance 20 ethernet   -> Enables service instance 20 under the physical
 interface
               encapsulation dot1q 20-21
               !
            service instance 30 ethernet   -> Enables service instance 30 under the physical
 interface
                encapsulation dot1q 30
                !
                !
          interface GigabitEthernet3
             no ip address
             negotiation auto
             isis network point-to-point
             isis three-way-handshake cisco
             channel-group 1
```

Configuring Bridge-domain on IOS-XE Router

```
enable
configure terminal
 bridge-domain 10
  mac aging-time 30    -> Configures aging time for all MACs learnt under bridge-domain
  member Port-channel1 service-instance 10   Links SI 10 on Port-channel1 with Bridge-domain
 10
  member evpn-instance 10   -> Links EVI 10 with Bridge-domain 10
!
bridge-domain 20
 mac aging-time 30
 member Port-channel1 service-instance 20  -> Links SI 20 on Port-channel1 with Bridge-domain
 20
 member evpn-instance 20    -> Links EVI 20 with Bridge-domain 20
!
bridge-domain 30
```

```
 mac aging-time 30
 member Port-channel1 service-instance 30  -> Links SI 30 on Port-channel1 with Bridge-domain
 30
 member evpn-instance 30 ethernet-tag 30  -> Links EVI 30 with Bridge-domain 30
```

Configuring BGP on Provider Edge

```
router bgp 100
 bgp router-id 192.168.1.1
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 192.168.1.4 remote-as 100
 neighbor 192.168.1.4 update-source Loopback0
 !
 address-family ipv4
  neighbor 192.168.1.4 activate
 exit-address-family
 !
 address-family l2vpn evpn    -> Enables L2vpn evpn address family
  neighbor 192.168.1.4 activate
  neighbor 192.168.1.4 send-community both
  neighbor 192.168.1.4 soft-reconfiguration inbound
 exit-address-family
30
```

Configuring BGP on Core Router or Route Reflector

```
router bgp 100
 bgp router-id 192.168.1.4
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 192.168.1.1 remote-as 100
 neighbor 192.168.1.1 update-source Loopback0
 neighbor 192.168.1.2 remote-as 100
 neighbor 192.168.1.2 update-source Loopback0
 neighbor 192.168.1.3 remote-as 100
 neighbor 192.168.1.3 update-source Loopback0
 neighbor 192.168.1.5 remote-as 100
 neighbor 192.168.1.5 update-source Loopback0
 neighbor 192.168.1.6 remote-as 100
 neighbor 192.168.1.6 update-source Loopback0

 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 route-reflector-client
  neighbor 192.168.1.2 activate
  neighbor 192.168.1.2 route-reflector-client
  neighbor 192.168.1.3 activate
  neighbor 192.168.1.3 route-reflector-client
  neighbor 192.168.1.5 activate
  neighbor 192.168.1.5 route-reflector-client
  neighbor 192.168.1.6 activate
  neighbor 192.168.1.6 route-reflector-client
 exit-address-family
 !
 address-family l2vpn evpn     -> Enables L2vpn evpn address family
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 send-community both
  neighbor 192.168.1.1 route-reflector-client
  neighbor 192.168.1.1 soft-reconfiguration inbound
  neighbor 192.168.1.2 activate
  neighbor 192.168.1.2 send-community both
```

```
  neighbor 192.168.1.2 route-reflector-client
  neighbor 192.168.1.2 soft-reconfiguration inbound
  neighbor 192.168.1.3 activate
  neighbor 192.168.1.3 send-community both
  neighbor 192.168.1.3 route-reflector-client
  neighbor 192.168.1.3 soft-reconfiguration inbound
  neighbor 192.168.1.5 activate
  neighbor 192.168.1.5 send-community both
  neighbor 192.168.1.5 route-reflector-client
  neighbor 192.168.1.5 soft-reconfiguration inbound
  neighbor 192.168.1.6 activate
  neighbor 192.168.1.6 send-community both
  neighbor 192.168.1.6 route-reflector-client
  neighbor 192.168.1.6 soft-reconfiguration inbound
exit-address-family
```

Configuration Examples for EVPN Mulltihoming

Verifying EVPN Multihoming

Use the following commands to verify that the bridge domains are in established state and that bridge domain has learnt the local MAC address:

```
PE1# show bridge-domain 10 mac dynamic address
          Port               MAC Address
          Po1 ServInst 10    000c.2911.6d2a -> MAC learnt on port-channel 1 for service
instance 10
```

```
PE1#show bridge-domain 10
Bridge-domain 10 (2 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 30 second(s)    -> MAC aging timer for bridge-domain
    Port-channel1 service instance 10
    EVPN Instance 10
   AED MAC address    Policy  Tag       Age  Pseudoport
   -   000C.29F8.5078 forward static_r  0    OCE_PTR:0xe8e5dda0
   -   000C.2911.6D2A forward dynamic_c 28   Port-channel1.EFP10

PE1#show bridge-domain 10
Bridge-domain 10 (2 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 30 second(s)
    Port-channel1 service instance 10
    EVPN Instance 10
   AED MAC address    Policy  Tag       Age  Pseudoport
   -   000C.29F8.5078 forward static_r  0    OCE_PTR:0xe8e5dda0
   -   000C.2911.6D2A forward static_a  0    Port-channel1.EFP10
```

Note In the above output, MAC addresses with forward dynamic_c tags are locally learned addresses and MAC addresses with forward static_r tags are remote addresses learned through EVPN.

Use the following command to verify the number and type of EVIs configured on the PE, number of bridge-domains configured, and number of MACs learnt locally and remotely:

```
PE1#show l2vpn evpn summary
L2VPN EVPN
  EVPN Instances (excluding point-to-point): 3
    VLAN Aware:   1
    VLAN Based:   1
    VLAN Bundle:  1
  Bridge Domains: 3
  BGP: ASN 100, address-family l2vpn evpn configured
  Router ID: 192.168.1.1
  Label Allocation Mode: Per-BD
  Replication Type: Ingress
  Forwarding State: UP
  MAC Duplication: seconds 180 limit 5
  MAC Addresses: 6
    Local:     3
    Remote:    3
    Duplicate: 0
  IP Duplication: seconds 180 limit 5
  IP Addresses: 0
    Local:     0
    Remote:    0
    Duplicate: 0
  Maximum number of Route Targets per EAD-ES route: 200
```

Note In the above output, the remote MAC addresses' next hops are the addresses of the provider edge devices that these MAC addresses are learned from.

Use the following command to verify ethernet-segments attached to the PE:

```
PE1#show l2vpn evpn ethernet-segment detail
EVPN Ethernet Segment ID: 03AB.CDAB.CDAB.C100.0001
  Interface:              Po1
  Redundancy mode:        all-active
  DF election wait time:  3 seconds
  Split Horizon label:    16
  State:                  Ready
  Ordinal:                0
  RD:                     192.168.1.1:1
    Export-RTs:           100:10 100:20 100:30
  Forwarder List:         192.168.1.1 192.168.1.2 192.168.1.3
```

Use the following command to verify EVPN manager details regarding an EVI:

```
PE1#show l2vpn evpn evi detail
EVPN instance:    10 (VLAN Based)  í VLAN based EVI
  RD:             192.168.1.1:10 (auto)   -> RD derived from Loopback0 EVPN Router-ID:EVI
number
  Import-RTs:     100:10
  Export-RTs:     100:10
  Per-EVI Label:  none
  State:          Established   -> EVI state
  Encapsulation:  mpls
  Bridge Domain:  10
    Ethernet-Tag: 0
    BUM Label:    18
    Per-BD Label: 19
    State:        Established   -> BD state
    Pseudoports:    -> Access interface and DF election status for EVI 10
      Port-channel1 service instance 10 (DF state: PE-to-CE BUM blocked)

EVPN instance:    20 (VLAN Bundle)  -> VLAN bundled EVI
```

```
  RD:             192.168.1.1:20 (auto)
  Import-RTs:     100:20
  Export-RTs:     100:20
  Per-EVI Label:  none
  State:          Established
  Encapsulation:  mpls
  Bridge Domain:  20
    Ethernet-Tag: 0
    BUM Label:    20
    Per-BD Label: 21
    State:        Established
    Pseudoports:
      Port-channel1 service instance 20 (DF state: PE-to-CE BUM blocked)

EVPN instance:    30 (VLAN Aware)  -> VLAN aware EVI
  RD:             192.168.1.1:30 (auto)
  Import-RTs:     100:30
  Export-RTs:     100:30
  Per-EVI Label:  none
  State:          Established
  Encapsulation:  mpls
  Bridge Domain:  30
    Ethernet-Tag: 30
    BUM Label:    22
    Per-BD Label: 23
    State:        Established
    Pseudoports:    -> Elected DF for EVI 30
      Port-channel1 service instance 30 (DF state: forwarding)
```

Note Designated Forwarder (DF) is responsible for forwarding Broadcast, Unicast and Multicast (BUM) traffic on an ethernet segment. Route-type 4 is used to carry this information.

Use the following command to verify EVPN manager details for bridge-domain 10:

```
 PE1#show l2vpn evpn mac bridge-domain 10 detail
MAC Address:               000c.2911.6d2a
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:         03AB.CDAB.CDAB.C100.0001 -> ESI number assigned to the MAC learnt
 on this EFP
Ethernet Tag ID:           0
Next Hop(s):               Port-channel1 service instance 10 -> MAC learnt locally on
port-channel 1
                                   3.3.3.3
Local Address:             0.0.0.0
Label:                     17
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running

MAC Address:               000c.29f8.5078
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:          03AB.CDAB.CDAB.C200.0002
Ethernet Tag ID:           0
Next Hop(s):               6.6.6.6
Local Address:             1.1.1.1
Label:                     19
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running
```

Use the following command to verify EVPN manager details EVI 10:

```
PE1#show l2vpn evpn mac evi 10 detail
MAC Address:               000c.2911.6d2a
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:          03AB.CDAB.CDAB.C100.0001
Ethernet Tag ID:           0
Next Hop(s):               Port-channel1 service instance 10
                                  192.168.1.2
Local Address:             0.0.0.0
Label:                     19
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running

MAC Address:               000c.29f8.5078
EVPN Instance:             10
Bridge Domain:             10
Ethernet Segment:          03AB.CDAB.CDAB.C200.0002
Ethernet Tag ID:           0
Next Hop(s):               192.168.1.5
Local Address:             192.168.1.1
Label:                     23
Sequence Number:           0
MAC only present:          Yes
MAC Duplication Detection: Timer not running
```

Use the following command to verify that the information on BGP routes is sent to Layer 2 RIB :

```
PE1#show l2rib producers
   Producer(ID)     Client ID  Object Type  Admin Dist  Purge Time(sec)    State
   -------------    ---------  -----------  ----------  ---------------  -----------
  L2VPN(  9)          1       Topology          5             120        Converged
    BGP(  5)          0          MAC           20             600        Converged
  L2VPN(  9)          1          MAC           5             1800        Converged
    BGP(  5)          0          EAD           20             600        Converged
  L2VPN(  9)          1          EAD           6              120        Converged
    BGP(  5)          0     IMET_ROUTE         20             600        Converged
  L2VPN(  9)          1     IMET_ROUTE          6             120        Converged
    BGP(  5)          0       MAC-IP           20             600        Converged
  L2VPN(  9)          1       MAC-IP           6             1800        Converged
    BGP(  5)          0     ES_ROUTE           20             600        Converged
  L2VPN(  9)          1       ES_ROUTE          6             1800        Converged
```

Use the following command to verify Route Type 3 IMET tunnels created for each EVI:

```
PE1#show l2route evpn imet
  EVI       ETAG  Prod  Router IP Addr  Type   Label       Tunnel ID
----- ---------- ----- --------------- ----- ------- ---------------
   10          0   BGP     192.168.1.2     6      22     192.168.1.2
   10          0   BGP     192.168.1.3     6      22     192.168.1.3
   10          0   BGP     192.168.1.5     6      22     192.168.1.5
   10          0   BGP     192.168.1.6     6      22     192.168.1.6
   10          0 L2VPN     192.168.1.1     6      18     192.168.1.1
   20          0   BGP     192.168.1.2     6      20     192.168.1.2
   20          0   BGP     192.168.1.3     6      20     192.168.1.3
   20          0   BGP     192.168.1.5     6      20     192.168.1.5
   20          0   BGP     192.168.1.6     6      20     192.168.1.6
   20          0 L2VPN     192.168.1.1     6      20     192.168.1.1
   30         30   BGP     192.168.1.2     6      18     192.168.1.2
   30         30   BGP     192.168.1.3     6      18     192.168.1.3
   30         30   BGP     192.168.1.5     6      18     192.168.1.5
   30         30   BGP     192.168.1.6     6      18     192.168.1.6
   30         30 L2VPN     192.168.1.1     6      22     192.168.1.1
```

Use the following command to verify EAD-EVI route-type 1 for EVI 10 for BGP:

```
PE1# show ip bgp l2vpn evpn evi 10 route-type 1
 BGP routing table entry for [1][192.168.1.1:10][03ABCDABCDABC1000001][0]/23, version 109
Paths: (3 available, best #2, table evi_10)
  Flag: 0x8000
  Advertised to update-groups:
     1
  Refresh Epoch 4
  Local, (received & used), imported path from [1][192.168.1.2:10][03ABCDABCDABC1000001][0]/23
 (global)
    192.168.1.2 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath
      Rcvd Label: 19, Local Label: None
      Extended Community: RT:100:10
      Originator: 192.168.1.2, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (192.168.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, multipath,
best
      Rcvd Label: None, Local Label: 25
      Extended Community: RT:100:10
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 3
  Local, (received & used), imported path from [1][192.168.1.3:10][03ABCDABCDABC1000001][0]/23
 (global)
    192.168.1.3 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath(oldest)
      Rcvd Label: 19, Local Label: None
      Extended Community: RT:100:10
      Originator: 192.168.1.3, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0
BGP routing table entry for [1][192.168.1.1:10][03ABCDABCDABC2000002][0]/23, version 61
Paths: (2 available, best #1, table evi_10)
  Not advertised to any peer
  Refresh Epoch 2
  Local, (received & used), imported path from [1][192.168.1.5:10][03ABCDABCDABC2000002][0]/23
 (global)
    192.168.1.5 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath, best
      Rcvd Label: 19, Local Label: None
      Extended Community: RT:100:10
      Originator: 192.168.1.5, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 2
  Local, (received & used), imported path from [1][192.168.1.6:10][03ABCDABCDABC2000002][0]/23
 (global)
    192.168.1.6 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath(oldest)
      Rcvd Label: 25, Local Label: None
      Extended Community: RT:100:10
      Originator: 192.168.1.6, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0
```

Use the following command to verify EAD-ES route-type 1 output for EVI 10 in BGP database:

```
PE1# show ip bgp l2vpn evpn route-type 1

BGP routing table entry for [1][192.168.1.2:10][03ABCDABCDABC1000001][0]/23, version 2
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 6
  Local, (received & used)
```

```
    192.168.1.2 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Rcvd Label: 23, Local Label: None
      Extended Community: RT:100:10
      Originator: 192.168.1.2, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0x0
```

Use the following command to verify information regarding the PEs with active ESI configuration:

```
PE1#sh ip bgp l2vpn evpn route-type 4
BGP routing table entry for [4][192.168.1.1:1][03ABCDABCDABC1000001][32][192.168.1.1]/23,
version 99
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (192.168.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: EVPN ES-IMPORT:0xABCD:0xABCD:0xABC1
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [4][192.168.1.2:1][03ABCDABCDABC1000001][32][192.168.1.2]/23,
version 102
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 5
  Local, (received & used)
    192.168.1.2 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: EVPN ES-IMPORT:0xABCD:0xABCD:0xABC1
      Originator: 192.168.1.2, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [4][192.168.1.3:1][03ABCDABCDABC1000001][32][192.168.1.3]/23,
version 100
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 5
  Local, (received & used)
    192.168.1.3 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: EVPN ES-IMPORT:0xABCD:0xABCD:0xABC1
      Originator: 192.168.1.3, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [4][192.168.1.5:2][03ABCDABCDABC2000002][32][192.168.1.5]/23,
version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 5
  Local, (received-only)
    192.168.1.5 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: EVPN ES-IMPORT:0xABCD:0xABCD:0xABC2
      Originator: 192.168.1.5, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0
BGP routing table entry for [4][192.168.1.6:2][03ABCDABCDABC2000002][32][192.168.1.6]/23,
version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 5
  Local, (received-only)
    192.168.1.6 (metric 30) (via default) from 192.168.1.4 (192.168.1.4)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: EVPN ES-IMPORT:0xABCD:0xABCD:0xABC2
      Originator: 192.168.1.6, Cluster list: 192.168.1.4
      rx pathid: 0, tx pathid: 0
```

Use the following ether channel state output on tbe CE device:

```
CE1# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        b - BFD Session Wait
        S - Switched    R - Routed
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      NONE      Eth1/1(P)    Eth1/2(P)    Eth1/3(P)
```

Use the following Ether Channel state output on the PE device:

```
PE1#show etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1       Po1(RU)                 Gi3(P)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

Additional References for EVPN Multihoming

Standards and RFCs

Standard	Title
RFC 7432	BGP MPLS-Based Ethernet VPN

Feature Information for EVPN Multihoming

Feature Name	Releases	Feature Information
EVPN Multioming	Cisco IOS XE Fuji 16.9.x	The EVPN Multihoming feature utilizes the BGP MPLS-based Ethernet VPN (EVPN) functionality to achieve Multihoming between a Provider Edge and a Customer Edge device. The following command was introduced or modified: redundancy all-active

CHAPTER 73

EVPN Over MPLS with Integrated Routing and Bridging

EVPN Over MPLS routing and bridging (IRB) allows the device in an EVPN over MPLS network to perform both bridging and routing. IRB allows the devices to forward both Layer 2 or bridged and Layer 3 or routed traffic. A Bridge Domain performs bridging when it forwards traffic to the same subnet. Similarly, a Bridge Domain Interface performs routing when it forwards traffic to a different subnet. The devices in the network forward traffic to each other through the Distributed Anycast Gateways (DAG). The Ethernet VPN over MPLS Integrated IRB Single-Homing (SH) with Distributed Anycast Gateway feature provides support for symmetric IRB model. This feature is supported only on Cisco ASR 1000 Series Aggregation Services Routers.

In symmetric IRB, both the ingress and egress Bridge Domain Interfaces perform both bridging and routing. A packet first moves through a MAC VRF followed by an IP VRF of the ingress device. It then moves through an IP VRF followed by a MAC VRF on the PE of the egress device. The PEs of ingress and egress devicess equally share all the packet processing associated with intersubnet forwarding semantics.

In symmetric IRB, you are required to define only the endpoints on the ingress and egress Bridge Domain interfaces. Symmetric IRB offers better scalability with the BGP EVPN over MPLS fabric.

To support Ethernet VPN over MPLS Integrated IRB Single-Homing (SH) with Distributed Anycast Gateway, you need to configure the following on the Cisco ASR 1000 Series Aggregation Services Router:

- Host IP-MAC learning in single homing setup
- Symmetric IRB for IP-VRF to IP-VRF inter-subnet traffic over MPLS
- Distributed Anycast Gateway with Bridge-Domain
- Host MAC-IP Mobility
- ARP/ND suppression
- Unknown Unicast Suppression

From Cisco IOS XE Cupertino 17.7.1a, Multi-Homing All-Active hosts is supported.

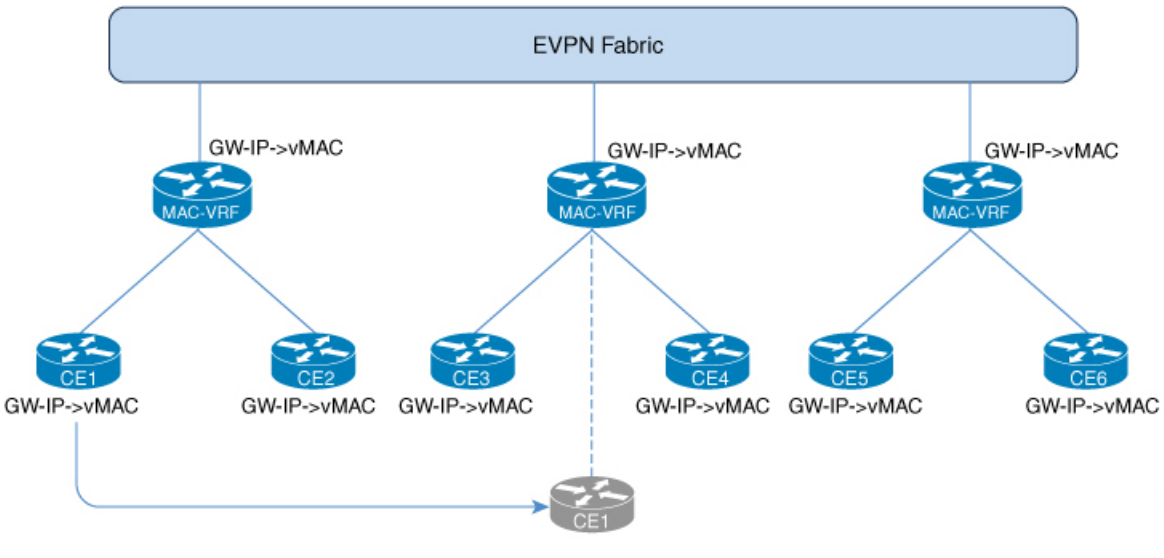
Information about EVPN Over MPLS with Distributed Anycast Gatways

Distributed Anycast Gateway with Bridge Domains

Distributed Anycast Gateway is a default gateway addressing mechanism in a BGP EVPN fabric. The feature enables the use of the same gateway IP and MAC address across all the devices in an EVPN over MPLS network. This ensures that every device functions as the default gateway for the workloads directly connected to it. The feature facilitates flexible workload placement, host mobility, and optimal traffic forwarding across the BGP EVPN fabric.

In this topology, the Distributed Anycast Gateways are directly attached to hosts or network with IP-VRF routing enabled on the IRB (BDI) interfaces on the gateways. To reduce the complexity, only virtual MAC DAGs is supported and the duplication address detection (DAD) for IPv6 on the BDI interfaces on Distributed Anycast Gateways is disabled.

Figure 101: Distributed Anycast Gateway with Bridge Domains



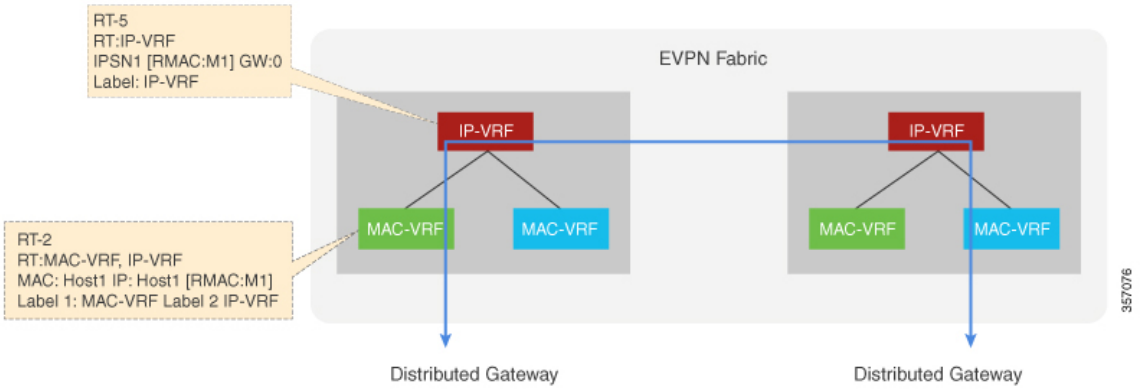
On the DAG, the bridge domain check if an Address Resolution Protocol or Neighbor Discovery Protocol packet from a local host is sent to the BDI (Gateway) IP addresses. If the packet is sent to BDI (Gateway) IP addresses, this packet is handled by local BDI and it is not flooded into the bridge domain and sent across the EVPN fabric.

Symmetric IRB with MPLS on Distributed Gateways

Symmetric IRB is a distributed routing model which utilizes direct IP-VRF to IP-VRF connectivity for inter-subnet traffic. To support Symmetric IRB, the native IRB needs to be enabled on Distributed Gateways by creating the BDIs, configuring virtual MAC, IP-VRF, and anycast IP address.

After the native IRB is enabled, BGP allocates the L3 label for the RT-2's and RT-5's per VRF basis and advertises it.

Figure 102: Symmetric IRB with MPLS on Distributed Gateways



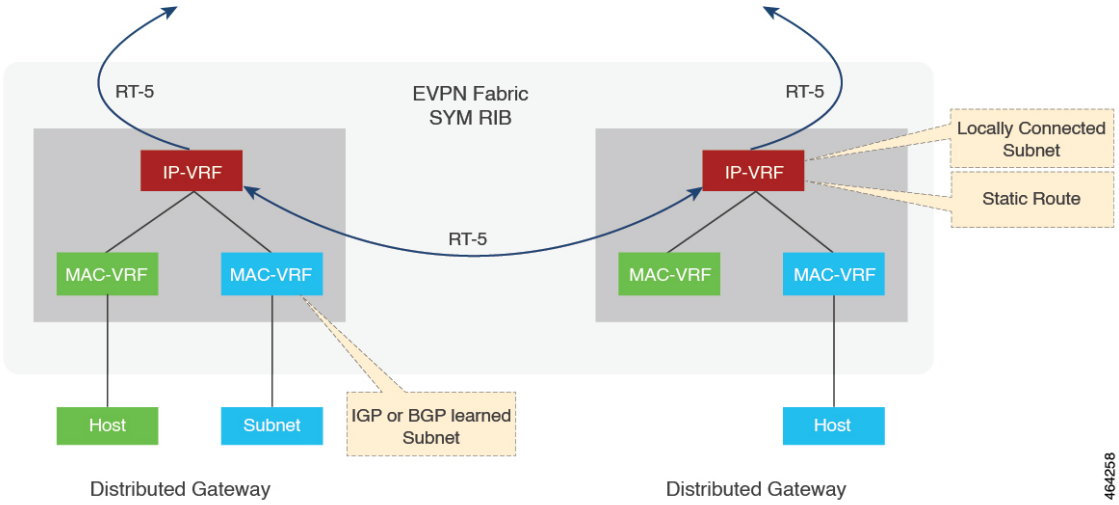
IP Prefix Route on Distributed Gateways

On Distributed Gateways, if the IP-VRF Stitching is configured and IP-VRF to EVPN redistribution is enabled, the IP Prefixes in IP-VRF are advertised as the RT-5 routes. These routes can be a locally connected subnet, a static route, or a route IP Prefix on the CE side.

On the receiving Distributed Gateways, RT-5 is imported into the corresponding IP-VRF and installed as a remote IP route.

In the case of a Distributed Anycast Gateway, many Distributed Gateways could advertise the same subnet prefix route. If the receiving Distributed Gateway has the same local subnet prefix, the local subnet prefix takes precedence. If a remote Distributed Gateway doesn't have the same local subnet prefix, remote routes are chosen or ECMP load balancing is used for forwarding. Host discovery and MAC-IP learning procedures are triggered when traffic is sent to the host before it's learned on any DAGs.

Figure 103: IP Prefix Route on Distributed Gateways



IP Prefix Route on Border Gateways

On Border Gateways that handle the Layer 3 hand-off to Multi-VRF, the IP-VRF Stitching needs to be configured and IP-VRF to EVPN redistribution needs to be enabled. The IP Prefix that is locally connected, a static route, or a learned route from the other side in IP-VRF is advertised as the RT-5 routes to EVPN Fabric. The IP Prefixes (RT-5) and Host Routes (RT-2) imported from EVPN are installed into IP-VRF and redistributed.

On Border Gateways that handle the Layer 3 hand-off to MPLS VPN (L3VPN), the IP-VRF Stitching needs to be configured. The cross-importing between VPN and EVPN Address-Family needs to be enabled as well. The IP Prefixes imported from VPN AF are advertised as RT-5 routes to EVPN Fabric. The IP Prefixes (RT-5) and Host Routes (RT-2) imported from EVPN AF are re-originated and advertised to the VPN side.

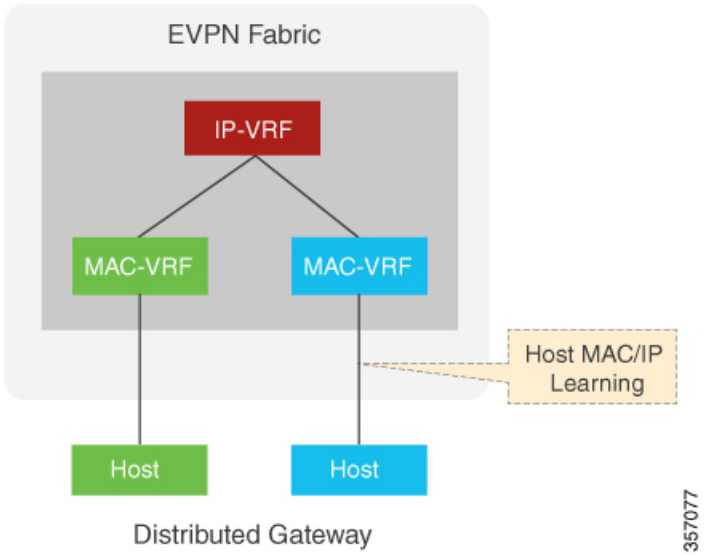
The host routes might be filtered if it's not desired to advertise or redistribute. The IP Prefix route can be used to inject the default route.

Host MAC-IP Binding on a Single-Homed DAG

The Host MAC-IP binding is learned by snooping Address Resolution Protocol (ARP), Neighbor Discovery Protocol, or DHCP packets. After the MAC-IP binding is learned, an age timer (AGE_TIME) is applied to the locally learned binding entry. The binding entry is refreshed whenever the host initiates ARP or ND procedures.

When the age timer expires, the MAC-IP binding is deleted and withdrawn from the network. To avoid premature deletion of the MAC-IP bindings, the gateways initiates the ARP or ND procedure to refresh the binding entries. It also initiates the ARP or ND probe procedure with the data plane *MAC_AGE_OUT* event. The local *MAC_AGE_OUT* event triggers the probe of all the MAC-IP bindings derived from that specific MAC to refresh the binding entries.

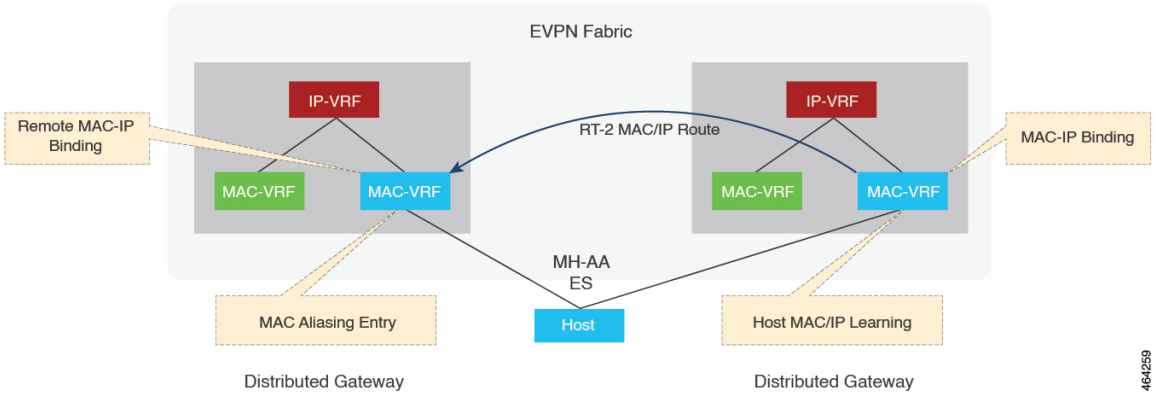
Figure 104: Host MAC-IP Binding on a Single-Homed DAG



Host MAC-IP Binding on Multi-Homing All-Active DAGs

For the hosts on a Multi-Homing All-Active Ethernet Segment, the Host MAC-IP Binding might be initially learned on only one of the multihoming peers. Other peers in the Multi-Homing Group rely on the received remote RT-2 to get the MAC-IP Binding or might locally learn the MAC-IP Bindings.

Figure 105: MAC-IP Binding on MH-AA Distributed Gateways



After the MAC-IP Binding is learned locally on any peer, an age timer (AGE_TIME) is applied to the binding entry, and the binding entry is refreshed whenever the host initiates ARP/ND to this peer. When the age timer expires, this locally learned MAC-IP Binding is deleted, and RT-2 is withdrawn or updated if RT-2 Proxy Route is enabled. When the age timer expires, the MAC-IP binding is deleted and withdrawn from the network.

To avoid premature deletion of the MAC-IP bindings, the gateways initiate the ARP or ND procedure to refresh the binding entries. This is triggered by a refresh timer (SEND_REFRESH_TIME). It also initiates the ARP or ND probe procedure with the data plane MAC_AGE_OUT event. The data plane usually has a

shorter age timer. The local MAC_AGE_OUT event triggers the probe of all the MAC-IP bindings derived from that specific MAC to refresh the binding entries across the Multi-Homing Group Peers.

During fast convergence and slow convergence, gateways can initiate the ARP/ND Refresh procedure using local Bindings or the previously received remote MAC-IP Bindings.

Host MAC-IP Mobility

The host MAC-IP mobility helps to handle the following events:

- Host Move Learn from Data Packet and GARP
- Host Move Detection for Silent Host

Also, the host MAC-IP mobility supports the following scenarios:

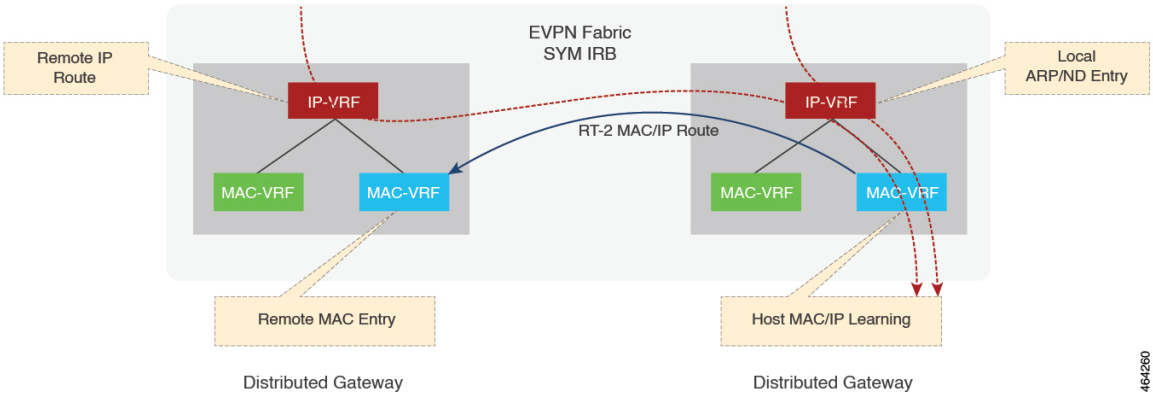
- Moving MAC from local to local
- Moving MAC from local to remote
- Moving MAC from remote to local
- Moving IP local to local
- Moving IP from local to remote
- Moving IP from remote to local

From Cisco IOS XE Cupertino 17.7.1a, the host MAC-IP mobility is supported for hosts on a Multi-Homing All-Active Ethernet Segment.

Host MAC-IP Synchronization

When the MAC-IP Route is imported on remote Distributed Gateways, it can be used for inter-subnet routing. Because Symmetric IRB uses the IP-VRF to IP-VRF Layer 3 Label or VNI for the routing, it doesn't need ARP/ND entries on remote Distributed Gateways, and a remote IP Route with Layer 3 Label (VNI) is installed.

Figure 106: MAC-IP Sync on Remote DAGs (SYM IRB)

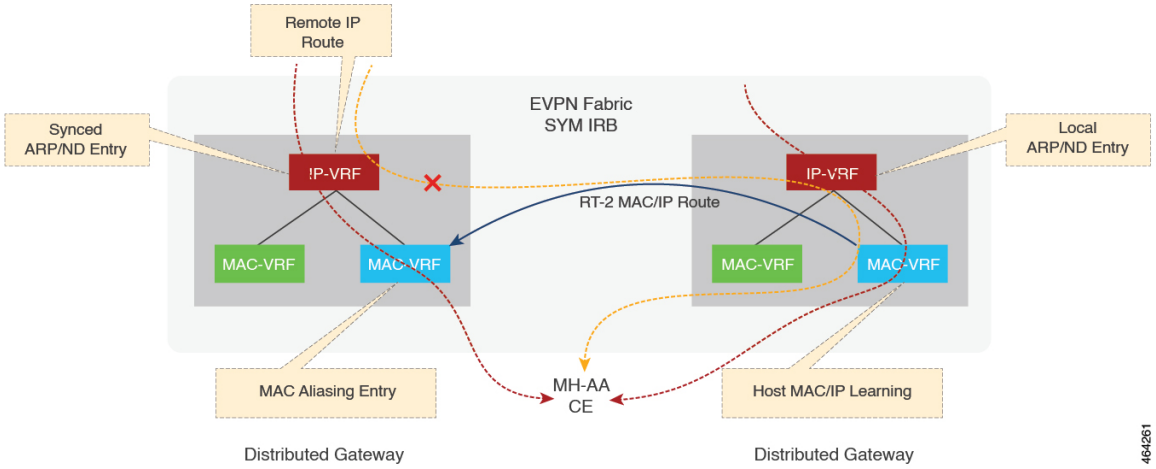


Considering the Multihoming Distributed Gateways, the remote Distributed Gateways could receive multiple MAC-IP routes from that multihoming group members with the same MAC and IP Binding for Layer 3 Load

Balancing. The fast convergence on the multihomed Gateways changes the Layer 2 Bridging in the MAC-VRF on the remote Gateways as it does, but it won't affect the remote IP Route entries and Layer 3 Load Balancing which are sourced from MAC-IP Routes. The withdrawal of the contributing remote MAC-IP route changes Layer 3 Load Balancing on the Remote Distributed Gateways. The withdrawal of all the contributing remote MAC-IP routes triggers the deletion of the remote MAC-IP Route on the Remote Distributed Gateway.

On a Multihoming All-Active Ethernet Segment, the Host MAC-IP Binding might be learned on only one of the multihoming peers. To enable the inter-subnet traffic forwarding to the Ethernet Segment LOCALLY on other multihoming peers, ARP/ND entries need to be synced to those peers via RT-2 MAC-IP Route.

Figure 107: MAC-IP Sync on MH All-Active DAGs (SYM IRB)



When a remote RT-2 MAC-IP Route for an MH-AA host is received from a multihoming peer, the RT-2 import on BGP installs a remote IP Route with the L3 Label (VNI). The ARP or ND entry needs to be installed or synced as well. From the forwarding point of view, both RIB (BGP and other routing components) and Adjacency (ARP or others) contribute to forwarding.

When BGP receives a Local or Aliasing MAC-IP Route from L2RIB, it is installed as static into RIB directly with a lower distance, so that RIB would prefer the local route, that is, the best path. During convergence or recovery, if the local or aliasing route is removed, RIB runs best path again.

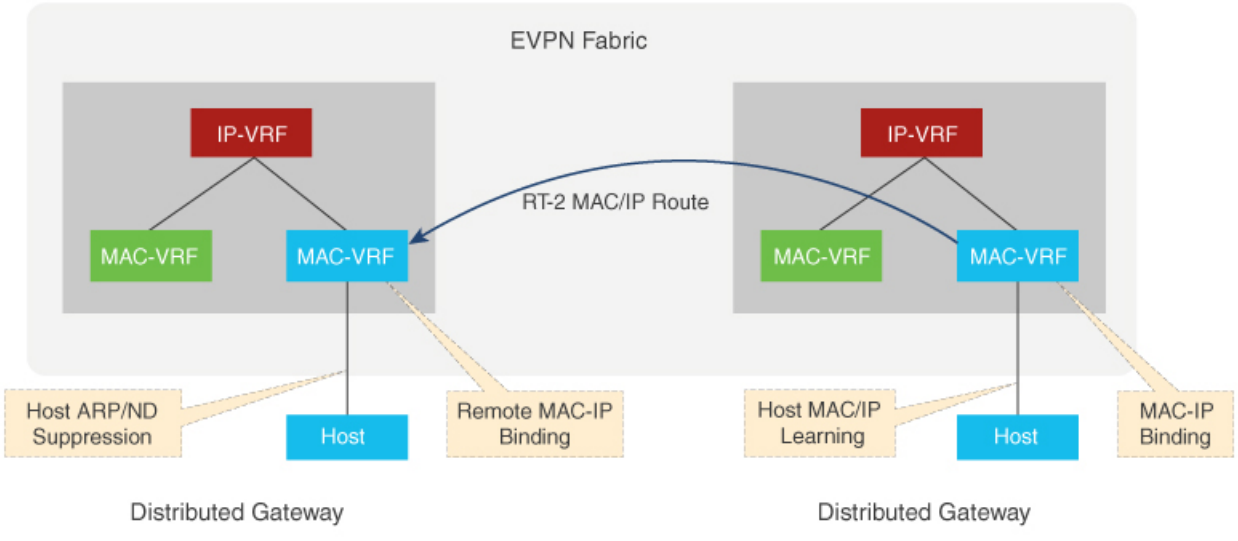
ARP and ND Flooding Suppression

The ARP and ND Flooding Suppression feature depends on device-tracking enabled on the same VLAN or interface. The Switch Integrated Security Features based (SIS-based) device tracking helps to track the presence,location, and movement of end-nodes in the network. SISF snoops traffic received by the device, extracts the device identity (MAC and IP address), and stores them in a binding table. SIS-based device tracking supports both IPv4 and IPv6.

When you enable IPv4 or IPv6 flooding suppression, it helps to minimize the flooding of a broadcast or multicast packet over the EVPN fabric and to remote CEs such as host, router, and switch. For example, Address Resolution packets such as ARP (broadcast) and NS (multicast). The multicast and broadcast suppression capabilities helps to preserve bandwidth in wireless networks.

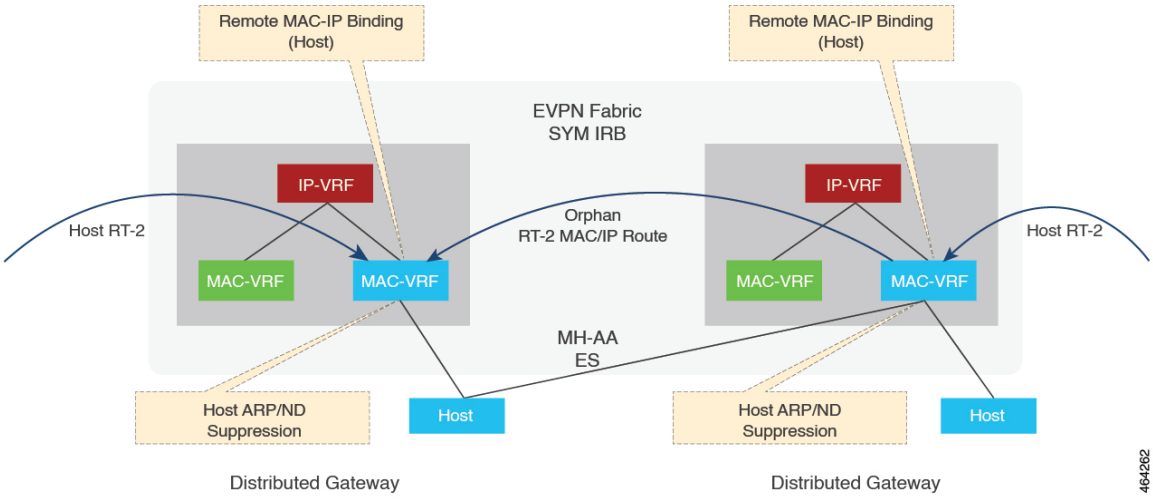
This feature helps to suppress the broadcast (ARP) or link-local multicast (NDP) messages circulating in the layer 2 domain and the packets are relayed after converting their L2 addresses to unicast. By default, this feature is enable and you can use the **disable flooding suppression** command to disable flooding suppression.

Figure 108: ARP and ND Flooding Suppression



From Cisco IOS XE Cupertino 17.7.1a, ARP and ND flooding suppression is supported on Multi-Homing All-Active (MH-AA) hosts. This feature reduces the ARP/ND Flooding traffic across a fabric since the gateway attached to the host might have already learned the MAC-IP Bindings from the control plane. By default, this feature is enabled. For more information, see *Disabling ARP or ND Suppression*.

Figure 109: ARP and ND Suppression on MH-AA Distributed Gateways

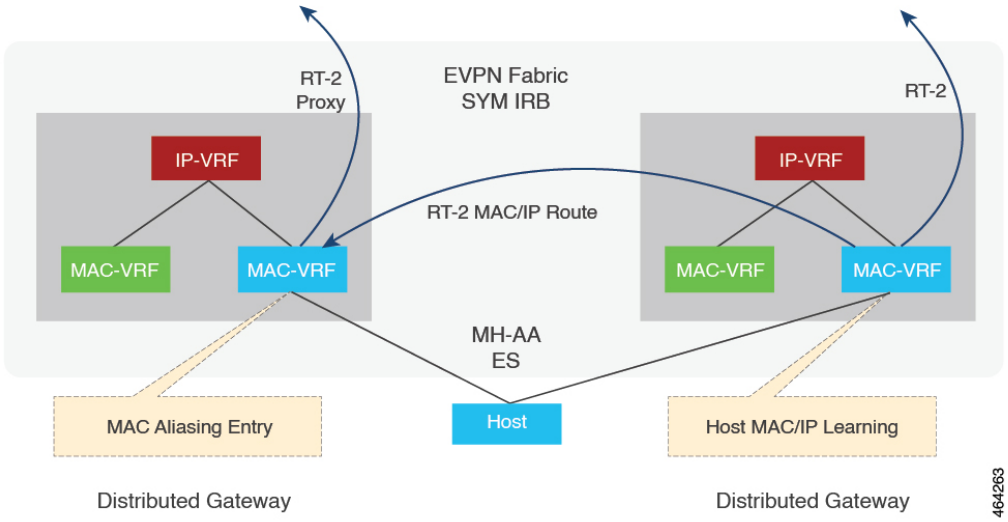


MAC-IP Proxy Route for Multi-Homing All-Active Hosts with Symmetric IRB

Due to load balancing between hosts and the gateways in a Multi-Homing Group, the host MAC-IP binding might be learned and advertised from only one or more gateways. For routing traffic from a remote gateway,

the host is only reachable through the advertising gateways. For Layer 2 Bridging, an aliasing path using the EAD Per EVI route is implemented to achieve Layer 2 load balancing. However, the aliasing path is only valid in a Layer 2 topology (MAC-VRF), and cannot be used in a Layer 3 topology (IP-VRF). To achieve Layer 3 ECMP for a Multi-Homing All-Active Host, a peer in the Multi-Homing Group can choose to be a proxy or readvertise a remote MAC-IP Route that it received for the All-Active host. This feature is enabled by default, and it can be implemented by a peer if it has not locally learned the MAC-IP route.

Figure 110: MAC-IP Proxy on MH All-Active DAGs



Prerequisites for EVPN Over MPLS

This feature requires the following configuration on Cisco ASR 1000 Series Aggregation Services Routers:

- Host MAC-IP learning
- Symmetric IRB for IP-VRF to IP-VRF inter-subnet traffic over MPLS
- Distributed anycast gateway with bridge-domain
- Host MAC-IP mobility
- ARP/ND flooding suppression
- Unknown unicast suppression

Restrictions EVPN over MPLS

- Only Dual-Homing (two peers) Active-Active is supported.
- Only Symmetric IRB is supported. Asymmetric IRB and centralized IRB are not supported for BGP EVPN over MPLS.
- Only Gateway virtual MAC address is supported.

- Global VRF is not supported for MPLS IRB.
- VPLS Stitching is not covered and verified for BGP EVPN over MPLS.
- RT-5-only based routing is not applicable to Multi-Homing All-Active Subnets on DAGs if RT-2 is disabled. This routing is not applicable because Multi-Homing peers need RT-2 MAC-IP route for ARP/ND SYNC.
- SISF security feature is not supported on Multi-Homing All-Active DAGs.

How to Configure EVPN over MPLS

Configure Basic EVPN over MPLS

Use the following steps to configure the basic EVPN over MPLS:

Configuring Layer 2 Virtual Private Network EVPN

```
interface Loopback2
 description L2VPN EVPN ROUTER-ID
 ip address 1.1.1.3 255.255.255.255
 ip ospf 1 area 0
 ipv6 address ABCD:1::3/128
end

l2vpn evpn
!! Only Ingress Replication is supported for EVPN MPLS
 replication-type ingress
 router-id Loopback2
end
!
```

Configuring Layer 2 Virtual Private Network EVPN Instance

```
!! EVPN MPLS supports vlan-based, vlan-aware, vlan-bundle types for L2
!! vlan-bundle type won't be supported for IRB
!! default encapsulation is MPLS
l2vpn evpn instance 1 vlan-based
```

Configuring Interface, Bridge-Domain and EFP

```
interface Ethernet0/1
 no ip address
 service instance 11 ethernet
  encapsulation dot1q 11
end

bridge-domain 11
 !! link point of BD, EFP and EVI
 member Ethernet0/1 service-instance 11
 member evpn-instance 1
end
```

Configuring Layer 2 Virtual Private Network EVPN BGP

```
interface Loopback0
 description BGP UPDATE SOURCE
 ip address 1.1.1.1 255.255.255.255
 ip ospf 1 area 0
 ipv6 address ABCD:1::1/128
end

router bgp 100
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 99.99.99.99 remote-as 100
 neighbor 99.99.99.99 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 99.99.99.99 activate
  neighbor 99.99.99.99 send-community both
 exit-address-family
end
```

Configure Basic EVPN over MPLS with IRB

Use the following steps to configure the basic EVPN over MPLS with IRB.

Configuring IP-VRF

```
vrf definition red
 rd 100:1
 !
 address-family ipv4
  route-target export 100:100
  route-target import 100:100
  route-target export 100:100 stitching
  route-target import 100:100 stitching
 exit-address-family
 !
 address-family ipv6
  route-target export 100:200
  route-target import 100:200
  route-target export 100:200 stitching
  route-target import 100:200 stitching
 exit-address-family
end
```

Configuring Bridge-Domain IRB Interface

```
interface BDI11
 !! virtual MAC for Distributed Anycast Gateway
 mac-address 0011.0011.0011
 vrf forwarding red
 ip address 192.168.11.254 255.255.255.0
 ipv6 address 2001:11::254/64
 encapsulation dot1Q 11
end
```

Configuring BGP IRB

```
router bgp 100
 bgp log-neighbor-changes
 bgp graceful-restart
```

```
 neighbor 99.99.99.99 remote-as 100
 neighbor 99.99.99.99 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 99.99.99.99 activate
  neighbor 99.99.99.99 send-community both
 exit-address-family
!! IP Prefix Advertisement
address-family ipv4 vrf red
  advertise l2vpn evpn
  redistribute connected
exit-address-family
address-family ipv6 vrf red
  advertise l2vpn evpn
  redistribute connected
exit-address-family
End
```

EVPN over MPLS with Multi-VRF Hand-off

Use the following steps to configure EVPN-MPLS Multi-VRF:

Configuring BGP Multi-VRF on Border Gateways

```
router bgp 100
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 99.99.99.99 remote-as 100
 neighbor 99.99.99.99 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 99.99.99.99 activate
  neighbor 99.99.99.99 send-community both
 exit-address-family
address-family ipv4 vrf red
  advertise l2vpn evpn
  redistribute connected
  neighbor 192.168.0.1 remote-as 10
  neighbor 192.168.0.1 activate
  !! If using MPLS between PE and CE
  neighbor 192.168.0.1 send-label
 exit-address-family
 !
 address-family ipv6 vrf red
  advertise l2vpn evpn
  redistribute connected
  neighbor 2001:0::1 remote-as 10
  neighbor 2001:0::1 activate
  !! If using MPLS between PE and CE
  !! neighbor 2001:0::1 send-label <= MPLS is only supported on IPv4 Core (6PE)
 exit-address-family
```

EVPN over MPLS L3VPN Hand-offff

Use the following steps to configure the VPN over MPLS Layer-3 VPN.

Configuring BGP VPNv4/VPNv6 on Border Gateways

```
router bgp 100
 bgp log-neighbor-changes
```

```
 bgp graceful-restart
 !! eBGP peering
 neighbor 10.5.0.1 remote-as 10
 neighbor 99.99.99.99 remote-as 100
 neighbor 99.99.99.99 update-source Loopback0
 !
 address-family vpnv4
 !! EVPN to VPNv4
  import l2vpn evpn re-originate
  neighbor 10.5.0.1 activate
  neighbor 10.5.0.1 send-community both
 exit-address-family
 !
 address-family vpnv6
 !! EVPN to VPNv6
  import l2vpn evpn re-originate
  neighbor 10.5.0.1 activate
  neighbor 10.5.0.1 send-community both
 exit-address-family
 !
 address-family l2vpn evpn
 !! VNPv4/VPNv6 to EVPN
  import vpnv4 unicast re-originate
  import vpnv6 unicast re-originate
  neighbor 99.99.99.99 activate
  neighbor 99.99.99.99 send-community both
  neighbor 99.99.99.99 next-hop-self
 exit-address-family
```

Layer 2 Multihoming Configuration for EVPN over MPLS

Use the following steps to configure Layer 2 Multihoming for EVPN over MPLS:

Configuring Ethernet Segment

```
l2vpn evpn ethernet-segment 1
 !! Support both type 3 and type 0
 identifier type 3 system-mac aabb.0000.0001
 !! Only all-active is support for EVPN MPLS
 redundancy all-active
 df-election wait-time 3
end
```

Configuring L2VPN EVPN Instance

```
!! EVPN MPLS supports vlan-based, vlan-aware, vlan-bundle types for L2
!! vlan-bundle type won’t be supported for IRB
!! default encapsulation is MPLS
l2vpn evpn instance 1 vlan-based
```

Configuring Interface, Bridge Domain, and EFP

```
bridge-domain 11
!! link point of BD, EFP, and EVI
member Port-channel1 service-instance 11
member evpn-instance 1

interface Port-channel1
no ip address
no negotiation auto
no mop enabled
```

```
no mop sysid
!! link point of interface and ESI
evpn ethernet-segment 1
lacp device-id 0005.0005.0005
service instance 11 ethernet
encapsulation dot1q 11
!
!

!! interface linking PE to MH CE
interface GigabitEthernet0/0/1
no ip address
negotiation auto
lacp rate fast
!! link to Port-channel
channel-group 1 mode active
```

Verification Examples for EVPN over MPLS

Show Device-tracking Policy

Use the following command to verify that all the SISF feature policies are attached to bridge domain:

```
show device-tracking policies
Target               Type  Policy               Feature         Target range
bd 11                 bd  evpn-device-track Device-tracking      bd     all
bd 11                 bd  evpn-flood-suppress  Flooding Suppress bd     all

show device-tracking policy evpn-device-track
Policy evpn-device-track configuration:
  security-level glean
  device-role node
  gleaning from Neighbor Discovery
  gleaning from DHCP
  gleaning from ARP
  gleaning from DHCP4
  NOT gleaning from protocol unkn
Policy evpn-device-track is applied on the following targets:
Target               Type  Policy               Feature        Target range
bd 11                  bd  evpn-device-track Device-tracking bd    all
```

Show MAC and IP Binding Tables on Local PE

Use the following command to verify that the MAC and IP Binding tables are on local PE:

```
show device-tracking database

    Network Layer Address      Link Layer Address Interface   vlan prlvl  age   state
Time left

ARP 192.168.255.3              0050.56b0.9d09  Te1/0/5         11  0005   34s  REACHABLE
90 s
ARP 192.168.1.8                0050.56b0.c8f5  Te1/0/6         11  0005   14s  REACHABLE
107 s try 0
ND  FE80::250:56FF:FEB0:C8F5   0050.56b0.c8f5  Te1/0/6         11  0005    7s  REACHABLE
116 s
ND  FE80::250:56FF:FEB0:9D09   0050.56b0.9d09  Te1/0/5         11  0005   28s  REACHABLE
95 s
ND  2001:192:168:255::3        0050.56b0.9d09  Te1/0/5         11  0005   32s  REACHABLE
```

```
90 s
ND  2001:192:168:1::8          0050.56b0.c8f5  Te1/0/6         11  0005   12s  REACHABLE
111 s
```

```
show l2vpn evpn mac ip
IP Address                EVI   BD.   MAC Address    Next Hop(s)
------------------------- ----- ----- -------------- -------------------------
192.168.1.8               1     11    0050.56b0.c8f5 Te1/0/6:11
192.168.255.3             1     11    0050.56b0.9d09 Te1/0/5:11
2001:192:168:1::8         1     11    0050.56b0.c8f5 Te1/0/6:11
2001:192:168:255::3       1     11    0050.56b0.9d09 Te1/0/5:11
FE80::250:56FF:FEB0:9D09  1     11    0050.56b0.9d09 Te1/0/5:11
```

MAC and IP Binding Entries on Remote PE

Use the following command to verify that the MAC and IP Binding tables are on remote PE:

```
show l2vpn evpn mac ip

IP Address                EVI   VLAN  MAC Address    Next Hop(s)
------------------------- ----- ----- -------------- -------------------------
192.168.1.8               1     11    0050.56b0.c8f5 1.1.1.101
192.168.255.3             1     11    0050.56b0.9d09 1.1.1.101
2001:192:168:1::8         1     11    0050.56b0.c8f5 1.1.1.101
2001:192:168:255::3       1     11    0050.56b0.9d09 1.1.1.101
FE80::250:56FF:FEB0:9D09  1     11    0050.56b0.9d09 1.1.1.101
FE80::250:56FF:FEB0:C8F5  1     11    0050.56b0.c8f5 1.1.1.101
```

Displays the device-tracking database on Cisco ASR 1000 Series Aggregation Services Routers.

```
show device-tracking database

 Network Layer Address              Link Layer Address Interface        bd   prlvl  age
 state     Time left
L   192.168.1.100                        aabb.cc00.01ff  BD11             11  0100 2628mn
 REACHABLE
L   FE80::A8BB:CCFF:FE00:1FF               aabb.cc00.01ff  BD11             11  0100 2628mn
 REACHABLE
L   2001:192:168:1::100                    aabb.cc00.01ff  BD11             11  0100 2628mn
 REACHABLE
```

Show MAC and IP Binding Tables on Local PE in a Multi-Homing Setup

Use the following commands to verify the MAC and IP binding tables are on a local PE in a Multi-Homing setup.

```
#show device-tracking database
Binding Table has 6 entries, 3 dynamic (limit 1000000)
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


    Network Layer Address                  Link Layer Address     Interface  bd
prlvl      age        state      Time left
```

```
L   192.168.12.254                          0012.0012.0012         BD12       12
0100       16mn       REACHABLE
ARP 192.168.12.3                            aabb.aabb.0012         Po1        12
0005       8mn        STALE     try 0 1662 s
ND  FE80::A8BB:AAFF:FEBB:12                 aabb.aabb.0012         Po1        12
0005       3mn        REACHABLE  120 s
L   FE80::212:FF:FE12:12                    0012.0012.0012         BD12       12
0100       16mn       REACHABLE
L   2001:12::254                            0012.0012.0012         BD12       12
0100       16mn       REACHABLE
ND  2001:12::3                              aabb.aabb.0012         Po1        12
0005       8mn        STALE     try 0 1618 s
```

The PE that locally learned the MH MAC/IPs (192.168.12.3, 2001:12::3) has a MAC or IP binding in SISF.

```
#show l2vpn evpn mac ip
IP Address                            EVI   BD    MAC Address    Next Hop(s)
------------------------------------- ----- ----- -------------- -----------
192.168.12.3                          2     12    aabb.aabb.0012 Po1:12
                                                                   3.3.3.1
2001:12::3                            2     12    aabb.aabb.0012 Po1:12
                                                                   3.3.3.1
FE80::A8BB:AAFF:FEBB:12               2     12    aabb.aabb.0012 Po1:12
                                                                   3.3.3.1
```

The **Next Hops** column shows a local interface and a next hop to the other MH PE.

```
#show l2vpn evpn mac ip summary
EVI   BD    Ether Tag  Remote IP  Local IP   Dup IP
----- ----- ---------- ---------- ---------- ----------
2     12    0          0          3          0

Total                  0          3          0
```

The MH MAC or IPs are Local in the PE that locally learned these MAC or IPs.

Show MAC and IP Binding Tables on Local Proxy PE in a Multi-Homing Setup

Use the following commands to verify the MAC and IP binding tables are on a local proxy PE in a Multi-Homing setup.

```
#show device-tracking database
Binding Table has 3 entries, 0 dynamic (limit 1000000)
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


    Network Layer Address                   Link Layer Address     Interface  bd
prlvl      age        state      Time left
L   192.168.12.254                          0012.0012.0012         BD12       12
0100       254mn      REACHABLE
L   FE80::212:FF:FE12:12                    0012.0012.0012         BD12       12
0100       254mn      REACHABLE
L   2001:12::254                            0012.0012.0012         BD12       12
0100       254mn      REACHABLE
```

The PE that is a proxy for the MH MAC or IPs does not have a MAC or IP binding in SISF for the MH MAC or IPs.

```
#show l2vpn evpn mac ip
IP Address                            EVI   BD    MAC Address    Next Hop(s)
------------------------------------- ----- ----- -------------- -----------
192.168.12.3                          2     12    aabb.aabb.0012 Po1:12
                                                                 4.4.4.1
2001:12::3                            2     12    aabb.aabb.0012 Po1:12
                                                                 4.4.4.1
FE80::A8BB:AAFF:FEBB:12               2     12    aabb.aabb.0012 Po1:12
                                                                 4.4.4.1
```

The **Next Hops** column shows a local interface and a next hop to the other MH PE.

```
#show l2vpn evpn mac ip summary
EVI   BD    Ether Tag  Remote IP  Local IP   Dup IP
----- ----- ---------- ---------- ---------- ----------
2     12    0          0          3          0

Total                  0          3          0
```

The MH MAC/IPs are Local in the PE that is proxy for these MAC or IPs.

Show MAC and IP Binding Tables on Remote PE in a Multi-Homing Setup

Use the following commands to verify the MAC and IP binding tables are on a remote PE in a Multi-Homing setup.

```
#show device-tracking database
Binding Table has 3 entries, 0 dynamic (limit 1000000)
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


    Network Layer Address                    Link Layer Address     Interface  bd
prlvl      age        state      Time left
L   192.168.12.254                           0012.0012.0012         BD12       12
0100       127s       DOWN
L   FE80::212:FF:FE12:12                     0012.0012.0012         BD12       12
0100       127s       DOWN
L   2001:12::254                             0012.0012.0012         BD12       12
0100       127s       DOWN
```

The remote PE does not have a MAC/IP binding in SISF for the MH MAC/IPs.

```
#show l2vpn evpn mac ip
IP Address                            EVI   BD    MAC Address    Next Hop(s)
------------------------------------- ----- ----- -------------- -----------
192.168.12.3                          2     12    aabb.aabb.0012 3.3.3.1
                                                                 4.4.4.1
2001:12::3                            2     12    aabb.aabb.0012 3.3.3.1
                                                                 4.4.4.1
FE80::A8BB:AAFF:FEBB:12               2     12    aabb.aabb.0012 3.3.3.1
                                                                 4.4.4.1
```

The **Next Hops** column shows remote next hops to the MH PEs for the MH MAC/IPs.

```
#show l2vpn evpn mac ip summary
EVI   BD    Ether Tag  Remote IP  Local IP   Dup IP
```

```
----- ----- ---------- ---------- ---------- ----------
2     12    0          3          0          0

Total                  3          0          0
```

The MH MAC/IPs are Remote in the remote PE.

Device Tracking Counters

Use the following command to verify the device-tracking counters:

```
show device-tracking counters bd 11
Received messages on bd 11   :
Protocol        Protocol message
NDP             RS[4] RA[4] NS[1777] NA[2685]
DHCPv6
ARP             REQ[12] REP[1012]
DHCPv4
ACD&DAD         --[8]

Received multicast messages on bd 11   :
Protocol        Protocol message
NDP             RS[4] NS[8] NA[8]
DHCPv6
ARP             REQ[6] REP[4]
DHCPv4

Bridged messages from bd 11   :
Protocol        Protocol message
NDP             RS[4] RA[4] NS[2685] NA[1778]
DHCPv6
ARP             REQ[1029] REP[4]
Protocol        Protocol message
NDP
DHCPv6
ARP
DHCPv4
ACD&DAD

Probe message on bd 11   :
Type            Protocol message
PROBE_SEND      NS[908] REQ[1023]
PROBE_REPLY     NA[907] REP[995]

Dropped messages on bd 11   :
Feature             Protocol Msg [Total dropped]
Device-tracking:    NDP      NA  [907]
                    reason:  Silent drop [907]

                    ARP      REP [995]
                    reason:  Silent drop [995]
```

QFP BD SISF Statistics and Snoop Protocols

Use the following command to verify the QFP BD SISF statistics and snoop protocols:

```
show platform hard qfp ac feat bridge data 10
infra-1001x-2#show pla hard qfp ac fea brid da 10
QFP L2BD Bridge Domain information

BD id                  : 10
```

```
State enabled          : Yes
Aging timeout (sec)    : 300
……
Unknown unicast olist  : Yes
otv_aed_enabled : No
otv_enabled : No
mcast_snooping_enabled : No
Feature : evpn
SISF snoop protocols   : arp, ndp, dhcpv4, dhcpv6
Mac learned            : 1
……

Bridge Domain statistics

Total bridged               pkts : 577        bytes: 48602
Total unknown unicast       pkts : 7          bytes: 636
Total broadcasted           pkts : 1737       bytes: 181506
Total to BDI                pkts : 0          bytes: 0
Total injected              pkts : 1056       bytes: 105012
……
Total UUF suppression drop  pkts : 0          bytes: 0
Total sisf ctrl punt        pkts : 1577       bytes: 143058
```

Unknown Unicast Flooding Suppression

Use the following command to verify the unknown unicast flooding suppression status:

```
#show bridge-domain 12
Bridge-domain 12 (3 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 5 minute(s)
Unknown Unicast Flooding Suppression: Enabled
```

Debug Commands (EVPN)

Use the following debug command to troubleshoot:

EVPN Debug Commands

- debug l2vpn evpn event
- debug l2vpn evpn event detail

Event Trace Debug Commands

- monitor event-trace sequence-number
- monitor event-trace timestamps datetime msec localtime
- monitor event-trace evpn event size 1000000
- monitor event-trace evpn event include event error major detai
- show tech-support evpn

L2RIB Debug Command

- debug l2rib event

- debug l2rib event detail
- debug l2rib error

Debug Commands (SISF)

- debug device-tracking switcher
- debug device-tracking parser
- debug device-tracking flooding-suppression
- debug device-tracking hw-api
- show device-tracking events
- show device-tracking messages
- show device-tracking counters bd <bd-id>
- show tech-support sisf
- debug platform software fhs all
- Debug ip bgp all update
- debug ip bgp l2vpn evpn evi event [detail]
- debug ip bgp l2vpn evpn evi context [detail]

Advertising Proxy MAC-IP Route

Proxy MAC-IP route is enabled by default. Use the following command to disable proxy MAC-IP route:

```
l2vpn evpn
 multihoming proxy-mac-ip disable
```

Suppressing Unknown Unicast Flooding

Use the following command to suppress unknown unicast flooding:

```
flooding-suppression unknown-unicast
```

This command is supported only at the bridge domain level. By default, suppression of unknown unicast flooding is disabled.

Configuring Bridge Domain MAC Age Timer

Use the following command to configure the bridge domain MAC Age timer:

```
bridge-domain 11
 mac aging-time 10
```

The default aging time is 5 minutes for a bridge domain and 30 minutes for overlay bridge domains. The range is from 1 to 600 minutes.

Configuring ARP and ND Timers

Use the following command to configure the ARP timeout:

```
int BDI11
  arp timeout 600
```

Use the following command to configure the ND cache expiry:

```
int BDI12
  ipv6 nd cache expire 300
```

Configuring IP Local Learning, Limits, and Timers

Use the following command to disable IP local learning from the data plane:

```
l2vpn evpn
 ip local-learning disable
```

Use the following command to limit the number of locally learned IP addresses that can be stored:

```
ip local-learning limit per-mac ipv4
ip local-learning limit per-mac ipv6
```

The default number of IPv4 addresses is 4 and IPv6 addresses is 12.

Use the following command to configure timers:

```
ip local-learning time poll| reachable | stale time
```

The default polling interval is 1 minute, the reachable lifetime is 5 minutes, and the stale lifetime is 30 minutes.

Configuring ARP and ND Flooding Suppression

To configure the ARP and ND flooding suppression, perform the following steps.

Note By default, ARP/ND flooding suppression is enabled.

```
Device(config)#l2vpn evpn
Device(config-evpn)#flooding-suppression address-resolution ?
  disable  Disable flooding suppression
```

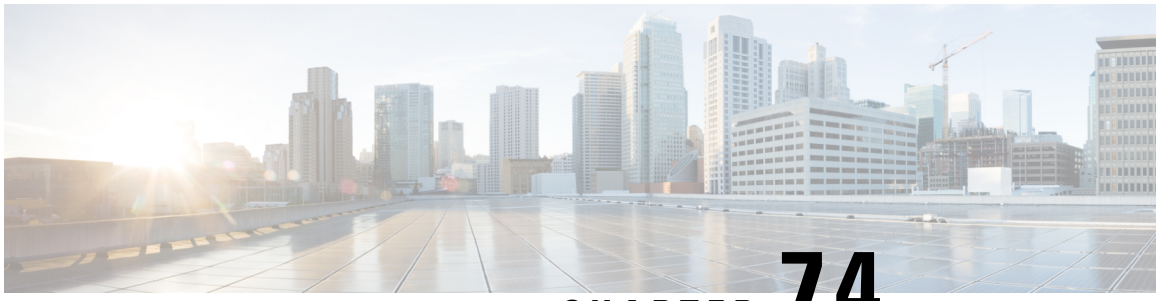
Additional References for EVPN Single-Homing

Standards and RFCs

Standard	Title
RFC 7432	BGP MPLS-Based Ethernet VPN

Feature Information for EVPN MPLS IRB with Distributed Anycast Gateways

Feature Name	Releases	Feature Information
EVPN MPLS IRB with Distributed Anycast Gateways	Cisco IOS XE Bengaluru 17.4.	The Ethernet VPN over MPLS Integrated Routing and Bridging (IRB) Single-Homing (SH) with Distributed Anycast Gateway feature provides support for symmetric IRB) model on SH Distributed Anycast Gateways for BGP EVPN over MPLS on Cisco ASR 1000 Series Aggregation Services Routers.
ARP and ND Flooding Suppression	Cisco IOS XE Bengaluru 17.4.	This feature helps to suppress the broadcast (ARP) or link-local multicast (NDP) messages circulating in the layer 2 domain, by either dropping them, or rewriting the layer2 destination from broadcast/multicast to unicast.
Support for EVPN over MPLS IRB Multi-Homing	Cisco IOS XE Cupertino 17.7.1a	This feature enables redundant network connectivity via Multi-homing by allowing a CE device to connect to more than one PE device therefore preventing disruptions in the network. Note that only dual-homing is supported in this release.



CHAPTER 74

Unknown Unicast Flooding Supperssion

This chapter describes how to configure unknown unicast flooding supperssion on the Cisco ASR 1000 Series Routers. This chapter contains these topics:

About Unknown Unicast Flooding on Bridge Domain

Occasionally, unknown unicast traffic is flooded to all the provider edge device because the device does not know the destination MAC address of a received packet. By default the unknown unicast traffic will be flooded to all the devices. To prevent forwarding such traffic, you can configure **unknown-unicast-suppress Suppress unknown unicast flooding** command.

Limitations for Unknown Unicast Suppression

If the Unknown Unicast Flooding Suppression is on, the unicast traffic towards an unknown host will be dropped. A host becomes unknown or silent when its MAC address ages out from the MAC address table on the PE. The PE might rely on the Unknown Unicast Flooding to re-learn the MAC address.

Enabling Unknown Unicast Flooding on Bridge Domain

To enable unknown unicast flooding suppression, perform the following steps.

Note By default, the unknown unicast flooding is disabled.

SUMMARY STEPS

1. **configure terminal**
2. **bridge-domain** {*interface number*}

3. **flooding-suppression unknown-unicast**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**bridge-domain** {*interface number*} **Example:** `Router(config)# bridge-domain 10`	10 Configures the bridge domain on the interface.
Step 3	**flooding-suppression unknown-unicast** **Example:** `Router(config-bdomain)# flooding-suppression unknown-unicast`	Enables unknown unicast flooding suppression on the bridge domain.
Step 4	**end** **Example:** `Router(config-bdomain)# end`	(Optional) Returns to privileged EXEC mode.

Verifying the Unknown Unicast Flooding Suppression

Verify that you have enabled the unknown unicst flooding suppression by entering the following command:

```
Device(config-bdomain)#do show run | sec bridge
bridge-domain 10
flooding-suppression unknown-unicast
```

This examples shows the packets that are suppressed and dropped.

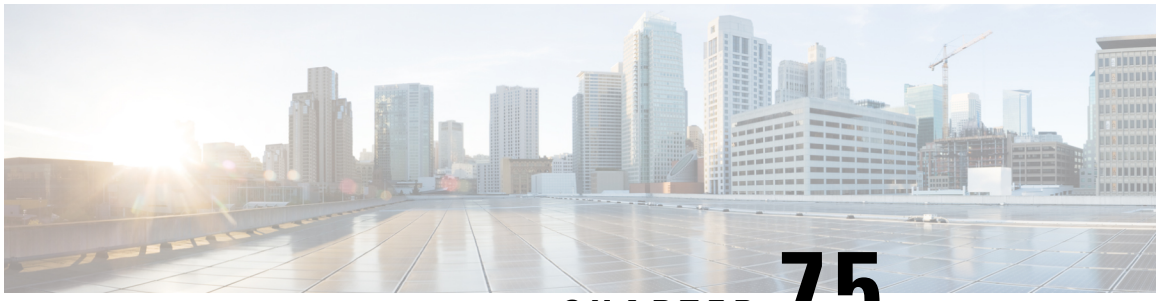
```
Device# show pla hard qfp ac fe bridge-domain datapath 1
QFP L2BD Bridge Domain information


BD id                  : 1
State enabled          : Yes
Aging timeout (sec)    : 300
Aging active entry     : Yes
Max mac limit          : 65536
Unkwn mac limit flood  : Yes
mac_learn_enabled      : Yes
mac_learn_controled    : Yes
Unknown unicast olist  : Yes
otv_aed_enabled : No
otv_enabled : No
mcast_snooping_enabled : No
Feature : evpn, uuf-suppression
```

```
SISF snoop protocols   : arp, ndp, dhcpv4, dhcpv6
Mac learned            : 0
BDI outer vtag         : 00000000
BDI inner vtag         : 00000000

Replication tree info:
  Global replication    : depth encode 0X2000001, (head 0X29D3D000)
  Split-horizon-group 0 : depth encode 00000000, (head 00000000)
  Split-horizon-group 1 : depth encode 00000000, (head 00000000)

Bridge Domain statistics

Total bridged               pkts : 0          bytes: 0
Total unknown unicast       pkts : 0          bytes: 0
Total broadcasted           pkts : 0          bytes: 0
Total to BDI                pkts : 0          bytes: 0
Total injected              pkts : 0          bytes: 0
Total mac-sec violation drop pkts : 0         bytes: 0
Total mac-sec move drop     pkts : 0          bytes: 0
Total mac-sec unknown drop  pkts : 0          bytes: 0
Total source filter drop    pkts : 0          bytes: 0
Total bfib policy drop      pkts : 0          bytes: 0
Total replication start drop pkts : 0         bytes: 0
Total recycle tail drop     pkts : 0          bytes: 0
Total static MAC move drop  pkts : 0          bytes: 0
Total BD disabled drop      pkts : 0          bytes: 0
Total STP state drop        pkts : 0          bytes: 0
Total UUF suppression drop  pkts : 0          bytes: 0
```

Feature Information for Unknown Unicast Flooding Suppression

Table 145: Feature Information for Unknown Unicast Flooding Suppression

Feature Name	Releases	Feature Information
Unknown Unicast Flooding Suppression	Cisco IOS XE Bengaluru 17.4	This feature was introduced.



CHAPTER 75

BGP EVPN over MultiProtocol Label Switching

The BGP EVPN over MultiProtocol Label Switching (MPLS) feature utilizes the functionality defined in RFC 7432 (BGP MPLS-based Ethernet VPN), to achieve EVPN over MPLS functionality between Provider Edge (PE) and Customer Edge (CE) devices.

Feature Information for BGP EVPN Over MPLS

Feature Name	Releases	Feature Information
BGP EVPN Over MPLS	Cisco IOS XE Fuji 16.9.x	The BGP EVPN over MPLS feature utilizes the functionality defined in RFC 7432, to achieve EVPN over MPLS functionality between Provider Edge (PE) and Customer Edge (CE) devices.

Information about BGP EVPN over MultiProtocol Label Switching

BGP MPLS based Ethernet VPN (EVPN) Overview

EVPN (RFC 7432) addresses the following requirements:

- PE node redundancy with load-balancing based on L2/L3/L4 flows from CE to PE.
- Flow-based multi-pathing of traffic from local PE to remote PEs across core and vice-versa
- Geo-redundant PE nodes with optimum unicast forwarding.

- • Flexible redundancy grouping, where a PE can be a member of multiple redundancy groups, each containing a different set of CEs.

EVPN Building Blocks

There are three fundamental building blocks for EVPN technology, EVPN Instance (EVI), Ethernet Segment (ES), EVPN BGP routes and extended communities:

- EVI is a VPN connection on a PE router. It is the equivalent of IP VPN Routing and Forwarding (VRF) in Layer 3 VPN. It is also known as MAC-VRF.
- ES is a connection with a customer site (device or network) and is associated with access-facing interfaces. Access-facing interfaces are assigned unique IDs that are referred to as Ethernet Segment Identifiers (ESI). A site can be connected to one or more PEs. The ES connection has the same ESI in each PE connected to the site.
- RFC 7432 defines routes and extended communities to enable EPVN support. In Cisco IOS XE Fuji 16.8.x Software Release, Route Type 2 and Route Type 3 are supported.

  In BGP MPLS-based EVPN, an EVI is configured for every PE device for each customer associated with the PE device. In this case, a customer is any customer edge device that is attached to the PE device. The CE device can be a host, a switch or a router. Each EVI has a unique Route Distinguisher (RD) and one or more Route Targets (RT).

  For EVPN Single-Homing feature, a CE device is attached to a single PE device and has an Ethernet Segment with ESI=0.

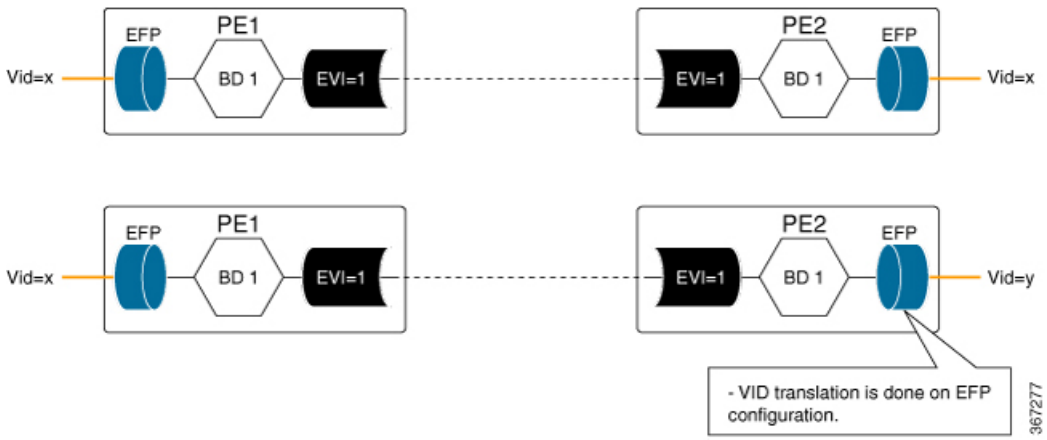
Service Interfaces

The following are types of EVPN VLAN service interfaces:

VLAN-based Service Interface

In VLAN-based service interface, each VLAN is associated to one bridge domain and one EVI.

Figure 111: VLAN-Based Service Interface



For VLAN-based Service Interface, Type 1 Route Distinguisher, a unique number used to distinguish identical routes in different VRFs, is used for EVIs as recommended by the RFC 7432. The Route Distinguishers and Router Targets, which are used to share routes between different VRFs, are autogenerated to ensure unique Route Distinguisher numbers across EVIs.

VLAN Bundle Service Interface

In VLAN Bundle Service Interface, multiple VLANs share the same bridge table.

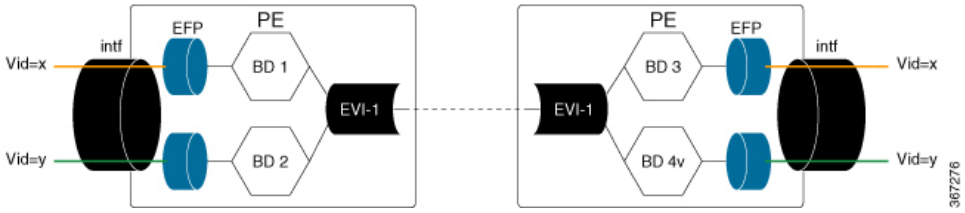
Figure 112: VLAN Bundle Service Interface

Each EVPN instance corresponds to multiple broadcast domains maintained in a single bridge table per MAC-VRF. For VLAN Bundle Service Interface service to work, MAC addresses must be unique across all VLANs for an EVI.

VLAN-Aware Bundle Service Interface

For VLAN-aware Bundle Service Interface, each VLAN is associated with one bridge domain, but there can be multiple bridge domains associated with one EVI.

Figure 113: VLAN-Aware Bundle Service Interface



An EVPN instance consists of multiple broadcast domains where each VLAN has one bridge table. Multiple bridge tables (one per VLAN) are maintained by a single MAC-VRF that corresponds to the EVPN instance.

BGP EVPN over MPLS Inter-AS and Prefix SID

BGP EVPN over MPLS Route Target 5 Inter-Autonomous Systems (Inter-AS) Option C is used to advertise the site prefixes across the network that forms the overlay. BGP Segment Routing Prefix segment identifiers (SID) is used to advertise the Node SID of the PE device sourcing the prefix across the network that forms the underlay.

Inter-AS Option C for EVPN

The BGP EVPN over MPLS feature supports InterAS option C for EVPN session spanning across AS domains. When the session spans across multiple AS domains, the next hop of the BGP EVPN update that is set by the egress PE remains unchanged across the AS domains, and is be the next hop for the underlay for ingress (headend) PE. Underlay could be either established through BGP Labeled Unicast (BGP-LU) across AS domains or by a centralised controller.

Route Type 5 – IP Prefix

BGP EVPN over MPLS feature implements Route Type 5, as defined by RFC 7432.

The BGP EVPN Route Type 5 update carries IP prefixes and includes a tunnel encapsulation attribute that indicates the VPN session encapsulation. Absence of tunnel encapsulation attribute in BGP EVPN Route Target 5 update indicates MPLS VPN encapsulation for the BGP EVPN Route Type 5 VPN prefix.

Route Type 5 is used to advertise prefixes independently of the MAC advertisement routes, and possible prefix next-hops in the overlay topology, namely -ESI, IRB IP address, Floating IP address.

Route type 5 implementation has the following features:

- Prefix advertisement is not linked to MAC mobility.
- Network Virtual Interfaces (NVE) that have EVPN configured but do not support the optional prefix-advertisement route, can easily identify the route and ignore it without processing the route.
- When selecting routes, MAC information is not compared by BGP.
- Flexible overlay next-hop (IRB, floating IP, ESI) can be configure to address different end-point types.

The below table shows the format of the prefix-advertisement:

Table 146: Prefix Advertisement Route Format

Field	Value	LEength (Octets)
Route Type	0x05	1
Length	25	1
EVI RD	Type 1 (IPv4Addr) RD unique across all EVIs on the PE.	8
ESI	Ethernet Segment Identifier	10
Ethernet Tag	0 or valid Ethernet Tag	4
IP Address	Gateway IP Address	4 or 16
Label	Valid MPLS label allocated per [EVI, ESI, EtherTag] tuple	3
EVI RT	Type 0 (2byteAS) route target	8

Importing IP Routes to EVPN

To import locally sourced (redistributed) and provider edge to customer edge BGP IPv4 and IPv6 routes into EVPN, add these routes to VPNv4 table using the following configuration:

```
router bgp
address-family {ipv4 | ipv6} vrf <vrfname>
advertise l2vpn evpn
```

To import VPNv4 or VPNv6 routes learned from a neighbour into EVPN, use the following configuration:

```
router bgp
address-family l2vpn evpn
import {vpnv4 | vpnv6} unicast
```

To add EVPN routes learned from an EVPN neighbor to EVPN the table, edo one of the following:

- configure no bgp default route-target filter command in the router BGP configuration
- add a route-target in EVPN route matching the stitching configuration under VRF address-family

Route Reoriginate

When the EVPN prefix is imported into the VPN VRF table, a new VPN prefix may be reoriginated. Updates may be sent to another VPN neighbor based on the configuration. The reoriginated prefix is the imported VPN prefix where the route target extended communities are replaced with a new set of route target extended communities.

To configure route reorigination, use the following configuration:

```
enable
configure terminal
router bgp 100
 address-family vpnv4|vpnv6
  import l2vpn evpn [re-originate [stitching-rt]]
```

When the **import {vpnv4 | vpnv6} unicast** command is configured under the L2VPN EVPN address-family configuration mode and if the route targets of the VPNv4 or VPNv6 route are within the export stitching route targets, the reoriginated EVPN prefix retains the same route targets. Otherwise, route targets of reoriginated EVPN prefix are swapped using export stitching route target configuration.

When the **import {vpnv4 | vpnv6} unicast re-originate** command is configured under the L2VPN EVPN address-family, the reoriginated EVPN prefix route target is replaced by the export stitching route targets.

When the **import {vpnv4 | vpnv6} unicast** command is not configured, then remote VPNv4 or VPNv6 prefix is not imported into EVPN table.

When the **import l2vpn evpn** command is configured under the VPNv4 or VPNv6 address family and if the route targets of the EVPN prefix are within the subset of normal VRF export route targets in VRF, the imported VPNv4 or VPNv6 prefix retains the same route targets. Otherwise, the imported VPNv4 or VPNv6 route targets are swapped with the VRF export route targets and advertised to VPNv4/VPNv6 neighbors.

When the **import l2vpn evpn re-originate** command is configured under the VPNv4 or VPNv6 address-family, the route targets of the imported VPNv4 or VPNv6 prefix are replaced with the VRF export route targets and advertised to VPNv4/VPNv6 neighbors.

When the **import l2vpn evpn re-originate stitching-rt** command is configured, the imported VPN prefix route target is replaced by the VRF import stitching route target and advertised to VPNv4 or VPNv6 neighbors.

Note When the **import l2vpn evpn** command is not configured, the imported VPN prefix is not advertised to VPN neighbors.

EVPN Encapsulation

You can choose the EVPN encapsulation using the neighbour EVPN neighbor configuration by including the encapsulation in the EVPN updates to the neighbor:

```
enable
configure terminal
 router bgp 100
  address-family l2vpn evpn
   neighbor <address> encap {mpls | vxlan}
```

Note

- If VxLan is also configured for the VRF IPv4 address family, EVPN routes imported from VPNv4 table could have both MPLS and VxLAN encapsulation data. EVPN routes imported from VPNv6 table would only have MPLS Encap as VxLAN is not supported for VPNv6 prefixes.
- If EVPN neighbor is configured to only send MPLS encapsulation, then the prefix is advertised to neighbour, only if MPLS encapsulation data with the prefix exists.
- If EVPN neighbor is configured to only send VxLAN encapsulation, then the prefix is advertised to neighbor only if there exists VxLAN encap data with the prefix.
- If EVPN neighbor is configured to only send VxLAN encapsulation, then the prefix is advertised to neighbor only if there exists VxLAN encap data with the prefix.

How to Configure BGP EVPN over MultiProtocol Label Switching

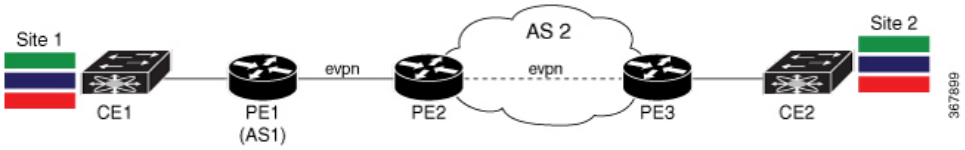
Configuring BGP over MPLS

Following is sample configuration for configuring BGP over MPLS

```
enable
 configure terminal
  router bgp 100
    template peer-policy policy1
     encapsulation mpls  ====> template policy config for neighbor encapsulation preference

      address-family l2vpn evpn
       neighbor 10.1.1.1 encapsulation mpls ===> neighbor encapsulation preference
       exit
      address-family ipv6 vrf vrf1
       advertise l2vpn evpn 2000 ===> import locally sourced or PE-CE vpnv6 routes into
evpn and the limit on the number of prefixes that can be imported (Optional).
       network 0::0/0 evpn ===> only import default route into evpn from this vrf ipv6 table
```

Configuring BGP EVPN over MPLS (Inter AS)

Figure 114: BGP over MPLS (Inter AS)



PE1 Configuration

```
vrf def red
     address-family ipv4
   route-target export  1:1 stitching
   route-target import  2:2 stitching

     address-family ipv6
   route-target export  1:1 stitching
   route-target import  2:2 stitching

  router bgp 1
     address-family l2vpn evpn
  neighbor 192.0.2.10 activate
  neighbor 192.0.2.10 encap mpls

     address-family ipv4 vrf red
  advertise l2vpn evpn
     address-family ipv6 vrf red
advertise l2vpn evpn
```

PE2 Configuration

```
router bgp 1
     no bgp default route-target filter
     address-family l2vpn evpn
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 encap mpls
  neighbor 10.1.1.1 next-hop-unchanged
  neighbor 203.0.113.10 activate
  neighbor 203.0.113.10 encap mpls
  neighbor 203.0.113.10 next-hop-unchanged
```

PE3 Configuration

```
vrf def red
     address-family ipv4
   route-target export  2:2 stitching
   route-target import  1:1 stitching
     address-family ipv6
   route-target export  2:2 stitching
   route-target import  1:1 stitching

  router bgp 1
     address-family l2vpn evpn
  neighbor 192.0.2.10 encap mpls
     address-family ipv4 vrf red
  advertise l2vpn evpn
     address-family ipv6 vrf red
  advertise l2vpn evpn
```

Configuring BGP EVPN over MPLS (InterAS L3VPN)

PE1 Configuration

```
vrf def red
     address-family ipv4
   route-target export  1:1 stitching
   route-target import  2:2 stitching

     address-family ipv6
   route-target export  1:1 stitching
   route-target import  2:2 stitching

  router bgp 1
     address-family l2vpn evpn
   neighbor 192.0.2.10 activate
  neighbor 192.0.2.10 encap mpls

     address-family ipv4 vrf red
  advertise l2vpn evpn
     address-family ipv6 vrf red
  advertise l2vpn evpn
```

PE2 Configuration

```
vrf def red
     address-family ipv4
   route-target import  2:2
   route-target import  1:1 stitching
     address-family ipv6
   route-target import  2:2
   route-target import  1:1 stitching

  router bgp 1
     address-family l2vpn evpn
   import vpnv4 unicast
  import vpnv6 unicast
  neighbor 10.1.1.1  encap mpls
  neighbor 10.1.1.1  next-hop-unchanged
     address-family vpnv4
  import l2vpn evpn
  neighbor 203.0.113.10 activate
  neighbor 203.0.113.10 next-hop-unchanged

     address-family vpnv6
  import l2vpn evpn
  neighbor 203.0.113.10 activate
  neighbor 203.0.113.10 next-hop-unchanged
```

PE3 Configuration

```
vrf def red
     address-family ipv4
   route-target export  2:2
   route-target import  1:1
address-family ipv6
   route-target export  2:2
   route-target import  1:1
    router bgp 1
address-family vpnv4
  neighbor 192.0.2.10 activate
```

Configuration Examples for BGP EVPN over MPLS

Verifying EVPN Neighbor

Use the following command to verify EVPN Neighbour configuration when no encapsulation is configured:

```
PE3# show bgp l2vpn evpn neighbors
For address family: L2VPN E-VPN
  Session: 192.0.2.10
  BGP table version 38, neighbor version 38/0
  Output queue size : 0
  Index 1, Advertise bit 0
  1 update-group member
  Community attribute sent to this neighbor
  Extended-community attribute sent to this neighbor
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
  Prefers VxLAN if VTEP is UP else MPLS
```

Use the following command to verify EVPN Neighbour configuration when MPLS encapsulation is configured:

```
PE3# show bgp l2vpn evpn neighbors 10.1.1.1
 For address family: L2VPN E-VPN
  Session: 10.1.1.1
  BGP table version 38, neighbor version 1/38
  Output queue size : 0
  Index 0, Advertise bit 0
  Community attribute sent to this neighbor
  Extended-community attribute sent to this neighbor
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
  Prefers MPLS
```

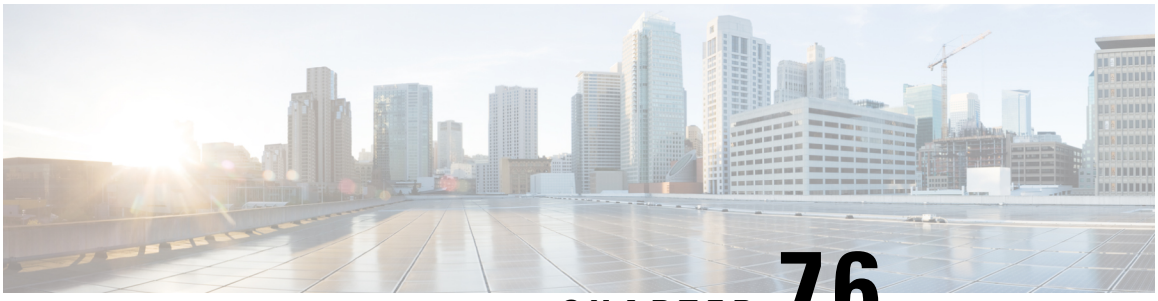
Additional References for BGP EVPN over MultiProtocol Label Switching

Standards and RFCs

Standard	Title
RFC 7432	BGP MPLS-Based Ethernet VPN

PART VI

MPLS Layer 3 VPNs



CHAPTER 76

MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

The Multiprotocol Label Switching (MPLS) VPN Inter-AS with Autonomous System Boundary Routers (ASBRs) Exchanging VPN-IPv4 Addresses feature allows a MPLS VPN to span service providers and autonomous systems. This module explains how to enable ASBRs to use Exterior Border Gateway Protocol (EBGP) to exchange IPv4 Network Layer Reachability Information (NLRI) in the form of VPN-IPv4 addresses.

Prerequisites for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

- Before you configure Exterior Border Gateway Protocol (EBGP) routing between autonomous systems or subautonomous systems in an Multiprotocol Label Switching (MPLS) VPN, ensure that you have properly configured all MPLS VPN routing instances and sessions. The configuration tasks outlined in this section build from those configuration tasks. Perform the following tasks as described in the Configuring MPLS Layer 3 VPNs module:
  - Define VPN routing instances
  - Configure BGP routing sessions in the MPLS core
  - Configure provider-edge-provider-edge (PE-to-PE) routing sessions in the MPLS core
  - Configure BGP provider-edge-customer-edge (PE-to-CE) routing sessions
  - Configure a VPN-IPv4 EBGP session between directly connected Autonomous System Boundary Routers (ASBRs)

Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Multihop VPN-IPv4 Exterior Border Gateway Protocol (EBGP) is not supported.

Information About MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

MPLS VPN Inter-AS Introduction

An autonomous system is a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. Also, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless to the customer.

Benefits of MPLS VPN Inter-AS

An MultiprotocolLabel Switching (MPLS) VPN Inter-AS provides the following benefits:

- Allows a VPN to cross more than one service provider backbone: Service providers running separate autonomous systems can jointly offer MPLS VPN services to the same customer. A VPN can begin at one customer site and traverse different VPN service provider backbones before arriving at another site of the same customer. Previously, MPLS VPN could traverse only a single Border Gateway Protocol (BGP) autonomous system service provider backbone. This feature allows multiple autonomous systems to form a continuous (and seamless) network between customer sites of a service provider.

- Allows a VPN to exist in different areas: A service provider can create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.

- Allows confederations to optimize Internal Border Gateway Protocol (IBGP) meshing: IBGP meshing in an autonomous system is more organized and manageable. An autonomous system can be divided into multiple, separate subautonomous systems and then classify them into a single confederation (even though the entire VPN backbone appears as a single autonomous system). This capability allows a service provider to offer MPLS VPNs across the confederation because it supports the exchange of labeled VPN-IPv4 Network Layer Reachability Information (NLRI) between the subautonomous systems that form the confederation.

Use of Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Separate autonomous systems from different service providers can communicate by exchanging IPv4 Network Layer Reachability Information (NLRI) in the form of VPN-IPv4 addresses. The Autonomous System Border Routers (ASBRs) use Exterior Border Gateway Protocol (EBGP) to exchange network reachability information. Then an Interior Gateway Protocol (IGP) distributes the network layer information for VPN-IPv4 prefixes throughout each VPN and each autonomous system. Routing information uses the following protocols:

- Within an autonomous system, routing information is shared using an IGP.
- Between autonomous systems, routing information is shared using an EBGP. An EBGP allows a service provider to set up an interdomain routing system that guarantees the loop-free exchange of routing information between separate autonomous systems.

The primary function of an EBGP is to exchange network reachability information between autonomous systems, including information about the list of autonomous system routes. The autonomous systems use EBGP border edge devices to distribute the routes, which include label switching information. Each border edge device rewrites the next hop and labels. See the Information Exchange in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, on page 1461 section for more information.

Interautonomous system configurations supported in an MPLS VPN are as follows:

- **Interprovider VPN--** MPLS VPNs that include two or more autonomous systems, connected by separate border edge devices. The autonomous systems exchange routes using EBGP. No IGP or routing information is exchanged between the autonomous systems.
- **BGP confederations--** MPLS VPNs that divide a single autonomous system into multiple subautonomous systems, and classify them as a single, designated confederation. The network recognizes the confederation as a single autonomous system. The peers in the different autonomous systems communicate over EBGP sessions; however, they can exchange route information as if they were IBGP peers.

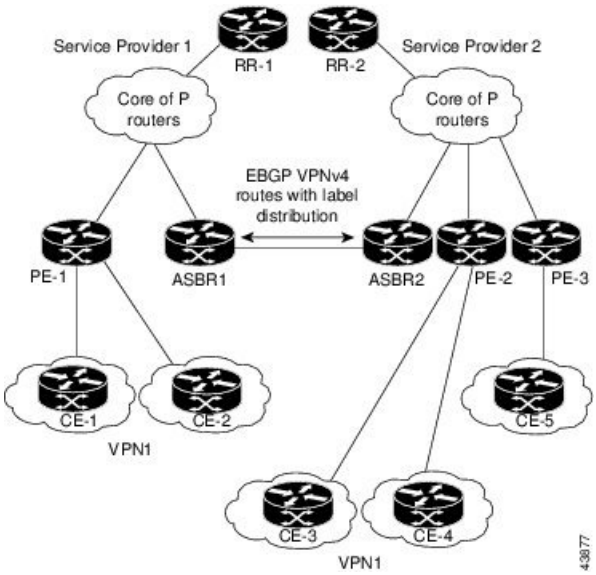
Information Exchange in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

This section contains the following topics:

Transmission of Information in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

The figure below illustrates an Multiprotocol Label Switching (MPLS) VPN consisting of two separate autonomous systems. Each autonomous system operates under different administrative control and runs a different Interior Gateway Protocol (IGP). Service providers exchange routing information through Exterior Border Gateway Protocol (EBGP) border edge devices (ASBR1, ASBR2).

Figure 115: EBGP Connection Between Two MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



This configuration uses the following process to transmit information:

SUMMARY STEPS

1. The provider edge device (PE-1) assigns a label for a route before distributing that route. The PE device uses the multiprotocol extensions of Border Gateway Protocol (BGP) to transmit label mapping information. The PE device distributes the route as a VPN-IPv4 address. The address label and the VPN identifier are encoded as part of the IPv4 Network Layer Reachability Information (NLRI).
2. The two route reflectors (RR-1 and RR-2) reflect VPN-IPv4 internal routes within the autonomous system. The border edge devices (ASBR1 and ASBR2) of the autonomous systems advertise the VPN-IPv4 external routes.
3. The EBGP border edge device (ASBR1) redistributes the route to the next autonomous system (ASBR2). ASBR1 specifies its own address as the value of the EBGP next-hop attribute and assigns a new label. The address ensures the following:
4. The EBGP border edge device (ASBR2) redistributes the route in one of the following ways, depending on its configuration:

DETAILED STEPS

Step 1 The provider edge device (PE-1) assigns a label for a route before distributing that route. The PE device uses the multiprotocol extensions of Border Gateway Protocol (BGP) to transmit label mapping information. The PE device distributes the route as a VPN-IPv4 address. The address label and the VPN identifier are encoded as part of the IPv4 Network Layer Reachability Information (NLRI).

Step 2 The two route reflectors (RR-1 and RR-2) reflect VPN-IPv4 internal routes within the autonomous system. The border edge devices (ASBR1 and ASBR2) of the autonomous systems advertise the VPN-IPv4 external routes.

Step 3 The EBGP border edge device (ASBR1) redistributes the route to the next autonomous system (ASBR2). ASBR1 specifies its own address as the value of the EBGP next-hop attribute and assigns a new label. The address ensures the following:

- The next-hop device is always reachable in the service provider (P) backbone network.

- Tthe label assigned by the distributing device is properly interpreted. (The label associated with a route must be assigned by the corresponding next-hop device.)

Step 4 The EBGP border edge device (ASBR2) redistributes the route in one of the following ways, depending on its configuration:

- If the IBGP neighbors are configured with the **neighbor next-hop-self** command, ASBR2 changes the next-hop address of updates received from the EBGP peer, then forwards it.
- If the IBGP neighbors are not configured with the **neighbor next-hop-self** command, the next-hop address does not change. ASBR2 must propagate a host route for the EBGP peer through the IGP. To propagate the EBGP VPN-IPv4 neighbor host route, use the **redistribute connected subnets** command. The EBGP VPN-IPv4 neighbor host route is automatically installed in the routing table when the neighbor comes up. This is essential to establish the label switched path between PE devices in different autonomous systems.

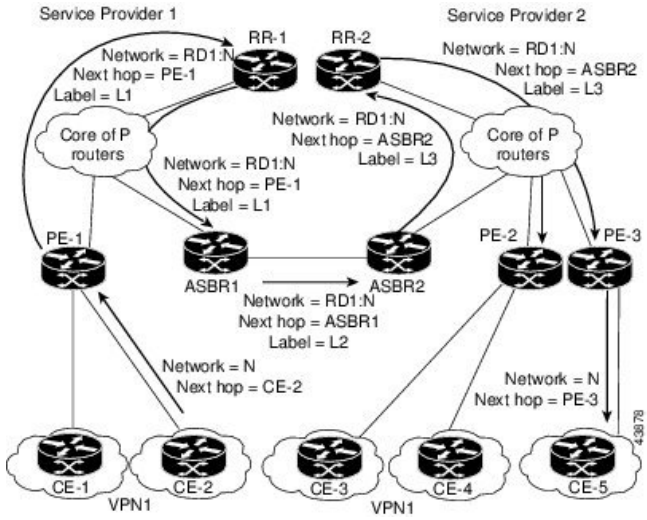
Exchange of VPN Routing Information in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Autonomous systems exchange VPN routing information (routes and labels) to establish connections. To control connections between autonomous systems, the provider edge (PE) devices and Exterior Border Gateway Protocol (EBGP) border edge devices maintain a Label Forwarding Information Base (LFIB). The LFIB manages the labels and routes that the PE devices and EBGP border edge devices receive during the exchange of VPN information.

The figure below illustrates the exchange of VPN route and label information between autonomous systems. The autonomous systems use the following conditions to exchange VPN routing information:

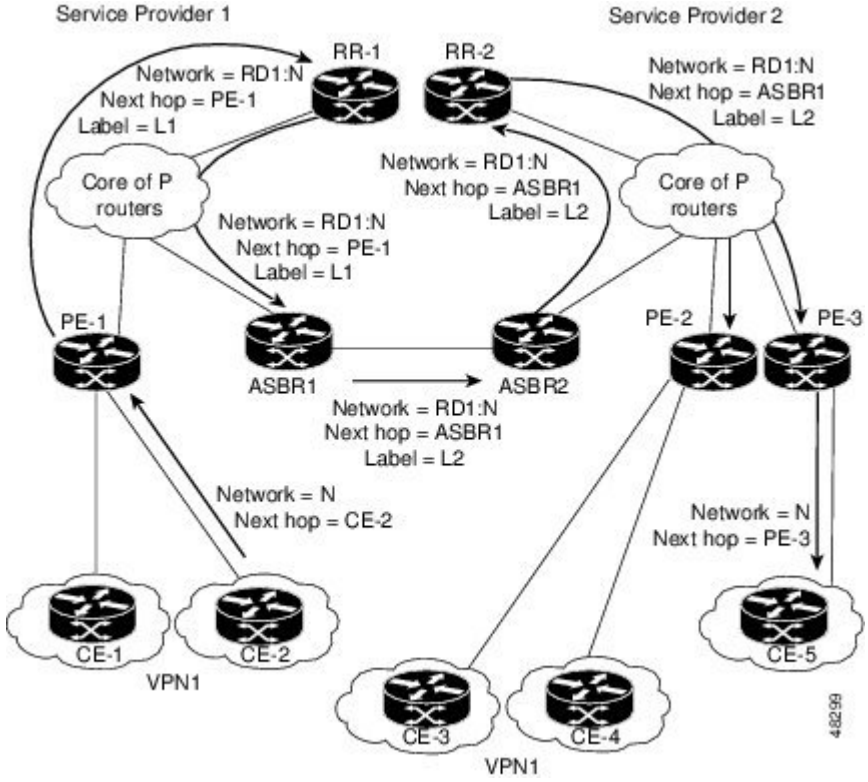
- Routing information includes:
  - The destination network (N)
  - The next-hop field associated with the distributing device
  - A local MPLS label (L)
- An RD1: route distinguisher is part of a destination network address. It makes the VPN-IPv4 route globally unique in the VPN service provider environment.
- The Autonomous System Border Routers (ASBRs) are configured to change the next-hop (next hop-self) when sending VPN-IPv4 Network Layer Reachability Information (NLRI) to the Internal Border Gateway Protocol (IBGP) neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to the IBGP neighbors.

Figure 116: Exchanging Routes and Labels Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



The figure below illustrates the exchange of VPN route and label information between autonomous systems. The only difference is that ASBR2 is configured with the **redistribute connected** command, which propagates the host routes to all PEs. The **redistribute connected** command is necessary because ASBR2 is not configured to change the next-hop address.

Figure 117: Exchanging Routes and Labels with the redistribute connected Command in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses



Packet Forwarding Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses

The figure below illustrates how packets are forwarded between autonomous systems in an interprovider network using the following packet forwarding method.

Packets are forwarded to their destination by means of Multiprotocol Label Switching (MPLS). Packets use the routing information stored in the Label Forwarding Information Base (LFIB) of each provider edge (PE) device and Exterior Border Gateway Protocol (EBGP) border edge device.

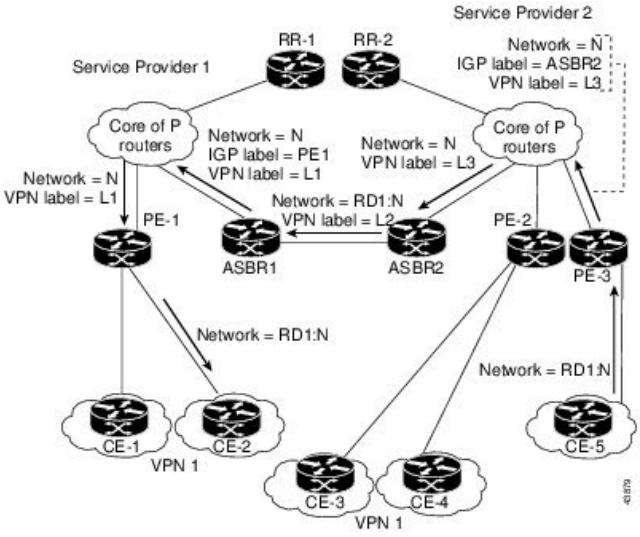
The service provider VPN backbone uses dynamic label switching to forward labels.

Each autonomous system uses standard multilevel labeling to forward packets between the edges of the autonomous system devices (for example, from CE-5 to PE-3). Between autonomous systems, only a single level of labeling is used, corresponding to the advertised route.

A data packet carries two levels of labels when traversing the VPN backbone:

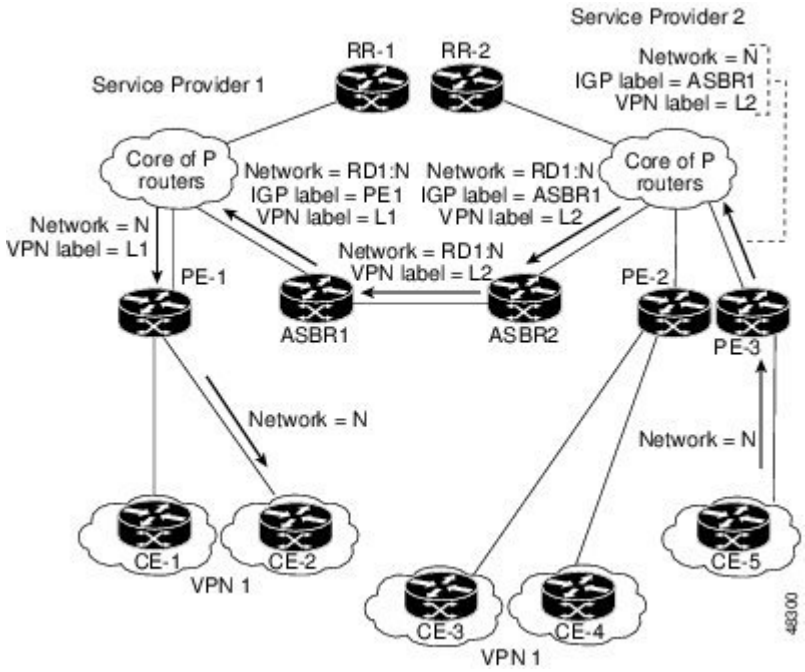
- The first label (IGP route label) directs the packet to the correct PE device or EBGP border edge device. (For example, the Interior Gateway Protocol (IGP) label of ASBR2 points to the ASBR2 border edge device.)
- The second label (VPN route label) directs the packet to the appropriate PE device or EBGP border edge device.

Figure 118: Forwarding Packets Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



The figure below shows the same packet forwarding method as described in the figure above, except the EBGP device (ASBR1) forwards the packet without reassigning it a new label.

Figure 119: Forwarding Packets Without a New Label Assignment Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



Use of a Confederation for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

A confederation is a collection of multiple subautonomous systems that are grouped together. A confederation reduces the total number of peer devices in an autonomous system. A confederation divides an autonomous system into subautonomous systems and assigns a confederation identifier to the autonomous systems. A VPN can span service providers running in separate autonomous systems or in multiple subautonomous systems that form a confederation.

In a confederation, each subautonomous system is fully meshed with other subautonomous systems. The subautonomous systems communicate using an Interior Gateway Protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Each subautonomous system also has an Exterior Border Gateway Protocol (EBGP) connection to the other subautonomous systems. The confederation EBGP (CEBGP) border edge devices forward next-hop-self addresses between the specified subautonomous systems. The next-hop-self address forces the Border Gateway Protocol (BGP) to use a specified address as the next hop rather than letting the protocol choose the next hop.

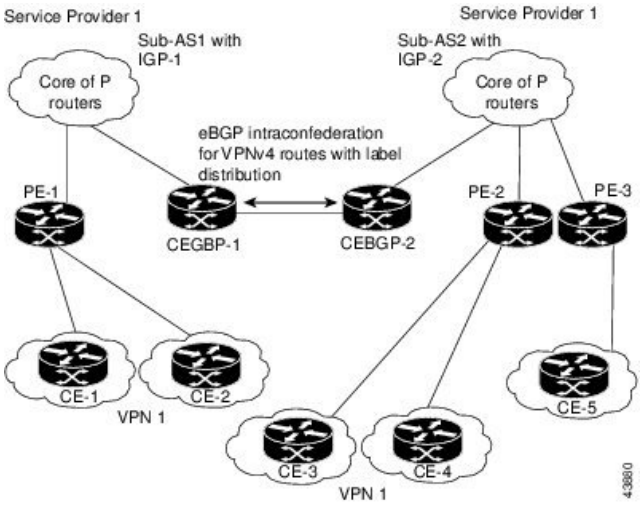
You can configure a confederation with separate subautonomous systems in either of two ways:

- You can configure a device to forward next-hop-self addresses between only the CEBGP border edge devices (both directions). The subautonomous systems (IBGP peers) at the subautonomous system border do not forward the next-hop-self address. Each subautonomous system runs as a single IGP domain. However, the CEBGP border edge device addresses are known in the IGP domains.
- You can configure a device to forward next-hop-self addresses between the CEBGP border edge devices (both directions) and within the IBGP peers at the subautonomous system border. Each subautonomous system runs as a single IGP domain but also forwards next-hop-self addresses between the PE devices in the domain. The CEBGP border edge device addresses are known in the IGP domains.

The figure below illustrates a typical MPLS VPN confederation configuration. In this confederation configuration:

- The two CEBGP border edge devices exchange VPN-IPv4 addresses with labels between the two subautonomous systems.
- The distributing device changes the next-hop addresses and labels and uses a next-hop-self address.
- IGP-1 and IGP-2 know the addresses of CEBGP-1 and CEBGP-2.

Figure 120: EBGP Connection Between Two Subautonomous Systems in a Confederation



In this confederation configuration:

- CEBGP border edge devices function as neighboring peers between the subautonomous systems. The subautonomous systems use EBGP to exchange route information.
- Each CEBGP border edge device (CEBGP-1, CEBGP-2) assigns a label for the route before distributing the route to the next subautonomous system. The CEBGP border edge device distributes the route as a VPN-IPv4 address by using the multiprotocol extensions of BGP. The label and the VPN identifier are encoded as part of the IPv4 Network Layer Reachability Information (NLRI).
- Each provider edge (PE) and CEBGP border edge device assigns its own label to each VPN-IPv4 address prefix before redistributing the routes. The CEBGP border edge devices exchange VPN-IPv4 addresses with the labels. The next-hop-self address is included in the label (as the value of the EBGP next-hop attribute). Within the subautonomous systems, the CEBGP border edge device address is distributed throughout the IBGP neighbors, and the two CEBGP border edge devices are known to both confederations.

How to Configure MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Configuring the ASBRs to Exchange VPN-IPv4 Addresses

To configure an Exterior Border Gateway Protocol (EBGP) Autonomous System Border Router (ASBR) to exchange VPN-IPv4 routes with another autonomous system, perform this task.

Note Issue the **redistribute connected subnets** command in the Interior Gateway Protocol (IGP) configuration portion of the device to propagate host routes for VPN-IPv4 EBGP neighbors to other devices and provider edge devices. Alternatively, you can specify the next-hop-self address when you configure Internal Border Gateway Protocol (IBGP) neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **address-family vpnv4** [**unicast**]
6. **neighbor** *peer-group-name* **remote-as** *as-number*
7. **neighbor** *peer-group-name* **activate**
8. **exit-address-family**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 1`	Creates an EBGP routing process and assigns it an autonomous system number. • The autonomous system number is passed along and identifies the device to EBGP devices in another autonomous system.

	Command or Action	Purpose
Step 4	**no bgp default route-target filter** **Example:** `Device(config)# no bgp default route-target filter`	Disables BGP route-target filtering and places the device in configuration mode. • All received BGP VPN-IPv4 routes are accepted by the device.
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Configures a routing session to carry VPNv4 addresses across the VPN backbone and places the device in address family configuration mode. • Each address has been made globally unique by the addition of an 8-byte route distinguisher (RD). • The unicast keyword specifies a unicast prefix.
Step 6	**neighbor** *peer-group-name* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 1 remote-as 2`	Enters the address family configuration mode and specifies a neighboring EBGP peer group. • This EBGP peer group is identified to the specified autonomous system.
Step 7	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor 1 activate`	Activates the advertisement of the VPNv4 address family to a neighboring EBGP device.
Step 8	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits from the address family submode of the router configuration mode.
Step 9	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring EBGP Routing to Exchange VPN Routes Between Subautonomous Systems in a Confederation

Perform this task to configure EBGP routing to exchange VPN routes between subautonomous systems in a confederation.

Note To ensure that the host routes for VPN-IPv4 EBGP neighbors are propagated (by means of the IGP) to the other devices and provider edge devices, specify the **redistribute connected** command in the IGP configuration portion of the CEBGP device. If you are using OSPF, make sure that the OSPF process is not enabled on the CEBGP interface where the "redistribute connected" subnet exists.

Note In this confederation, subautonomous system IGP domains must know the addresses of CEBGP-1 and CEBGP-2. If you do not specify a next-hop-self address as part of the router configuration, ensure that the addresses of all PE devices in the subautonomous system are distributed throughout the network, not just the addresses of CEBGP-1 and CEBGP-2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *sub-autonomous-system*
4. **bgp confederation identifier** *as-number*
5. **bgp conferderation peers** *sub-autonomous-system*
6. **no bgp default route-target filter**
7. **address-family vpnv4** [**unicast**]
8. **neighbor** *peer-group-name* **remote-as** *as-number*
9. **neighbor** *peer-group-name* **next-hop-self**
10. **neighbor** *peer-group-name* **activate**
11. **exit-address-family**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *sub-autonomous-system* **Example:** `Device(config)# router bgp 2`	Creates an EBGP routing process and assigns it an autonomous system number and enters the device in configuration mode. • The subautonomous system number is passed along to identify the device to EBGP devices in other subautonomous systems.
Step 4	**bgp confederation identifier** *as-number* **Example:** `Device(config-router)# bgp confederation identifier 100`	Defines an EBGP confederation by specifying a confederation identifier associated with each subautonomous system. • The subautonomous systems appear as a single autonomous system.

	Command or Action	Purpose
Step 5	**bgp conferderation peers** *sub-autonomous-system* **Example:** `Device(config-router)# bgp confederation peers 1`	Specifies the subautonomous systems that belong to the confederation (identifies neighbors of other subautonomous systems within the confederation as special EBGP peers).
Step 6	**no bgp default route-target filter** **Example:** `Device(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received BGP VPN-IPv4 routes are accepted by the device.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Configures a routing session to carry VPNv4 addresses across the VPN backbone. Each address is made globally unique by the addition of an 8-byte RD. Enters address family configuration mode. • The **unicast** keyword specifies a unicast prefix.
Step 8	**neighbor** *peer-group-name* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 1 remote-as 1`	Enters the address family configuration mode and specifies a neighboring EBGP peer group. • This EBGP peer group is identified to the specified subautonomous system.
Step 9	**neighbor** *peer-group-name* **next-hop-self** **Example:** `Device(config-router-af)# neighbor 1 next-hop-self`	Advertises the device as the next hop for the specified neighbor. • If a next-hop-self address is specified as part of the router configuration, the **redistribute connected** command need not be used.
Step 10	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor R activate`	Activates the advertisement of the VPNv4 address family to a neighboring PE device in the specified subautonomous system.
Step 11	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits from the address family submode of the router configuration mode.
Step 12	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Perform this task to display the VPN-IPv4 Label Forwarding Information Base (LFIB) entries.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
3. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
4. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all**	**rd** *route-distinguisher*	**vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Device# show ip bgp vpnv4 all labels`	Displays VPN address information from the BGP table. • Use the **all** and **labels** keywords to display information about all VPNv4 labels.
Step 3	**show mpls forwarding-table** [*network* {*mask*	*length*}	**labels** *label* [*-label*]	**interface** *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB (such as VPNv4 prefix/length and BGP next-hop destination for the route).
Step 4	**disable** **Example:** `Device# disable`	Returns to user EXEC mode.

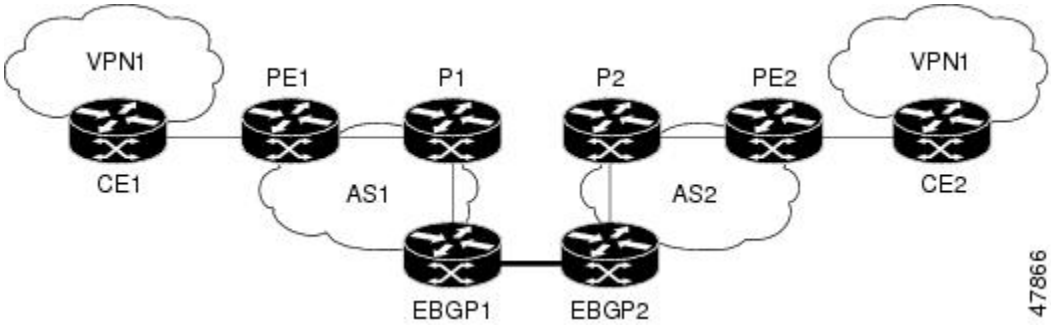
Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Example: Configuring MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

The network topology in the figure below shows two autonomous systems, which are configured as follows:

- Autonomous system 1 (AS1) includes provider edge 1 (PE1), P1, and Exterior Border Gateway Protocol 1(EBGP1). The Interior Gateway Protocol (IGP) is Open Shortest Path First(OSPF).
- Autonomous system 2 (AS2) includes PE2, P2, and EBGP2. The IGP is Intermediate System to Intermediate System (IS-IS).
- Customer edge 1 (CE1) and CE2 belong to the same VPN, which is called VPN1.
- The P devices are route reflectors.
- EBGP1 is configured with the **redistribute connected subnets** command.
- EBGP2 is configured with the **neighbor next-hop-self** command.

Figure 121: Configuring Two Autonomous Systems



Example: Configuration for Autonomous System 1 CE1

The following example shows how to configure CE1 in VPN1 in a topology with two autonomous systems:

```
interface Loopback1
 ip address 10.1.0.4 255.0.0.0
!
interface GigabitEthernet0/0/0
   no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/5/3 point-to-point
  ip address 10.1.0.2 255.0.0.0
 frame-relay interface-dlci 22
!
```

```
router ospf 1
 network 192.168.3.0 255.255.0.0 area 0
```

Example: Configuration for Autonomous System 1 PE1

The following example shows how to configure PE1 in AS1 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface GigabitEthernet0/0/0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/0.3 point-to-point
 ip vrf forwarding V1
 ip address 192.168.2.4 255.255.0.0
 frame-relay interface-dlci 22
!
interface GigabitEthernet0/5/3
 ip address 192.168.3.5 255.255.0.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 192.168.41.0 255.255.0.0 area 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.41.0 255.255.0.0 area 0
!
router bgp 1
 no synchronization
 neighbor 1 peer-group
 neighbor 1 remote-as 1
 neighbor 1 update-source Loopback0
 neighbor 192.168.11.10 peer-group R
 no auto-summary
 !
 address-family ipv4 vrf V1
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 192.168.11.10 peer-group R
  no auto-summary
  exit-address-family
```

Example: Configuration for Autonomous System 1 P1

The following example shows how to configure P1 in AS1 in a topology with two autonomous systems:

```
ip cef
!
interface Loopback0
 ip address 10.1.2.1 255.0.0.0
!
interface GigabitEthernet0/4/7
 ip address 10.1.0.4 255.0.0.0
 tag-switching ip
!
interface GigabitEthernet0/5/3
 ip address 10.2.0.3 255.0.0.0
 duplex auto
 speed auto
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.0.2 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 192.168.3.4 peer-group R
 neighbor 192.168.3.5 peer-group R
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 192.168.3.4 peer-group R
  neighbor 192.168.3.5 peer-group R
  exit-address-family
```

Example: Configuration for Autonomous System 1 EBGP1

The following example shows how to configure EBGP1 in AS1 in a topology with two autonomous systems:

```
ip cef
!
interface Loopback0
 ip address 10.2.2.1 255.0.0.0
!
!
ip cef
!
interface Loopback0
 ip address 10.2.2.1 255.0.0.0
!
interface GigabitEthernetEthernet0/5/3
 ip address 10.1.0.5 255.0.0.0
 tag-switching ip
!
interface GigabitEthernet0/0/0
!
interface GigabitEthernet0/0/0.1 point-to-point
```

```
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.1.0.5 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor 10.1.0.2 remote-as 2
 neighbor 10.1.0.2 peer-group R
 no auto-summary
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 10.1.0.2 activate
  neighbor 10.1.0.2 send-community extended
  neighbor 10.1.0.2 peer-group R
  no auto-summary
  exit-address-family
```

Example: Configuration for Autonomous System 2 EBGP2

The following example shows how to configure EBGP2 in AS2 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 2:103
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.1.1.2 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.1.1.2 255.0.0.0
!
interface GigabitEthernet0/4/7
 no ip address
 encapsulation frame-relay
 load-interval 30
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/3 point-to-point
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
interface GigabitEthernet0/0/4
 no ip address
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
```

```
!
interface GigabitEthernet0/0/4.1 point-to-point
 ip address 10.1.0.5 255.0.0.0
 pvc 1/100
!
router isis
 net 49.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.1.0.1 remote-as 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 update-source Loopback0
 neighbor 10.1.1.2 next-hop-self
!
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 10.1.0.1 activate
  neighbor 10.1.0.1 send-community extended
  neighbor 10.1.1.2 activate
  neighbor 10.1.1.2 next-hop-self
  neighbor 10.1.1.2 send-community extended
  exit-address-family
```

Example: Configuration for Autonomous System 2 P2

The following example shows how to configure P2 in AS2 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 2:108
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.1.0.2 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.1.0.2 255.0.0.0
!
interface GigabitEthernet0/0/0
 ip address 10.2.1.4 255.0.0.0
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/0/3
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/0/3.1 point-to-point
 ip unnumbered Loopback0
```

```
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
router isis
 net aa.0002.0000.0000.0008.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 2
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 10.1.2.1 peer-group R
 neighbor 10.0.1.2 peer-group R
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 10.1.2.1 peer-group R
  neighbor 10.0.1.2 peer-group R
  exit-address-family
```

Example: Configuration for Autonomous System 2 PE2

The following example shows how to configure PE2 in AS2 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 2:109
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 192.168.11.10 255.255.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 192.168.11.10 255.255.0.0
!
interface GigabitEthernet0/5/3
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/5/3.1 point-to-point
 ip vrf forwarding V1
 ip unnumbered Loopback1
 frame-relay interface-dlci 24
!
interface GigabitEthernet0/0/0
```

```
 ip address 192.168.2.10 255.255.0.0
 ip router isis
 tag-switching ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.2.2 255.255.0.0 area 0
!
router isis
 net 49.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 192.168.3.2 remote-as 2
 neighbor 192.168.3.2 update-source Loopback0
 !
 address-family ipv4 vrf V1
  redistribute connected
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.3.2 activate
  neighbor 192.168.3.2 send-community extended
  exit-address-family v
```

Example: Configuration for Autonomous System 2 CE2

The following example shows how to configure CE2 in VPN1 in a topology with two autonomous systems:

```
interface Loopback0
 ip address 192.168.2.2 255.255.0.0
!
interface GigabitEthernet0/0/0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/0.1 point-to-point
 ip unnumbered Loopback0
 frame-relay interface-dlci 24
!
router ospf 1
 network 192.168.4.6 255.255.0.0 area 0
```

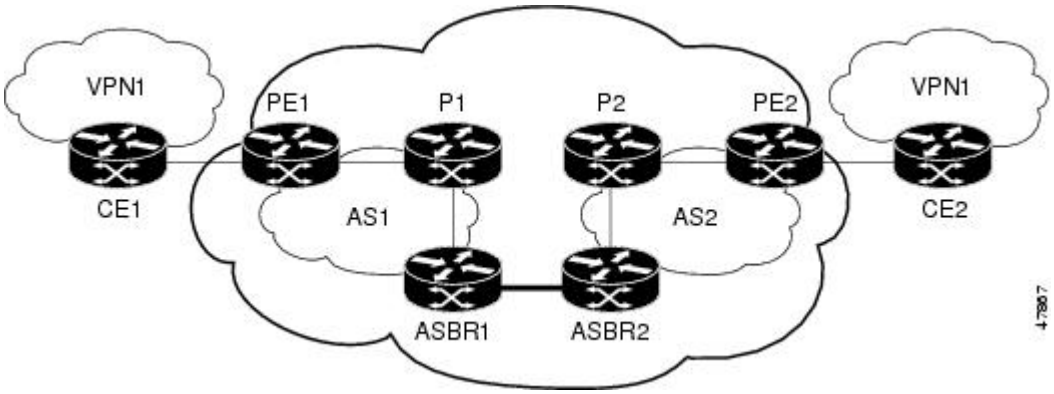
Example: Configuring MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses in a Confederation

The network topology in the figure below shows a single internet service provider, which is partitioning the backbone with confederations. The autonomous system number of the provider is 100. The two autonomous systems run their own IGPs and are configured as follows:

- Autonomous system 1 (AS1) includes provider edge 1 (PE1), P1, Autonomous System Border Router 1 (ASBR1). The Interior Gateway Protocol (IGP) is Open Shortest Path First (OSPF).

- Autonomous system 2 (AS2) includes PE2, P2, ASBR2. The IGP is Intermediate System to Intermediate System (IS-IS).
- Customer edge 1 (CE1) and CE2 belong to the same VPN, which is called VPN1.
- The P devices are route reflectors.
- ASBR1 is configured with the **redistribute connected subnets** command.
- ASBR2 is configured with the **neighbor next-hop-self**command.

Figure 122: Configuring Two Autonomous Systems in a Confederation



Example: Configuration for Autonomous System 1 CE1

The following example shows how to configure CE1 in VPN1 in a confederation topology:

```
interface Loopback1
 ip address 192.168.3.4 255.255.255.255
!
interface GigabitEthernet0/4/7
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/4/7.1 point-to-point
 ip address 192.168.1.3 255.255.0.0
 frame-relay interface-dlci 22
!
router ospf 1
 network 192.168.0.1 255.255.0.0 area 0
```

Example: Configuration for Autonomous System 1 PE1

The following example shows how to configure PE1 in AS1 in a confederation topology:

```
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface GigabitEthernet0/0/0
```

```
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/0.3 point-to-point
 ip vrf forwarding V1
 ip address 10.0.2.4 255.0.0.0
 frame-relay interface-dlci 22
!
interface GigabitEthernet0/4/7
 ip address 10.1.2.6 255.0.0.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.8.4 255.0.0.0 area 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 10.1.8.4 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 bgp confederation identifier 100
 bgp confederation identifier 100
 neighbor 1 peer-group
 neighbor 1 remote-as 1
 neighbor 1 update-source Loopback0
 neighbor 10.2.1.2 peer-group R
 no auto-summary
 !
 address-family ipv4 vrf V1
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 10.2.1.2 peer-group R
  no auto-summary
  exit-address-family
```

Example: Configuration for Autonomous System 1 P1

The following example shows how to configure P1 in AS1 in a confederation topology:

```
ip cef
!
interface Loopback0
 ip address 10.0.0.2 255.0.0.0
!
interface GigabitEthernet0/0/0
 ip address 10.2.1.1 255.0.0.0
 tag-switching ip
!
interface GigabitEthernet0/4/7
 ip address 10.2.2.1 255.0.0.0
 duplex auto
 speed auto
```

```
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.2.2 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 10.0.0.4 peer-group R
 neighbor 10.0.0.5 peer-group R
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 10.1.0.4 peer-group R
  neighbor 10.1.0.5 peer-group R
  exit-address-family
```

Example: Configuration for Autonomous System 1 ASBR1

The following example shows how to configure ASBR1 in AS1 in a confederation topology:

```
ip cef
!
interface Loopback0
 ip address 10.0.0.4 255.0.0.0
!
interface GigabitEthernet0/0/0
 ip address 10.2.1.40 255.255.255.0
 tag-switching ip
!
interface GigabitEthernet0/5/3
 no ip address
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface GigabitEthernet0/5/3.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 pvc 1/100
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.3 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 1
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor 10.0.0.2 remote-as 2
 neighbor 10.0.0.2 next-hop-self
```

```
 neighbor 10.0.0.2 peer-group R
 no auto-summary
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 next-hop-self
  neighbor 10.0.0.2 send-community extended
  neighbor 10.0.0.2 peer-group R
  no auto-summary
  exit-address-family
```

Example: Configuration for Autonomous System 2 ASBR2

The following example shows how to configure ASBR2 in AS2 in a confederation topology:

```
ip cef
!
ip vrf V1
 rd 2:103
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.0.3 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.0.0.3 255.0.0.0
!
interface GigabitEthernet0/4/7
 no ip address
 encapsulation frame-relay
 load-interval 30
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/4/7.2 point-to-point
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
interface GigabitEthernet0/5/3
 no ip address
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface GigabitEthernet0/5/3.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 pvc 1/100
!
router isis
 net aa.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
```

```
 bgp confederation peers 1
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 next-hop-self
 neighbor 10.0.0.8 remote-as 2
 neighbor 10.0.0.8 update-source Loopback0
 neighbor 10.0.0.8 next-hop-self
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 next-hop-self
  neighbor 10.0.0.1 send-community extended
  neighbor 10.0.0.8 activate
  neighbor 10.0.0.8 next-hop-self
  neighbor 10.0.0.8 send-community extended
  exit-address-family
```

Example: Configuration for Autonomous System 2 P2

The following example shows how to configure P2 in AS2 in a confederation topology:

```
ip cef
!
ip vrf V1
 rd 2:108
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.0.8 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.0.0.8 255.0.0.0
!
interface GigabitEthernet0/0/0
 ip address 10.9.1.2 255.0.0.0
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/5/3
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/5/3.1 point-to-point
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
router isis
 net aa.0002.0000.0000.0008.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
```

```
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 2
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 10.0.0.3 peer-group R
 neighbor 10.0.0.9 peer-group R
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 10.0.0.3 peer-group R
  neighbor 10.0.0.9 peer-group R
  exit-address-family
```

Example: Configuration for Autonomous System 2 PE2

The following example shows how to configure PE2 in AS2 in a confederation topology:

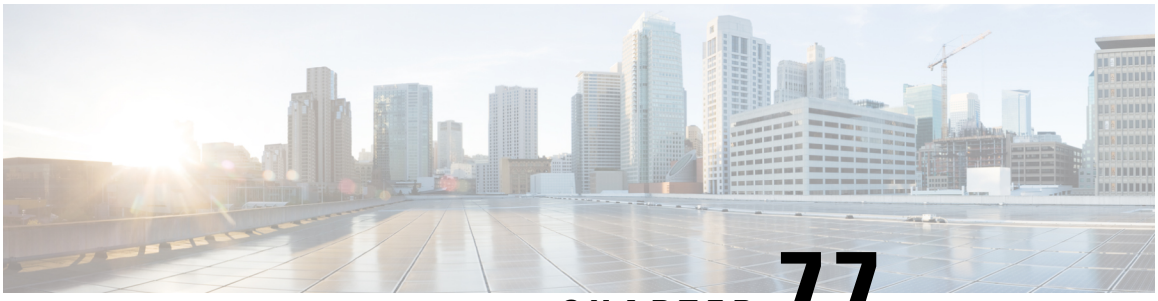
```
ip cef
!
ip vrf V1
 rd 2:109
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.0.9 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.0.0.9 255.0.0.0
!
interface GigabitEthernet0/0/4
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/4.1 point-to-point
 description Bethel
 ip vrf forwarding V1
 ip unnumbered Loopback1
 frame-relay interface-dlci 24
!
interface GigabitEthernet0/4/7
 ip address 10.9.1.1 255.0.0.0
 ip router isis
 tag-switching ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 10.0.0.2 255.0.0.0 area 0
```

```
!
router isis
 net aa.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor 10.0.0.8 remote-as 2
 neighbor 10.0.0.8 update-source Loopback0
 !
 address-family ipv4 vrf V1
  redistribute connected
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.8 activate
  neighbor 10.0.0.8 send-community extended
  exit-address-family
```

Example: Configuration for Autonomous System 2 CE2

The following example shows how to configure CE2 in VPN1 in a confederation topology:

```
interface Loopback0
 ip address 10.0.0.11 255.0.0.0
!
interface GigabitEthernet0/0/7
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/7.1 point-to-point
ip unnumbered Loopback0
 frame-relay interface-dlci 24
!
router ospf 1
 network 10.0.1.2 255.0.0.0 area 0
```



CHAPTER 77

MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

The MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels feature allows a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) to span service providers and autonomous systems. This module explains how to configure an MPLS VPN Inter-AS network so that the Autonomous System Boundary Routers (ASBRs) exchange IPv4 routes with MPLS labels of the provider edge (PE) routers. Route reflectors (RRs) exchange VPN-IPv4 routes by using multihop, multiprotocol, external Border Gateway Protocol (eBGP).

Prerequisites for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

The network must be properly configured for MPLS VPN operation before you configure MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels.

The table below lists the Cisco 12000 series line card support in Cisco IOS S releases.

Table 147: Cisco 12000 Series Line Card Support in Cisco IOS S Releases

Type	Line Cards	Cisco IOS Release Supported
ATM	4-Port OC-3 ATM 1-Port OC-12 ATM 4-Port OC-12 ATM 8-Port OC-3 ATM	12.0(22)S 12.0(23)S 12.0(27)S
Channelized interface	2-Port CHOC-3 6-Port Ch T3 (DS1) 1-Port CHOC-12 (DS3) 1-Port CHOC-12 (OC-3) 4-Port CHOC-12 ISE 1-Port CHOC-48 ISE	12.0(22)S 12.0(23)S 12.0(27)S
Electrical interface	6-Port DS3 12-Port DS3 6-Port E3 12-Port E3	12.0(22)S 12.0(23)S 12.0(27)S
Ethernet	3-Port GbE	12.0(23)S 12.0(27)S
Packet over SONET (POS)	4-Port OC-3 POS 8-Port OC-3 POS 16-Port OC-3 POS 1-Port OC-12 POS 4-Port OC-12 POS 1-Port OC-48 POS 4-Port OC-3 POS ISE 8-Port OC-3 POS ISE 16-Port OC-3 POS ISE 4-Port OC-12 POS ISE 1-Port OC-48 POS ISE	12.0(22)S 12.0(23)S 12.0(27)S

Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

- For networks configured with eBGP multihop, you must configure a label switched path (LSP) between nonadjacent routers.
- The physical interfaces that connect the BGP speakers must support Cisco Express Forwarding or distributed Cisco Express Forwarding and MPLS.

Information About MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

MPLS VPN Inter-AS Introduction

An autonomous system is a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. Also, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless to the customer.

Benefits of MPLS VPN Inter-AS

An MultiprotocolLabel Switching (MPLS) VPN Inter-AS provides the following benefits:

- Allows a VPN to cross more than one service provider backbone: Service providers running separate autonomous systems can jointly offer MPLS VPN services to the same customer. A VPN can begin at one customer site and traverse different VPN service provider backbones before arriving at another site of the same customer. Previously, MPLS VPN could traverse only a single Border Gateway Protocol (BGP) autonomous system service provider backbone. This feature allows multiple autonomous systems to form a continuous (and seamless) network between customer sites of a service provider.
- Allows a VPN to exist in different areas: A service provider can create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.
- Allows confederations to optimize Internal Border Gateway Protocol (IBGP) meshing: IBGP meshing in an autonomous system is more organized and manageable. An autonomous system can be divided into multiple, separate subautonomous systems and then classify them into a single confederation (even though the entire VPN backbone appears as a single autonomous system). This capability allows a service provider to offer MPLS VPNs across the confederation because it supports the exchange of labeled VPN-IPv4 Network Layer Reachability Information (NLRI) between the subautonomous systems that form the confederation.

Information About Using MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

This feature can configure a MPLS VPN Inter-AS network so that the ASBRs exchange IPv4 routes with MPLS labels of the PE routers. RRs exchange VPN-IPv4 routes by using multihop, multiprotocol, External Border Gateway Protocol (eBGP). This method of configuring the Inter-AS system is often called MPLS VPN Inter-AS--IPv4 BGP Label Distribution.

Benefits of MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

An Inter-AS system can be configured so that the ASBRs exchange the IPv4 routes and MPLS labels has the following benefits:

- Saves the ASBRs from having to store all the VPN-IPv4 routes. Using the route reflectors to store the VPN-IPv4 routes and forward them to the PE routers results in improved scalability compared wtih configurations where the ASBR holds all of the VPN-IPv4 routes and forwards the routes based on VPN-IPv4 labels.
- Simplifies the configuration at the border of the network by having the route reflectors hold the VPN-IPv4 routes.
- Enables a non-VPN core network to act as a transit network for VPN traffic. You can transport IPv4 routes with MPLS labels over a non-MPLS VPN service provider.
- Eliminates the need for any other label distribution protocol between adjacent LSRs. If two adjacent label switch routers (LSRs) are also BGP peers, BGP can handle the distribution of the MPLS labels. No other label distribution protocol is needed between the two LSRs.

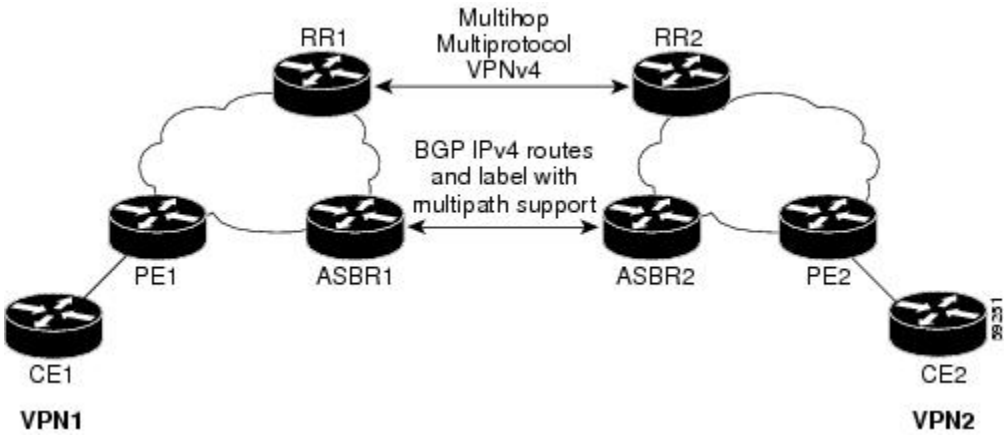
How the Inter-AS Works When ASBRs Exchange IPv4 Routes with MPLS Labels

A VPN service provider network to exchange IPv4 routes with MPLS labels can be configured. The VPN service provider network can be configured as follows:

- Route reflectors exchange VPN-IPv4 routes by using multihop, multiprotocol eBGP. This configuration also preserves the next-hop information and the VPN labels across the autonomous systems.
- A local PE router (for example, PE1 in the figure below) needs to know the routes and label information for the remote PE router (PE2). This information can be exchanged between the PE routers and ASBRs in one of two ways:
  - Internal Gateway Protocol (IGP) and Label Distribution Protocol (LDP): The ASBR can redistribute the IPv4 routes and MPLS labels it learned from eBGP into IGP and LDP and vice versa.
  - Internal Border Gateway Protocol (iBGP) IPv4 label distribution:The ASBR and PE router can use direct iBGP sessions to exchange VPN-IPv4 and IPv4 routes and MPLS labels.

Alternatively, the route reflector can reflect the IPv4 routes and MPLS labels learned from the ASBR to the PE routers in the VPN. This is accomplished by the ASBR exchanging IPv4 routes and MPLS labels with the route reflector. The route reflector also reflects the VPN-IPv4 routes to the PE routers in the VPN. For example,

in VPN1 of the figure below, RR1 reflects to PE1 the VPN-IPv4 routes it learned and IPv4 routes and MPLS labels learned from ASBR1. Using the route reflectors to store the VPN-IPv4 routes and forward them through the PE routers and ASBRs allows for a scalable configuration.



BGP Routing Information

BGP routing information includes the following items:

- A network number (prefix), which is the IP address of the destination.
- Autonomous system path, which is a list of the other autonomous systems through which a route passes on its way to the local router. The first autonomous system in the list is closest to the local router; the last autonomous system in the list is farthest from the local router and usually the autonomous system where the route began.
- Path attributes, which provide other information about the autonomous system path, for example, the next hop.

Types of BGP Messages and MPLS Labels

MPLS labels are included in the update messages that a router sends. Routers exchange the following types of BGP messages:

- Keepalive messages--Routers exchange keepalive messages to determine if a neighboring router is still available to exchange routing information. The router sends these messages at regular intervals. (Sixty seconds is the default for Cisco routers.) The keepalive message does not contain routing data; it contains only a message header.
- Notification messages--When a router detects an error, it sends a notification message.
- Open messages--After a router establishes a TCP connection with a neighboring router, the routers exchange open messages. This message contains the number of the autonomous system to which the router belongs and the IP address of the router that sent the message.
- Update messages--When a router has a new, changed, or broken route, it sends an update message to the neighboring router. This message contains the NLRI, which lists the IP addresses of the usable routes. The update message includes any routes that are no longer usable. The update message also includes path attributes and the lengths of both the usable and unusable paths. Labels for VPN-IPv4 routes are encoded in the update message as specified in RFC 2858. The labels for the IPv4 routes are encoded in the update message as specified in RFC 3107.

How BGP Sends MPLS Labels with Routes

When BGP (eBGP and iBGP) distributes a route, it can also distribute an MPLS label that is mapped to that route. The MPLS label mapping information for the route is carried in the BGP update message that contains the information about the route. If the next hop is not changed, the label is preserved.

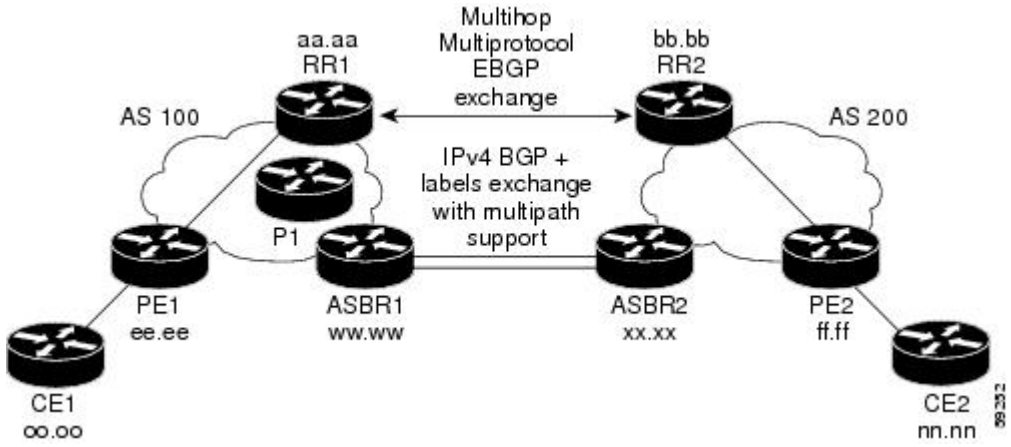
When you issue the **neighbor send-label** command on both BPG routers, the routers advertise to each other that they can then send MPLS labels with the routes. If the routers successfully negotiate their ability to send MPLS labels, the routers add MPLS labels to all outgoing BGP updates.

How to Configure MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

To configure MPLS VPN Inter-AS with ASBRs exchanging IPv4 routes and MPLS labels, perform the tasks in the following sections:

The figure below shows the following sample configuration:

- The configuration consists of two VPNs.
- The ASBRs exchange the IPv4 routes with MPLS labels.
- The route reflectors exchange the VPN-IPv4 routes using multihop MPLS eBGP.
- The route reflectors reflect the IPv4 and VPN-IPv4 routes to the other routers in their autonomous system.



Configuring the ASBRs to Exchange IPv4 Routes and MPLS Labels

Perform this task to configure the ASBRs to exchange IPv4 routes and MPLS labels. This configuration procedure uses ASBR1 as an example.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*

4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv4** [**multicast** | **unicast** | **mdt** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor***ip-address* **send-label**
8. **exit-address-family**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor hh.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **mdt** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes.

	Command or Action	Purpose
		• The **mdt** keyword specifies an IPv4 multicast distribution tree (MDT) address family session. • The **vrf** *vrf-name* keyword and argument specify the name of the VPN routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor hh.0.0.1 activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor***ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor hh.0.0.1 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the Route Reflectors to Exchange VPN-IPv4 Routes

Perform this task to enable the route reflectors to exchange VPN-IPv4 routes by using multihop, multiprotocol eBGP.

This procedure also specifies that the next hop information and the VPN label are to be preserved across the autonomous systems. This procedure uses RR1 as an example of the route reflector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
6. **address-family vpnv4** [**unicast**]

7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name*} **next-hop unchanged**
9. **exit-address-family**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. The autonomous system number identifies RR1 to routers in other autonomous systems.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor bb.bb.bb.bb remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `Router(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.

	Command or Action	Purpose
		• The *ttl* argument specifies the time-to-live in the range from 1 to 255 hops.
Step 6	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor bb.bb.bb.bb activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **next-hop unchanged** **Example:** `Router(config-router-af)# neighbor ip-address next-hop unchanged`	Enables an eBGP multihop peer to propagate the next hop unchanged. • The *ip-address* argument specifies the IP address of the next hop. • The *peer-group-name* argument specifies the name of a BGP peer group that is the next hop.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the Route Reflector to Reflect Remote Routes in Its Autonomous System

Perform this task to enable the RR to reflect the IPv4 routes and labels learned by the ASBR to the PE routers in the autonomous system.

This is accomplished by making the ASBR and PE router route reflector clients of the RR. This procedure also explains how to enable the RR to reflect the VPN-IPv4 routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name* **activate**
6. **neighbor** *ip-address* **route-reflector-client**
7. **neighbor** *ip-address* **send-label**
8. **exit-address-family**
9. **address-family vpnv4** [**unicast**]
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**
11. **neighbor** *ip-address* **route-reflector-client**
12. **exit-address-family**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard IPv4 address prefixes. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes.

	Command or Action	Purpose
		• The **vrf** *vrf-name* keyword and argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name* **activate** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 6	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ees route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client. • The *ip-address* argument specifies the IP address of the BGP neighbor being configured as a client.
Step 7	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 11	**neighbor** *ip-address* **route-reflector-client** **Example:**	Enables the RR to pass iBGP routes to the neighboring router.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor ee.ee.ee.ee route-reflector-client`	

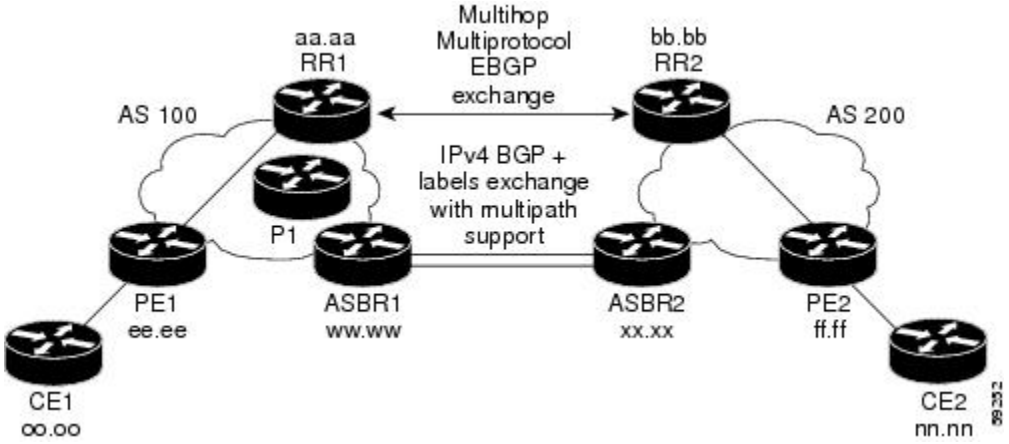
Step 12	**exit-address-family**
	Example:
	`Router(config-router-af)#`
	`exit-address-family`
		Exits address family configuration mode.

Step 13	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Verifying the MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels Configuration

If you use ASBRs to distribute the IPv4 labels and route reflectors to distribute the VPN-IPv4 routes, use the following procedures to help verify the configuration:

The figure below shows the configuration that is referred to in the next several sections.



Verifying the Route Reflector Configuration

Perform this task to verify the route reflector configuration.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name* } [**summary**] [**labels**]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name* } [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all summary`	(Optional) Displays VPN address information from the BGP table. • Use the **all** and **summary** keywords to verify that a multihop, multiprotocol eBGP session exists between the route reflectors and that the VPNv4 routes are being exchanged between the route reflectors. The last two lines of the command output show the following information: • • Prefixes are being learned from PE1 and then passed to RR2. • Prefixes are being learned from RR2 and then passed to PE1. • Use the **all** and **labels** keywords to verify that the route reflectors exchange VPNv4 label information.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying that CE1 Can Communicate with CE2

Perform this task to verify that router CE1 has NLRI for router CE2.

SUMMARY STEPS

1. **enable**
2. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | [**protocol** [*protocol-id*]] | [**list** [*access-list-number* | *access-list-name*]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip route** [*ip-address* [*mask*] [**longer-prefixes**]]	[**protocol** [*protocol-id*]]	[**list** [*access-list-number*	*access-list-name*] **Example:** `Router# show ip route nn.nn.nn.nn`	Displays the current state of the routing table. • Use the *ip-address* argument to verify that CE1 has a route to CE2. • Use this command to verify the routes learned by CE1. Make sure that the route for CE2 is listed.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to privileged EXEC mode.

Verifying that PE1 Can Communicate with CE2

Perform this task to verify that router PE1 has NLRI for router CE2.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [[]] [**summary***output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]
3. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix* | *length* [**longer-prefixes**] [*output-modifiers*]]] [**network-address** *mask*]] **longer-prefixes** [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]
4. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
5. **show mpls forwarding-table** [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
6. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]
7. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
8. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [[]] [**summary***output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]	(Optional) Displays the IP routing table associated with a VRF. • Use this command to verify that router PE1 learns routes from router CE2 (nn.nn.nn.nn).

	Command or Action	Purpose
	Example: `Router# show ip route vrf vpn1 nn.nn.nn.nn`	

Step 3	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix* | *length* [**longer-prefixes**] [*output-modifiers*]]] [**network-address** *mask*]] **longer-prefixes** [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]
	Example:
	`Router# show ip bgp vpnv4 vrf vpn1 nn.nn.nn.nn`
	Example:
	`Router# show ip bgp vpnv4 all nn.nn.nn.nn`
		(Optional) Displays VPN address information from the BGP table.
		• Use the **vrf** or **all** keyword to verify that router PE2 is the BGP next-hop to router CE2.

Step 4	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef vrf vpn1 nn.nn.nn.nn`	(Optional) Displays entries in the Forwarding Information Base (FIB) or displays a summary of the FIB. • Use this command to verify that the Cisco Express Forwarding entries are correct.
Step 5	**show mpls forwarding-table** [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table`	(Optional) Displays the contents of the MPLS LFIB. • Use this command to verify the IGP label for the BGP next hop router (autonomous system boundary).
Step 6	**show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**] **Example:** `Router# show ip bgp ff.ff.ff.ff`	(Optional) Displays entries in the BGP routing table. • Use the **show ip bgp** command to verify the label for the remote egress PE router (PE2).
Step 7	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all labels`	(Optional) Displays VPN address information from the BGP table. • Use the **all** and **summary** keywords to verify the VPN label of CE2, as advertised by PE2.
Step 8	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying that PE2 Can Communicate with CE2

Perform this task to ensure that PE2 can access CE2.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary**[*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]
3. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
4. **show ip bgp vpnv4** { **all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
5. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
6. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary**[*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]] **Example:** `Router# show ip route vrf vpn1 nn.nn.nn.nn`	(Optional) Displays the IP routing table associated with a VRF. • Use this command to check the VPN routing and forwarding table for CE2. The output provides next-hop information.
Step 3	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn1 nn.nn.nn.nn`	(Optional) Displays the contents of the LFIB. • Use the **vrf** keyword to check the VPN routing and forwarding table for CE2. The output provides the label for CE2 and the outgoing interface.
Step 4	**show ip bgp vpnv4** { **all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all labels`	(Optional) Displays VPN address information from the BGP table. • Use the **all** and **labels** keywords to check the VPN label for CE2 in the multiprotocol BGP table.
Step 5	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]	(Optional) Displays entries in the FIB or displays a summary of the FIB.

	Command or Action	Purpose
	Example: `Router# show ip cef vpn1 nn.nn.nn.nn`	• Use this command to check the Cisco Express Forwarding entry for CE2. The command output shows the local label for CE2 and the outgoing interface.
Step 6	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying the ASBR Configuration

Perform this task to verify that the ASBRs exchange IPv4 routes with MPLS labels or IPv4 routes without labels as prescribed by a route map.

Verifying the ASBR Configuration

SUMMARY STEPS

1. **enable**
2. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]
3. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
4. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]
	Example:
	`Router# show ip bgp ff.ff.ff.ff`
		(Optional) Displays entries in the BGP routing table.
		• Use this command to check that:
		  • ASBR1 receives an MPLS label for PE2 from ASBR2.
		  • ASBR1 receives IPv4 routes for RR2 without labels from ASBR2.
		  • ASBR2 distributes an MPLS label for PE2 to ASBR1.
		  • ASBR2 does not distribute a label for RR2 to ASBR1.

Step 3	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:**	(Optional) Displays entries in the FIB or displays a summary of the FIB.

	Command or Action	Purpose

	`Router# show ip cef ff.ff.ff.ff`
	Example:
	`Router# show ip cef bb.bb.bb.bb`
		• Use thiscommand from ASBR1 and ASBR2 to check that:
		• The Cisco Express Forwarding entry for PE2 is correct.
		• The Cisco Express Forwarding entry for RR2 is correct.

Step 4	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

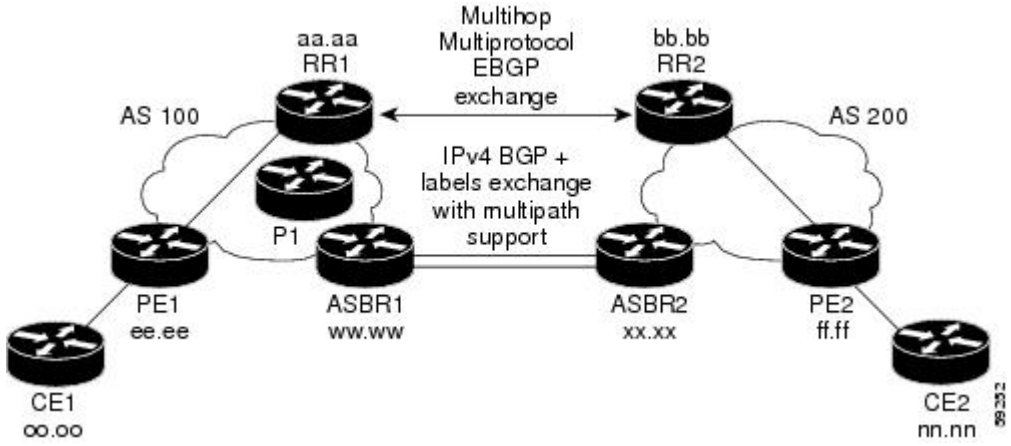
Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over an MPLS VPN Service Provider Examples

Configuration examples for Inter-AS using BGP to distribute routes and MPLS labels over an MPLS VPN service provider included in this section are as follows:

The figure below shows two MPLS VPN service providers. The service provider distributes the VPN-IPv4 routes between the route reflectors. The MPLS VPN service providers distribute the IPv4 routes with MPLS labels between the ASBRs.

The configuration example shows the following two techniques you can use to distribute the VPN-IPv4 routes and the IPv4 routes with MPLS labels of the remote RRs and PEs to the local RRs and PEs:

- Autonomous system 100 uses the RRs to distribute the VPN-IPv4 routes learned from the remote RRs. The RRs also distribute the remote PE address and label learned from ASBR1 using IPv4 labels.
- In Autonomous system 200, the IPv4 routes that ASBR2 learned are redistributed into IGP.



Route Reflector 1 Configuration Example (MPLS VPN Service Provider)

The configuration example for RR1 specifies the following:

- RR1 exchanges VPN-IPv4 routes with RR2 using multiprotocol, multihop eBGP.
- The VPN-IPv4 next-hop information and the VPN label are preserved across the autonomous systems.
- RR1 reflects to PE1:
  - The VPN-IPv4 routes learned from RR2
  - The IPv4 routes and MPLS labels learned from ASBR1

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
!
interface Ethernet0/3
 ip address dd.0.0.2 255.0.0.0
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
```

```
 neighbor ee.ee.ee.ee route-reflector-client                !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client                !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client                !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                    !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                 !with next hop unchanged

 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
snmp-server engineID local 00000009020000D0584B25C0
snmp-server community public RO
snmp-server community write RW
no snmp-server ifindex persist
snmp-server packetsize 2048
!
end
```

ASBR1 Configuration Example (MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

In this example, ASBR1 uses route maps to filter routes:

- A route map called OUT specifies that ASBR1 should distribute the PE1 route (ee.ee) with labels and the RR1 route (aa.aa) without labels.
- A route map called IN specifies that ASBR1 should accept the PE2 route (ff.ff) with labels and the RR2 route (bb.bb) without labels.

```
ip subnet-zero
mpls label protocol ldp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
!
interface Ethernet0/2
 ip address hh.0.0.2 255.0.0.0
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet0/2
```

```
 network ww.ww.ww.ww 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100

router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor hh.0.0.1 remote-as 200
 no auto-summary
 !
!
address-family ipv4                        ! Redistributing IGP into BGP
 redistribute ospf 10                      ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa activate             ! get into the BGP table
 neighbor aa.aa.aa.aa send-label
 neighbor hh.0.0.1 activate
 neighbor hh.0.0.1 advertisement-interval 5
 neighbor hh.0.0.1 send-label
 neighbor hh.0.0.1 route-map IN in         ! accepting routes in route map IN.
 neighbor hh.0.0.1 route-map OUT out       ! distributing routes in route map OUT.
 neighbor kk.0.0.1 activate
 neighbor kk.0.0.1 advertisement-interval 5
 neighbor kk.0.0.1 send-label
 neighbor kk.0.0.1 route-map IN in         ! accepting routes in route map IN.
 neighbor kk.0.0.1 route-map OUT out       ! distributing routes in route map OUT.
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ee.ee.ee.ee log                !Setting up the access lists
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log
route-map IN permit 10                              !Setting up the route maps
 match ip address 2
 match mpls-label
!
route-map IN permit 11
 match ip address 4
!
route-map OUT permit 12
 match ip address 3
!
route-map OUT permit 13
 match ip address 1
 set mpls-label
!
end
```

Route Reflector 2 Configuration Example (MPLS VPN Service Provider)

RR2 exchanges VPN-IPv4 routes with RR1 through multihop, multiprotocol eBGP. This configuration also specifies that the next-hop information and the VPN label are preserved across the autonomous systems:

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
```

```
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa ebgp-multihop 255
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor ff.ff.ff.ff remote-as 200
 neighbor ff.ff.ff.ff update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor aa.aa.aa.aa activate
 neighbor aa.aa.aa.aa next-hop-unchanged              !Multihop VPNv4 session with RR1
 neighbor aa.aa.aa.aa send-community extended              !with next-hop-unchanged
 neighbor ff.ff.ff.ff activate
 neighbor ff.ff.ff.ff route-reflector-client          !VPNv4 session with PE2
 neighbor ff.ff.ff.ff send-community extended
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
end
```

ASBR2 Configuration Example (MPLS VPN Service Provider)

ASBR2 exchanges IPv4 routes and MPLS labels with ASBR1. However, in contrast to ASBR1, ASBR2 does not use the RR to reflect IPv4 routes and MPLS labels to PE2. ASBR2 redistributes the IPv4 routes and MPLS labels learned from ASBR1 into IGP. PE2 can now reach these prefixes.

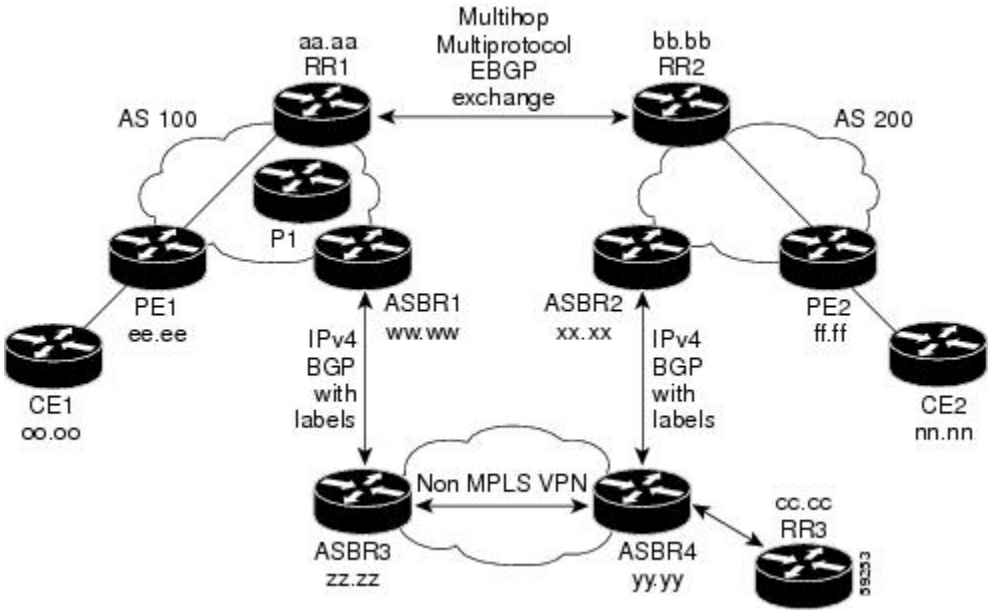
```
ip subnet-zero
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
!
interface Ethernet1/0
 ip address hh.0.0.1 255.0.0.0
!
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets           ! Redistributing the routes learned from
 passive-interface Ethernet1/0               ! ASBR1(eBGP+labels session) into IGP
```

```
 network xx.xx.xx.xx 0.0.0.0 area 200         ! so that PE2 will learn them
 network jj..0.0 0.255.255.255 area 200
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor hh.0.0.2 remote-as 100
 no auto-summary
 !
address-family ipv4
 redistribute ospf 20                         ! Redistributing IGP into BGP
 neighbor hh.0.0.2 activate                   ! so that PE2 & RR2 loopbacks
 neighbor hh.0.0.2 advertisement-interval 5   ! will get into the BGP-4 table.
 neighbor hh.0.0.2 route-map IN in
 neighbor hh.0.0.2 route-map OUT out
 neighbor hh.0.0.2 send-label
 neighbor kk.0.0.2 activate
 neighbor kk.0.0.2 advertisement-interval 5
 neighbor kk.0.0.2 route-map IN in
 neighbor kk.0.0.2 route-map OUT out
 neighbor kk.0.0.2 send-label
 no auto-summary
 no synchronization
 exit-address-family
!
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ff.ff.ff.ff log          !Setting up the access lists
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
route-map IN permit 11                       !Setting up the route maps
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
end
```

Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over a Non-MPLS VPN Service Provider Examples

Configuration examples for Inter-AS using BGP to distribute routes and MPLS labels over a non MPLS VPN service provider included in this section are as follows:

The figure below shows two MPLS VPN service providers that are connected through a non MPLS VPN service provider. The autonomous system in the middle of the network is configured as a backbone autonomous system that uses LDP or Tag Distribution Protocol (TDP) to distribute MPLS labels. Traffic engineering tunnels can also be used instead of TDP or LDP to build the LSP across the non MPLS VPN service provider.



Route Reflector 1 Configuration Example (Non-MPLS VPN Service Provider)

The configuration example for RR1 specifies the following:

- RR1 exchanges VPN-IPv4 routes with RR2 using multiprotocol, multihop eBGP.
- The VPN-IPv4 next-hop information and the VPN label are preserved across the autonomous systems.
- RR1 reflects to PE1:
  - The VPN-IPv4 routes learned from RR2
  - The IPv4 routes and MPLS labels learned from ASBR1

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
!
interface Serial1/2
 ip address dd.0.0.2 255.0.0.0
 clockrate 124061
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
```

```
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client               !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client               !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client              !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                   !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                with next-hop-unchanged
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
snmp-server engineID local 00000009020000D0584B25C0
snmp-server community public RO
snmp-server community write RW
no snmp-server ifindex persist
snmp-server packetsize 2048
!
end
```

ASBR1 Configuration Example (Non-MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

In this example, ASBR1 uses route maps to filter routes:

- A route map called OUT specifies that ASBR1 should distribute the PE1 route (ee.ee) with labels and the RR1 route (aa.aa) without labels.
- A route map called IN specifies that ASBR1 should accept the PE2 route (ff.ff) with labels and the RR2 route (bb.bb) without labels.

```
ip subnet-zero
ip cef distributed
mpls label protocol ldp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
!
```

```
interface Serial3/0/0
 ip address kk.0.0.2 255.0.0.0
 ip route-cache distributed
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Serial3/0/0
 network ww.ww.ww.ww 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100

router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor kk.0.0.1 remote-as 200
 no auto-summary
!
 address-family ipv4
 redistribute ospf 10                      ! Redistributing IGP into BGP
 neighbor aa.aa.aa.aa activate             ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa send-label           ! get into BGP table
 neighbor kk.0.0.1 activate
 neighbor kk.0.0.1 advertisement-interval 5
 neighbor kk.0.0.1 send-label
 neighbor kk.0.0.1 route-map IN in    ! Accepting routes specified in route map IN
 neighbor kk.0.0.1 route-map OUT out  ! Distributing routes specified in route map OUT
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ee.ee.ee.ee log
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log
!
route-map IN permit 10
 match ip address 2
 match mpls-label
!
route-map IN permit 11
 match ip address 4
!
route-map OUT permit 12
 match ip address 3
!
route-map OUT permit 13
 match ip address 1
 set mpls-label
!
end
```

Route Reflector 2 Configuration Example (Non-MPLS VPN Service Provider)

RR2 exchanges VPN-IPv4 routes with RR1 using multihop, multiprotocol eBGP. This configuration also specifies that the next-hop information and the VPN label are preserved across the autonomous systems:

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa ebgp-multihop 255
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor ff.ff.ff.ff remote-as 200
 neighbor ff.ff.ff.ff update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor aa.aa.aa.aa activate
 neighbor aa.aa.aa.aa next-hop-unchanged              !MH vpnv4 session with RR1
 neighbor aa.aa.aa.aa send-community extended              !with next-hop-unchanged
 neighbor ff.ff.ff.ff activate
 neighbor ff.ff.ff.ff route-reflector-client          !vpnv4 session with PE2
 neighbor ff.ff.ff.ff send-community extended
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
end
```

ASBR2 Configuration Example (Non-MPLS VPN Service Provider)

ASBR2 exchanges IPv4 routes and MPLS labels with ASBR1. However, in contrast to ASBR1, ASBR2 does not use the RR to reflect IPv4 routes and MPLS labels to PE2. ASBR2 redistributes the IPv4 routes and MPLS labels learned from ASBR1 into IGP. PE2 can now reach these prefixes.

```
ip subnet-zero
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
!
interface Ethernet0/1
 ip address qq.0.0.2 255.0.0.0
!
```

```
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets            !redistributing the routes learned from
 passive-interface Ethernet0/1                !ASBR2 (eBGP+labels session) into IGP
 network xx.xx.xx.xx 0.0.0.0 area 200         !so that PE2 will learn them
 network jj.0.0.0 0.255.255.255 area 200
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor qq.0.0.1 remote-as 100
 no auto-summary
!
address-family ipv4                          ! Redistributing IGP into BGP
                                                              redistribute ospf 20
                    ! so that PE2 & RR2 loopbacks
 neighbor qq.0.0.1 activate                  !  will get into the BGP-4 table
 neighbor qq.0.0.1 advertisement-interval 5
 neighbor qq.0.0.1 route-map IN in
 neighbor qq.0.0.1 route-map OUT out
 neighbor qq.0.0.1 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 11
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
!
end
```

ASBR3 Configuration Example (Non-MPLS VPN Service Provider)

ASBR3 belongs to a non MPLS VPN service provider. ASBR3 exchanges IPv4 routes and MPLS labels with ASBR1. ASBR3 also passes the routes learned from ASBR1 to ASBR4 through RR3.

Note Do not redistribute eBGP routes learned into iBGP if you are using iBGP to distribute the routes and labels. This is not a supported configuration.

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address yy.yy.yy.yy 255.255.255.255
interface Hssi4/0
 ip address mm.0.0.0.1 255.0.0.0
 mpls ip
 hssi internal-clock
 !
interface Serial5/0
 ip address kk.0.0.1 255.0.0.0
 load-interval 30
 clockrate 124061
!
router ospf 30
log-adjacency-changes
auto-cost reference-bandwidth 1000
redistribute connected subnets
network yy.yy.yy.yy 0.0.0.0 area 300
network mm.0.0.0 0.255.255.255 area 300
!
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
 neighbor kk.0.0.2 remote-as 100
 no auto-summary
 !
 address-family ipv4
 neighbor cc.cc.cc.cc activate            ! iBGP+labels session with RR3
 neighbor cc.cc.cc.cc send-label
 neighbor kk.0.0.2 activate               ! eBGP+labels session with ASBR1
 neighbor kk.0.0.2 advertisement-interval 5
 neighbor kk.0.0.2 send-label
 neighbor kk.0.0.2 route-map IN in
 neighbor kk.0.0.2 route-map OUT out
 no auto-summary
 no synchronization
 exit-address-family
!
ip classless
!
access-list 1 permit ee.ee.ee.ee log
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log
!
route-map IN permit 10
 match ip address 1
```

```
  match mpls-label
!
route-map IN permit 11
   match ip address 3
!
route-map OUT permit 12
 match ip address 2
  set mpls-label
!
route-map OUT permit 13
   match ip address 4
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

Route Reflector 3 Configuration Example (Non-MPLS VPN Service Provider)

RR3 is a non MPLS VPN RR that reflects IPv4 routes with MPLS labels to ASBR3 and ASBR4.

```
ip subnet-zero
mpls label protocol ldp
mpls traffic-eng auto-bw timers
no mpls ip
!
interface Loopback0
 ip address cc.cc.cc.cc 255.255.255.255
!
interface POS0/2
 ip address pp.0.0.1 255.0.0.0
 crc 16
 clock source internal
!
router ospf 30
 log-adjacency-changes
 network cc.cc.cc.cc 0.0.0.0 area 300
 network pp.0.0.0 0.255.255.255 area 300
!
router bgp 300
 bgp log-neighbor-changes
 neighbor zz.zz.zz.zz remote-as 300
 neighbor zz.zz.zz.zz update-source Loopback0
 neighbor yy.yy.yy.yy remote-as 300
 neighbor yy.yy.yy.yy update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor zz.zz.zz.zz activate
 neighbor zz.zz.zz.zz route-reflector-client
 neighbor zz.zz.zz.zz send-label               ! iBGP+labels session with ASBR3
 neighbor yy.yy.yy.yy activate
 neighbor yy.yy.yy.yy route-reflector-client
 neighbor yy.yy.yy.yy send-label               ! iBGP+labels session with ASBR4
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

ASBR4 Configuration Example (Non-MPLS VPN Service Provider)

ASBR4 belongs to a non MPLS VPN service provider. ASBR4 and ASBR3 exchange IPv4 routes and MPLS labels by means of RR3.

Note Do not redistribute eBGP routes learned into iBGP if you are using iBGP to distribute the routes and labels. This is not a supported configuration.

```
ip subnet-zero
ip cef distributed
!
interface Loopback0
 ip address zz.zz.zz.zz 255.255.255.255
!
interface Ethernet0/2
 ip address qq.0.0.1 255.0.0.0
!
interface POS1/1/0
 ip address pp.0.0.2 255.0.0.0
 ip route-cache distributed
 !
interface Hssi2/1/1
 ip address mm.0.0.2 255.0.0.0
 ip route-cache distributed
 mpls label protocol ldp
 mpls ip
 hssi internal-clock
!
router ospf 30
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet0/2
 network zz.zz.zz.zz 0.0.0.0 area 300
 network pp.0.0.0 0.255.255.255 area 300
 network mm.0.0.0 0.255.255.255 area 300
 !
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
 neighbor qq.0.0.2 remote-as 200
 no auto-summary
 !
 address-family ipv4
 neighbor cc.cc.cc.cc activate
 neighbor cc.cc.cc.cc send-label
 neighbor qq.0.0.2 activate
 neighbor qq.0.0.2 advertisement-interval 5
 neighbor qq.0.0.2 send-label
 neighbor qq.0.0.2 route-map IN in
 neighbor qq.0.0.2 route-map OUT out
 no auto-summary
 no synchronization
 exit-address-family
!
ip classless
!
```

```
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 10
 match ip address 1
  match mpls-label
!
route-map IN permit 11
   match ip address 3
!
route-map OUT permit 12
 match ip address 2
  set mpls-label
!
route-map OUT permit 13
   match ip address 4
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

Additional References

Related Documents

Related Topic	Document Title
MPLS	MPLS Product Literature

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1700	*Assigned Numbers*

RFC	Title
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh IBGP*
RFC 2842	*Capabilities Advertisement with BGP-4*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3107	*Carrying Label Information in BGP-4*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

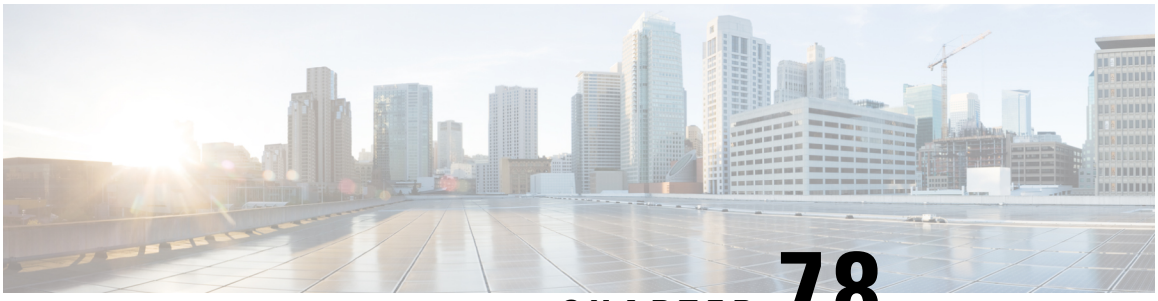
Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 148: Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

Feature Name	Releases	Feature Configuration Information
MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels	12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(27)S 12.0(29)S Cisco IOS XE Release 2.5	This module explains how to configure an MPLS VPN Inter-AS network so that the ASBRs exchange IPv4 routes with MPLS labels of the provider edge (PE) routers. Route reflectors (RRs) exchange VPN-IPv4 routes by using multihop, multiprotocol, external Border Gateway Protocol (eBGP). In Cisco IOS XE Release 2.5, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.



CHAPTER 78

MPLS VPN--Inter-AS Option AB

The MPLS VPN--Inter-AS Option AB feature combines the best functionality of an Inter-AS Option (10) A and Inter-AS Option (10) B network to allow a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) service provider to interconnect different autonomous systems to provide VPN services. These networks are defined in RFC 4364 section 10 "Multi-AS Backbones," subsections a and b, respectively.

When different autonomous systems are interconnected in an MPLS VPN--Inter-AS Option AB configuration, the entire network configuration is scaled and simplified, and maintains IP quality of service (QoS) functions between Autonomous System Boundary Router (ASBR) peers.

In an Inter-AS Option A network, ASBR peers are connected by multiple subinterfaces with at least one interface VPN that spans the two autonomous systems. These ASBRs associate each subinterface with a VPN routing and forwarding (VRF) instance and a Border Gateway Protocol (BGP) session to signal unlabeled IP prefixes. As a result, traffic between the back-to-back VRFs is IP. In this scenario, the VPNs are isolated from each other, and because the traffic is IP, QoS mechanisms that operate on IP traffic can be applied to achieve customer Service Level Agreements (SLAs). The downside of this configuration is that one BGP session is needed for each subinterface (and at least one subinterface for each VPN), which causes scalability concerns as this network grows.

In an Inter-AS Option B network, ASBR peers are connected by one or more subinterfaces that are enabled to receive MPLS traffic. A Multiprotocol Border Gateway Protocol (MP-BGP) session is used to distribute labeled VPN prefixes between the ASBR. As a result, the traffic that flows between them is labeled. The downside of this configuration is that, because the traffic is MPLS, QoS mechanisms that can be applied only to IP traffic cannot be applied and the VRFs cannot be isolated.

Prerequisites for MPLS VPN--Inter-AS Option AB

Follow the appropriate configuration tasks outlined in the following documents:

- Configuring MPLS Layer 3 VPNs

- MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses
- MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

Before configuring the MPLS VPN--Inter-AS Option AB feature, perform these tasks:

- Enable Cisco Express Forwarding, which is required for the MPLS VPN routing and forwarding operation.
- Identify the VPNs for the MPLS VPN--Inter-AS Option AB network and configure the VRFs to which these VPNs belong. These VRFs are used for Inter-AS Option AB connections on the ASBR interface.

Restrictions for MPLS VPN--Inter-AS Option AB

- The In Service Software Upgrade (ISSU) feature can be configured only on the active Route Processor (RP) if the standby RP supports this feature. The ISSU feature can be configured if both the active and standby RP support this feature.
- Carrier Supporting Carrier (CSC) MPLS load-balancing on ASBR Option AB VRF interfaces is not supported.
- VPNv6 is not supported.

Information About MPLS VPN--Inter-AS Option AB

MPLS VPN--Inter-AS Option AB Introduction

MPLS VPN service providers need to interconnect different autonomous systems to provide service for multiple VPN customers. The MPLS VPN--Inter-AS Option AB feature allows the different autonomous systems to interconnect by using a single MP-BGP session in the global routing table to carry control plane traffic. This MP-BGP session signals VPN prefixes between two ASBRs for each VRF instance. The data plane traffic is on a VRF interface. This traffic can either be IP or MPLS.

Note Inter-AS connections can be configured between ASBRs that either have or do not have connections between different providers.

Benefits of MPLS VPN--Inter-AS Option AB

The MPLS VPN--Inter-AS Option AB feature provides the following benefits for service providers:

- Network configuration can be simplified because only one BGP session is configured for each VRF on the ASBR.
- One BGP session reduces CPU utilization.
- Networks can be scaled because a single MP-BGP session, which is enabled globally on the router, reduces the number of sessions required by multiple VPNs, while continuing to keep VPNs isolated and secured from each other.

- IP QoS functions between ASBR peers are maintained for customer SLAs.
- Dataplane traffic is isolated on a per-VRF basis for security purposes.

Option B Style Peering with Shared Link Forwarding

An enhancement to Inter-AS Option AB is the MPLS VPN—Inter-AS Option AB+ feature. This feature addresses the scalability concerns of MPLS VPN—Inter-AS Option A by using a single BGP session in the global routing table to signal VPN prefixes (as described in Inter-AS Option B).

The key difference between Option AB+ and Option B is in the route distribution between ASBRs. In Option AB+, at the ASBR, the route that is imported into the VRF (with the route distinguisher and route targets of the VRF) is distributed to the neighboring ASBR. In Option B, the original pre-import route (with the original RD and RTs) is distributed to the neighboring ASBR and not the imported route.

With Option AB+, the PE and ASBRs deploy MPLS forwarding over a global interface, similar to what is done in Option B, and the signaling is handled by a single MP-eBGP VPNv4 session. The provider edge and ASBRs thus use regular Option B style peering between them. They receive MPLS-VPN traffic over the shared link and forward the traffic as per an IP lookup in the VRF routing table. However, the traffic is MPLS encapsulated, like it is in Option B.

Route Distribution and Packet Forwarding in Non-CSC Networks

The following sections describe MPLS VPN--Inter-AS Option AB operation:

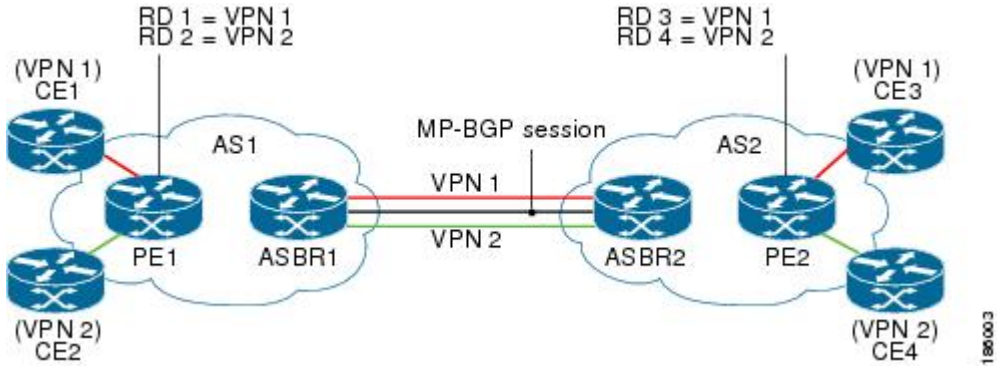
Note All imported routes are accomplished by configuring the appropriate route targets (RTs).

The following attributes describe the topology of the sample MPLS VPN--Inter-AS Option AB network shown in the figure below:

- Customer edge 1 (CE1) and CE3 belong to VPN 1.
- CE2 and CE 4 belong to VPN 2.
- Provider edge 1 (PE1) uses route distinguisher 1 (RD 1) for VPN 1 (VRF 1) and RD 2 for VPN 2 (VRF 2).
- PE2 uses RD 3 for VPN 1 (VRF 1) and RD 4 for VPN 2 (VRF 2).
- ASBR1 has VRF 1 provisioned with RD 5 and VRF 2 provisioned with RD 6.
- ASBR2 has VRF 1 provisioned with RD 7 and VRF 2 provisioned and RD 8.
- ASBR1 and ASBR2 have three links between them:
  - VRF 1
  - VRF 2
  - MP-BGP session

Note The VRFs configured on the ASBRs are called Option AB VRFs. The eBGP peers on the ASBRs are called Option AB Peers.

Figure 123: MPLS VPN Inter-AS Option AB Topology



Route Distribution for VPN 1

A route distinguisher (RD) is an identifier attached to a route that identifies which VPN belongs to each route. Each routing instance must have a unique RD autonomous system associated with it. The RD is used to place a boundary around a VPN so that the same IP address prefixes can be used in different VPNs without having these IP address prefixes overlap.

Note An RD statement is required if the instance type is a VRF.

The following process describes the route distribution process for VPN 1 in the figure above. Prefix “N” is used in this process to indicate the IP address of a VPN.

1. CE1 advertises the prefix N to PE1.
2. PE1 advertises a VPN prefix RD 1:N to ASBR1 through MP internal BGP (iBGP).
3. ASBR1 imports the prefix into VPN 1 and creates a prefix RD 5:N.
4. ASBR1 advertises the imported prefix RD 5:N to ASBR2. ASBR1 sets itself as the next hop for prefix RD 5:N and allocates a local label that is signaled with this prefix.
5. ASBR1 advertises the route with the export RT configured on the VRF rather than the originally received RTs. By default, ASBR1 does not advertise the source prefix RD 1:N to ASBR2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.

Note In an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1. ASBR2 receives the prefix RD 5:N and imports it into VPN 1 as RD 7:N.

2. ASBR2 advertises the route with the export RT configured on the VRF rather than the originally received RTs.
3. While importing the prefix, ASBR2 sets the next hop of RD 7:N to the ASBR1 interface IP address in VRF 1. The next hop table ID is also set to VRF 1. When installing the MPLS forwarding entry for RD 7:N, by default ASBR2 does not install the outgoing label in the forwarding process. This enables the traffic between the ASBRs to be IP.
4. ASBR2 advertises the imported prefix RD 7:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR2 does not advertise the source prefix RD 5:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.
5. PE2 imports the RD 7:N into VRF 1 as RD 3:N.

Packet Forwarding for VPN 1

The following packet forwarding process works the same as it does in an Option A scenario. The ASBR acts like the PE by terminating the VPN and then forwards its traffic as standard IP packets with no VPN label to the next PE, which in turn repeats the VPN process. Each PE router, therefore, treats the adjacent PE router as a CE router, and the standard Layer 3 MPLS VPN mechanisms are used for route redistribution with each autonomous system; that is, the PEs use external BGP (eBGP) to distribute unlabeled IPv4 addresses to each other.

Note Prefix “N” is used in this process to indicate the IP address of a VPN.

1. CE3 sends a packet destined for N to PE2.
2. PE2 encapsulates the packet with the VPN label allocated by ASBR2 and the Interior Gateway Protocol (IGP) label needed to tunnel the packet to ASBR2.
3. The packet arrives on ASBR2 with the VPN label. ASBR2 removes the VPN label and sends the packet as IP to ASBR1 on the VRF 1 interface.
4. The IP packet arrives at ASBR1 on the VRF 1 interface. ASBR1 then encapsulates the packet with the VPN label allocated by PE1 and the IGP label needed to tunnel the packet to PE1.
5. The packet arrives on PE1 with the VPN label. PE1 disposes the VPN label and forwards the IP packet to CE1.

Route Distribution for VPN 2

The following information describes the route distribution process for VPN 2 in the figure above:

1. CE2 advertises prefix N to PE1, where N is the VPN IP address.
2. PE1 advertises a VPN prefix RD 2:N to ASBR1 through MP-iBGP.
3. ASBR1 imports the prefix into VPN 2 and creates a prefix RD 6:N.
4. ASBR1 advertises the imported prefix RD 6:N to ASBR2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR1 does not advertise the source

prefix RD 2:N to ASBR2. This advertisement is suppressed as the prefix is being imported into an Option AB VRF.

Note In the case of an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1. ASBR2 receives the prefix RD 6:N and imports it into VPN 2 as RD 8:N.
2. While importing the prefix, ASBR2 sets the next hop of RD 8:N to ASBR1s interface address in VRF 2. The next hop table ID is also set to that of VRF 2. While installing the MPLS forwarding entry for RD 8:N, by default ASBR2 does not install the outgoing label in the forwarding process. This enables traffic between the ASBRs to be IP.
3. ASBR2 advertises the imported prefix RD 8:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR2 does not advertise the source prefix RD 6:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.
4. PE2 imports the RD 8:N into VRF 2 as RD 4:N.

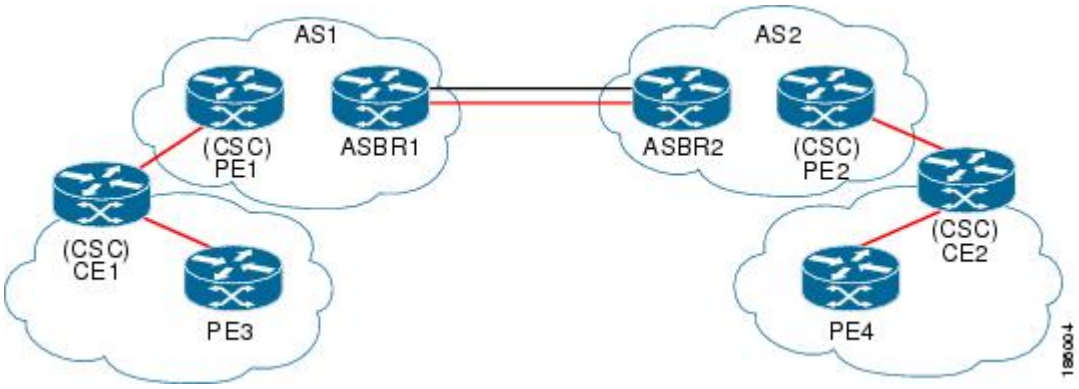
Route Distribution and Packet Forwarding for CSC

The following sections describe MPLS VPN--Inter-AS Option AB operation for a CSC scenario for VPN 1. These sections are similar to those found in Route Distribution and Packet Forwarding in Non-CSC Networks for VPN 1, except for the method in which MPLS labels are handled between the two ASBRs.

Note VPN 2 is not shown or discussed in this section.

The figure below shows how VPN 1 provides VPN service to a small customer carrier that in turn provides a VPN service to its customer. This configuration implies that VPN 1 is used to provide a label switched path (LSP) between the PE (PE 3 and PE 4) loopback interfaces of the small customer carrier.

Figure 124: MPLS VPN Inter-AS Option AB CSC Topology



Note The RD, RT, VRF, and Link provisioning in this section is the same as in the Route Distribution and Packet Forwarding in Non-CSC Networks example for VPN 1.

Route Distribution for VPN 1

The following information describe the route distribution process for VPN 1 in Figure 1 . Prefix "N" is used in these steps to indicate the IP address of a VPN.

1. CE1 advertises PE 3 loopback N to PE1.
2. PE1 advertises a VPN prefix RD 1:N to ASBR1 through MP-iBGP.
3. ASBR1 imports the prefix into VPN 1 and creates a prefix RD 5:N.
4. ASBR1 advertises the imported prefix RD 5:N to ASBR2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix.
5. ASBR1 advertises the route with the export RT configured on the VRF rather than the originally received RTs. By default, ASBR1 does not advertise the source prefix RD 1:N to ASBR2. This advertisement is suppressed as the prefix is being imported into an Option AB VRF.

Note In an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1. ASBR2 receives the prefix RD 5:N and imports it into VPN 1 as RD 7:N.
2. ASBR2 advertises the route with the export RT configured on the VRF rather than the originally received RTs.
3. While importing the prefix, ASBR2 sets the next hop of RD 7:N to the ASBR1 interface address in VRF 1. The next hop table ID is also set to that of VRF 1.

Note In a CSC scenario, an outgoing MPLS label can be installed in forwarding by making a configuration change.

1. While installing the MPLS forwarding entry for RD 7:N, ASBR2 installs the outgoing label during the forwarding process, which enables the traffic between the ASBRs to be MPLS traffic.
2. ASBR2 advertises the imported prefix RD 7:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR2 does not advertise the source prefix RD 5:N to PE2. This advertisement is suppressed as the prefix is being imported into an Option AB VRF.
3. PE2 imports the RD 7:N into VRF 1 as RD 3:N.

Packet Forwarding for VPN 1

The packet forwarding process shown below works the same as it does in an Option A scenario. See the Route Distribution and Packet Forwarding in Non-CSC Networks section for more information about Option A.

1. PE 4 sends an MPLS packet destined for N to CE2.
2. CE2 swaps the MPLS label and sends a packet destined for N to PE2.
3. PE2 encapsulates the packet with the VPN label allocated by ASBR2 and the IGP label needed to tunnel the packet to ASBR2.
4. The packet arrives on ASBR2 with the VPN label. ASBR2 swaps the received VPN label with the outgoing label received from ASBR1 and sends the MPLS packet on to the VRF 1 interface.
5. The MPLS packet arrives at ASBR1 on the VRF 1 interface. ASBR1 then swaps the received MPLS label with a label stack consisting of the VPN label allocated by PE1 and the IGP label needed to tunnel the packet to PE1.
6. The packet arrives on PE1 with the VPN label. PE1 disposes the VPN label and forwards the MPLS packet to CE1. CE1 in turn swaps the label and forwards the labeled packet to PE 3.

Shared Link Forwarding in Non-CSC Networks

Note All imported routes are accomplished by configuring the appropriate route targets (RTs).

The following attributes describe the sample network topology shown in the "Route Distribution and Packet Forwarding in Non-CSC Networks" section:

- Customer edge 1 (CE1) and CE3 belong to VPN 1.
- CE2 and CE 4 belong to VPN 2.
- Provider edge 1 (PE1) uses route distinguisher 1 (RD 1) for VPN 1 (VRF 1) and RD 2 for VPN 2 (VRF 2).
- PE2 uses RD 3 for VPN 1 (VRF 1) and RD 4 for VPN 2 (VRF 2).
- ASBR1 has VRF 1 provisioned with RD 5 and VRF 2 provisioned with RD 6.
- ASBR2 has VRF 1 provisioned with RD 7 and VRF 2 provisioned and RD 8.
- ASBR1 and ASBR2 have three links between them:
  - VRF 1
  - VRF 2
  - MP-BGP session

Note The VRFs configured on the ASBRs are called Option AB+ VRFs. The eBGP peers on the ASBRs are called Option AB+ Peers.

The following sections describe MPLS VPN—Inter-AS Option AB+ shared link forwarding in a non-CSC network:

Route Distribution for VPN 1

The following process describe the route distribution process for VPN 1 shown in the figure in the "Route Distribution and Packet Forwarding in Non-CSC Networks" section. Prefix “N” is used in these steps to indicate the IP address of a VPN.

1. CE1 advertises PE 3 loopback N to PE1.
2. PE1 advertises a VPN prefix RD 1:N to ASBR1 through MP-iBGP.
3. ASBR1 imports the prefix into VPN 1 and creates a prefix RD 5:N.
4. ASBR1 advertises the imported prefix RD 5:N to ASBR2. ASBR1 sets itself as the next hop for prefix RD 5:N and also allocates a local label that is signaled with this prefix.
5. By default, ASBR1 does not advertise the source prefix RD 1:N to ASBR2. This advertisement is suppressed because the prefix is being imported into an Option AB+ VRF.

Note In an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1. ASBR2 receives the prefix RD 5:N and imports it into VPN 1 as RD 7:N.
2. While importing the prefix, ASBR2 retains the next hop of RD7:N as received in the BGP update from ASBR2. This is the address of ASBR1 shared interface address in the global table. The next hop tableid is also left unchanged and corresponds to that of the global table
3. When installing the MPLS forwarding entry for RD 7:N, ASBR2 installs the outgoing label in the forwarding process. This enables the traffic between the ASBRs to be IP.
4. ASBR2 advertises the imported prefix RD 7:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix.
5. By default, ASBR2 does not advertise the source prefix RD 5:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB+ VRF.
6. PE2 imports the RD 7:N into VRF 1 as RD 3:N.

Packet Forwarding for VPN1

The following packet forwarding process works the same as it does in an Option B scenario.

1. CE3 sends a packet destined for N to PE2.
2. PE2 encapsulates the packet with the VPN label allocated by ASBR2 and the IGP label needed to tunnel the packet to ASBR2.
3. The packet arrives on ASBR2 with the VPN label. ASBR2 swaps the received VPN label with the outgoing label received from ASBR1 and sends the MPLS packet on the global shared link interface.

4. The MPLS packet arrives at ASBR1 on the global shared link interface. ASBR1 then swaps the received MPLS label with a label stack consisting of the VPN label allocated by PE1 and the IGP label needed to tunnel the packet to PE1.
5. The packet arrives on PE1 with the VPN label. PE1 removes the VPN label and forwards the IP packet to CE1.

How to Configure Inter-AS Option AB

The following sections describe how to configure the Inter-AS Option AB feature on an ASBR for either an MPLS VPN or an MPLS VPN that supports CSC:

Note If Inter-AS Option AB is already deployed in your network and you want to do Option B style peering for some prefixes (that is, implement Inter-AS Option AB+), configure the **inter-as-hybrid global** command as described in the "Configuring the Routing Policy for VPNs that Need Inter-AS Connections" section.

Configuring an Inter-AS Option AB Connection

The following sections are required and describe how to configure an Inter-AS Option AB connection on an ASBR:

Note See the Configuring MPLS Layer 3 VPNs feature module for more information on configuring PE and CE routers in an MPLS VPN.

Configuring the VRFs on the ASBR Interface for Each VPN Customer

Use the following steps to configure the VRFs on the ASBR interface for each VPN customer so that these VPNs have connectivity over the MPLS VPN--Inter-AS Option AB network.

Note The **mpls bgp forwarding** command is used only on the ASBR interface for VRFs that support CSC.

Use all of the steps in the following procedure to configure additional VRFs that need to be configured on the ASBR interface and the VRFs that need to be configured on the peer ASBR interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **mpls bgp forwarding**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 5/0`	Specifies the interface to configure and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 5	**mpls bgp forwarding** **Example:** `Router(config-if)# mpls bgp forwarding`	(Optional) Configures BGP to enable MPLS forwarding on connecting interfaces for VRFs that must support MPLS traffic. • This step applies to a CSC network only.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the MP-BGP Session Between ASBR Peers

BGP propagates reachability information for VPN-IPv4 prefixes among PE routers by means of the BGP multiprotocol extensions (see RFC 2283, *Multiprotocol Extensions for BGP-4*), which define support for address families other than IPv4. Using the extensions ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

Follow the steps in this section to configure the MP-BGP session on the ASBR.

Use all of the steps in the following procedure to configure the MP BGP session on the peer ASBR.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** {*ip-address* | *peer-group-name*} **inter-as-hybrid**
8. **exit-address-family**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 192.168.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes.

	Command or Action	Purpose
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 192.168.0.1 activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **inter-as-hybrid** **Example:** `Router(config-router-af)# neighbor 192.168.0.1 inter-as-hybrid`	Configures eBGP peer router (ASBR) as an Inter-AS Option AB peer. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • If any prefixes are imported into Option AB VRFs, then the imported paths are advertised to this peer. • If any prefixes are received from this peer and are imported into Option AB VRFs, then the imported paths are advertised to iBGP peers. **Note** Advertised routes have RTs that are configured on the VRF. Advertised routes do not have their original RTs.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the Routing Policy for VPNs that Need Inter-AS Connections

Use the steps in this section to configure VRFs for the VPNs that need Inter-AS connections between ASBR peers, by configuring the appropriate routing policy and Option AB configuration.

Use all of the steps in the following procedure to configure additional VPNs that need Inter-AS Option AB connectivity on this ASBR and the peer ASBR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family ipv4**
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. For Inter-AS Option AB+, go to Step 10; otherwise, go to Step 8.
8. **inter-as-hybrid** [**csc**]
9. **inter-as-hybrid** [**csc**] [**next-hop** *ip-address*]
10. **inter-as-hybrid next-hop global**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	Example:
	`Router(config-vrf)# rd 100:1`
		Creates routing and forwarding tables.
		• The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		  • 16-bit autonomous system number: your 32-bit number, for example, 101:3
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

Step 5	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF.
Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:**	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community.

	Command or Action	Purpose
	`Router(config-vrf-af)# route-target import 100:1`	• The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF list of import, export, or both (import and export) route-target extended communities.
Step 7	For Inter-AS Option AB+, go to Step 10; otherwise, go to Step 8.	—
Step 8	**inter-as-hybrid** [**csc**] **Example:** `Router(config-vrf-af)# inter-as-hybrid`	Specifies the VRF as an Option AB VRF, which has the following effects: • Routes imported to this VRF can be advertised to Option AB peers and VPNv4 iBGP peers. • When routes received from Option AB peers and are imported into the VRF, the next hop table ID of the route is set to the table ID of the VRF. • If the **csc** keyword is not used, a per-VRF label is allocated for imported routes. • When routes are received from Option AB peers and are imported next into the VRF, the learned out label can be installed only in forwarding when the **csc** keyword is used. The **csc** keyword implies the following: • A per-prefix label is allocated for imported routes. • For routes received from Option AB peers that are imported into the VRF, the learned out label is installed in forwarding.
Step 9	**inter-as-hybrid** [**csc**] [**next-hop** *ip-address*] **Example:** `Router(config-vrf-af)# inter-as-hybrid next-hop 192.168.1.0`	(Optional) Specifies the next hop IP address to be set on paths that are imported into the VRF and that are received from an Option AB peer. • The next hop context is also set to the VRF, which imports these paths. • The **csc** keyword implies the following: • A per-prefix label is allocated for imported routes.

	Command or Action	Purpose
		• For routes received from Option AB peers that are imported into the VRF, the learned out label is installed in forwarding.
Step 10	**inter-as-hybrid next-hop global** **Example:** `Router(config-vrf-af)# inter-as-hybrid next-hop global`	(For Option AB+) Enables Inter-AS Option AB+. • Specifies that the next-hop address for BGP updates to be set on paths that are imported to the VRF and that are received from an Option AB+ peer are placed in the global routing table. • The address used is the address of the interface that is at the remote end of the external BGP (eBGP) global shared link. The next-hop context is retained as global and not modified to that of the importing VRF.
Step 11	**end** **Example:** `Router(config-vrf-af)# end`	(Optional) Exits to privileged EXEC mode.

Changing an Inter-AS Option A Deployment to an Option AB Deployment

In an Option A deployment, the VRF instances are back-to-back between the ASBR routers and there is direct connectivity between PE routers of different autonomous systems. The PE routers are attached by multiple physical or logical interfaces, each of which is associated with a given VPN (through a VRF instance).

In the Option AB deployment, the different autonomous systems interconnect by using a single MP-BGP session in the global routing table to carry control plane traffic.

Use the following steps to change an MPLS VPN Inter-AS Option A deployment to an Option AB deployment.

1. Configure the MP-BGP session on the ASBR. BGP multiprotocol extensions are used to define support for address families other than IPv4 so that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other. See the Configuring the MP-BGP Session Between ASBR Peers, on page 1533 for detailed configuration information.
2. Identify the VRFs that need an upgrade from Option A and configure them for Option AB by using the **inter-as-hybrid** command. See the Configuring the Routing Policy for VPNs that Need Inter-AS Connections, on page 1535 for detailed configuration information.
3. Use the following steps in this section to remove the configuration for the eBGP (peer ASBR) neighbor.
4. Repeat all the steps in the following procedure to remove the configuration for additional eBGP (peer ASBR) neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **router bgp** *as-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **no neighbor** {*ip-address* | *peer-group-name*}
6. **exit-address-family**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf vpn4`	Configures each VRF that is identified in the MP-BGP session on the ASBR so that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other. • Enters address family configuration mode to specify an address family for a VRF.
Step 5	**no neighbor** {*ip-address* | *peer-group-name*} **Example:** `Router(config-router-af)# no neighbor 192.168.0.1`	Removes the configuration for the exchange of information with the neighboring eBGP (ASBR) router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 6	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuration Examples for MPLS VPN--Inter-AS Option AB

The following sections describe standard and CSC MPLS VPN configurations between two ASBR peers that use the Inter-AS AB feature:

Examples Inter-AS AB Network Configuration

The following examples show the configuration of an Inter-AS Option AB network that uses nonoverlapping IP addresses:

Example CE1

```
!
ip cef distributed
!
interface lo0
 ip address 192.168.13.13 255.255.255.255
 no shutdown
!
interface et4/0
 ip address 192.168.36.1 255.255.255.0
 no shutdown
!
router ospf 300
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface et4/0
 network 192.168.13.13 0.0.0.0 area 300
!
router bgp 300
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.36.2 remote-as 100
 neighbor 192.168.36.2 advertisement-interval 5
 address-family ipv4 no auto-summary
 redistribute connected
 neighbor 192.168.36.2 activate
```

Example CE2

```
!
ip cef distributed
!
interface lo0
```

```
 ip address 192.168.14.14 255.255.255.255
 no shutdown
!
interface et1/6
 ip address 192.168.37.1 255.255.255.0
 no ipv6 address
 no shutdown
!
router ospf 400
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface et1/6
 network 192.168.14.14 0.0.0.0 area 400
!
router bgp 400
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.0.2 remote-as 100
 neighbor 192.168.0.2 advertisement-interval 5
 address-family ipv4  no auto-summary
 redistribute connected
 neighbor 192.168.0.2 activate
!
```

Example PE1

```
!
ip cef distributed
!
ip vrf vpn1
   rd 100:1
   route-target import 100:1
   route-target import 200:1
   route-target export 100:1
!
ip vrf vpn2
   rd 100:2
   route-target import 100:2
   route-target import 200:2
   route-target export 100:2
!
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls label protocol ldp
!
interface lo0
 ip address 192.168.17.17 255.255.255.255
 no shutdown
!
interface gi3/1
 ip vrf forwarding vpn1
 ip address 192.168.36.2 255.255.255.0
 no shutdown
!
interface gi3/8
 mpls ip
 mpls label protocol ldp
```

```
 ip address 192.168.31.2 255.255.255.0
!
interface gi3/10
 mpls ip
 mpls label protocol ldp
 ip address 192.168.40.1 255.255.255.0
 no shutdown
!
interface gi3/13
 ip vrf forwarding vpn2
 ip address 192.168.0.2 255.0.0.0
 no shutdown
!
router ospf 100
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface gi3/1
 passive-interface gi3/13
 network 192.168.0.0 0.0.255.255 area 10
 network 192.168.17.17 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100
!
router bgp 100
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 no synchronization
 neighbor 192.168.19.19 remote-as 100
 neighbor 192.168.19.19 update-source Loopback0
 address-family ipv4 vrf vpn1
no auto-summary
 redistribute connected
 neighbor 192.168.36.1 remote-as 300
 neighbor 192.168.36.1 activate
 neighbor 192.168.36.1 advertisement-interval 5
 address-family ipv4 vrf vpn2  no auto-summary
 redistribute connected
 neighbor 192.168.37.1 remote-as 400
 neighbor 192.168.37.1 activate
 neighbor 192.168.37.1 advertisement-interval 5
 address-family vpnv4
 bgp scan-time import 5
 neighbor 192.168.19.19 activate
 neighbor 192.168.19.19 send-community extended
!
```

Example Route Reflector 1

```
!
ip cef distributed
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls ip
mpls label protocol ldp
!
interface lo0
 ip address 192.168.19.19 255.255.255.255
 no shutdown
```

```
!
interface gi3/3
 mpls ip
 mpls label protocol ldp
 ip address 192.168.40.2 255.255.255.0
 no shutdown
!
router ospf 100
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 network 192.168.19.19 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100 !
router bgp 100
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.11.11 remote-as 100
 neighbor 192.168.11.11 update-source Loopback0
 neighbor 192.168.17.17 remote-as 100
 neighbor 192.168.17.17 update-source Loopback0
 neighbor 192.168.11.11 route-reflector-client
 address-family ipv4
 no neighbor 192.168.17.17 activate
 neighbor 192.168.11.11 route-reflector-client
 address-family vpnv4
 bgp scan-time import 5
 neighbor 192.168.11.11 activate
 neighbor 192.168.11.11 send-community extended
 neighbor 192.168.17.17 activate
 neighbor 192.168.17.17 send-community extended
 neighbor 192.168.11.11 route-reflector-client
 neighbor 192.168.17.17 route-reflector-client
!
```

Example ASBR1

```
!
ip cef distributed
!
ip vrf vpn1
   rd 100:1
   route-target import 100:1
   route-target import 200:1
   route-target export 100:1
   inter-as-hybrid next-hop 192.168.32.2
exit
ip vrf vpn2
   rd 100:2
   route-target import 100:2
   route-target import 200:2
   route-target export 100:2
   inter-as-hybrid next-hop 192.168.33.2
exit
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
 mpls ip
 mpls label protocol ldp
interface lo0
 ip address 192.168.11.11 255.255.255.255
```

```
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/8
mpls ip
 mpls label protocol ldp
 ip address 192.168.13.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/10
   ip vrf forwarding vpn1
 ip address 192.168.32.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/11
   ip vrf forwarding vpn2
 ip address 192.168.33.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/46
 ip address 192.168.34.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
router ospf 100
nsf enforce global
 redistribute connected subnets
auto-cost reference-bandwidth 1000
passive-interface gi3/10
passive-interface gi3/11
passive-interface gi3/46
network 192.168.0.0 0.0.255.255 area 100
network 192.168.11.11 0.0.0.0 area 100

router bgp 100
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 no bgp default route-target filter
 bgp router-id 192.168.11.11
 neighbor 192.168.34.2 remote-as 200
 neighbor 192.168.34.2 advertisement-interval 5
 neighbor 192.168.19.19 remote-as 100
 neighbor 192.168.19.19 update-source Loopback0
 address-family ipv4
  no auto-summary
 address-family ipv4 vrf vpn1
  no auto-summary
 address-family ipv4 vrf vpn2
  no auto-summary
 address-family vpnv4
  bgp scan-time import 5
  neighbor 192.168.34.2 activate
  neighbor 192.168.34.2 send-community both
  neighbor 192.168.34.2 inter-as-hybrid
```

```
  neighbor 192.168.19.19 activate
  neighbor 192.168.19.19 send-community extended !
ip route vrf vpn1 192.168.12.12 255.255.255.255 gi3/10 192.168.32.2
ip route vrf vpn2 192.168.12.12 255.255.255.255 gi3/11 192.168.33.2
!
```

Example ASBR 3

```
!
ip cef distributed
!
ip vrf vpn1
   rd 200:1
   route-target import 100:1
   route-target import 200:1
   route-target export 200:1
   inter-as-hybrid next-hop 192.168.32.1
!
ip vrf vpn2
   rd 200:2
   route-target import 100:2
   route-target import 200:2
   route-target export 200:2
   inter-as-hybrid next-hop 192.168.33.1
!
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls label protocol ldp
!
interface lo0
 ip address 192.168.12.12 255.255.255.255
 no shutdown
!
interface po2/1/0
 mpls ip
 mpls label protocol ldp
 ip address 192.168.35.1 255.255.255.0
 crc 16
 clock source internal
 no shutdown
!
interface gi3/10
 ip vrf forwarding vpn1
 ip address 192.168.32.2 255.255.255.0
 no shutdown
!
interface gi3/11
 ip vrf forwarding vpn2
 ip address 192.168.33.2 255.255.255.0
 no shutdown
!
interface gi3/45
 ip address 192.168.34.2 255.255.255.0
 no shutdown
!
router ospf 200
nsf enforce global
redistribute connected subnets
auto-cost reference-bandwidth 1000
passive-interface gi3/10
```

```
passive-interface gi3/11
passive-interface gi3/45
network 192.168.0.0 0.0.255.255 area 200 network 192.168.12.12 0.0.0.0 area 200

router bgp 200
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 no bgp default route-target filter
 bgp router-id 192.168.12.12
 neighbor 192.168.34.1 remote-as 100
 neighbor 192.168.34.1 advertisement-interval 5
 neighbor 192.168.20.20 remote-as 200
 neighbor 192.168.20.20 update-source Loopback0
 address-family ipv4
 no auto-summary
 address-family ipv4 vrf vpn1
  no auto-summary
 address-family ipv4 vrf vpn2
  no auto-summary
 address-family vpnv4
  bgp scan-time import 5
  neighbor 192.168.34.1 activate
  neighbor 192.168.34.1 send-community both
  neighbor 192.168.34.1 inter-as-hybrid
  neighbor 192.168.20.20 activate
  neighbor 192.168.20.20 send-community extended !
ip route vrf vpn1 192.168.11.11 255.255.255.255 gi3/10 192.168.32.1
ip route vrf vpn2 192.168.11.11 255.255.255.255 gi3/11 192.168.33.1
!
```

Example PE2

```
!
ip cef distributed
!
ip vrf vpn1
   rd 200:1
   route-target import 100:1
   route-target import 200:1
   route-target export 200:1
!
ip vrf vpn2
   rd 200:2
   route-target import 100:2
   route-target import 200:2
   route-target export 200:2
!
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls label protocol ldp
!
interface lo0
 ip address 192.168.18.18 255.255.255.255
 no shutdown
!
interface po1/0/0
 mpls ip
 mpls label protocol ldp
```

```
 ip address 192.168.35.2 255.255.255.0
 crc 16
 clock source internal
 no shutdown
!
interface gi3/2
 ip vrf forwarding vpn1
 ip address 192.168.38.2 255.255.255.0
 no shutdown
!
interface gi3/8
 mpls ip
 mpls label protocol ldp
 ip address 192.168.4.1 255.255.255.0
 no shutdown
!
interface gi3/10
 ip vrf forwarding vpn2
 ip address 192.168.39.2 255.255.255.0
 no shutdown
!
router ospf 200
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface gi3/10
 passive-interface gi3/2
 network 192.168.0.0 0.0.255.255 area 200
 network 192.168.18.18 0.0.0.0 area 200
 network 192.168.0.0 0.0.255.255 area 200 !
 router bgp 200
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 no synchronization
 neighbor 192.168.20.20 remote-as 200
 neighbor 192.168.20.20 update-source Loopback0
 address-family ipv4 vrf vpn1
    no auto-summary
    redistribute connected
    neighbor 192.168.38.1 remote-as 500
    neighbor 192.168.38.1 activate
    neighbor 192.168.38.1 advertisement-interval 5
 address-family ipv4 vrf vpn2
    no auto-summary
    redistribute connected
    neighbor 192.168.9.1 remote-as 600
    neighbor 192.168.9.1 activate
    neighbor 192.168.9.1 advertisement-interval 5
 address-family vpnv4
    bgp scan-time import 5
    neighbor 192.168.20.20 activate
    neighbor 192.168.20.20 send-community extended
!
```

Example CE3

```
!
ip cef distributed
!
interface lo0
 ip address 192.168.15.15 255.255.255.255
```

```
 no shutdown
!
interface gi0/2
 ip address 192.168.38.1 255.255.255.0
 no shutdown
!
router ospf 500
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface gi0/2
 network 192.168.15.15 0.0.0.0 area 500
!
router bgp 500
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.38.2 remote-as 200
 neighbor 192.168.38.2 advertisement-interval 5
 address-family ipv4
 no auto-summary
 redistribute connected
 neighbor 192.168.38.2 activate
!
```

Example CE4

```
!
ip cef distributed
!
interface lo0
 ip address 192.168.16.16 255.255.255.255
 no shutdown
!
interface et6/2
 ip address 192.168.9.1 255.255.255.0
 no shutdown
!
router ospf 600
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface et6/2
 network 192.168.16.16 0.0.0.0 area 600
!
router bgp 600
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.39.2 remote-as 200
 neighbor 192.168.39.2 advertisement-interval 5
 address-family ipv4  no auto-summary
 redistribute connected
 neighbor 192.168.39.2 activate
!
```

Examples Inter-AS AB CSC Configuration

The following examples show the configuration of an Inter-AS Option AB network with CSC:

Example CE1

```
!
ip cef distributed
!
interface Loopback0
 ip address 192.168.20.20 255.255.255.255
!
interface Ethernet3/3
 ip address 192.168.41.2 255.255.255.0
!
!
router bgp 500
 bgp router-id 192.168.20.20
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.4.1 remote-as 300
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.4.1 activate
  neighbor 192.168.4.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CE2

```
!
ip cef distributed
!
interface Loopback0
 ip address 192.168.21.21 255.255.255.255
!
interface Ethernet0/0/7
 ip address 192.168.42.2 255.255.255.0
!
router bgp 600
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart  neighbor 192.168.42.1 remote-as 400
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.42.1 activate
  neighbor 192.168.42.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CE3

```
!
ip cef distributed
!
```

```
interface Loopback0
 ip address 192.168.22.22 255.255.255.255
!
interface Ethernet6/2
 ip address 192.168.43.2 255.255.255.0
!
router bgp 500
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart neighbor 192.168.43.1 remote-as 300
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.43.1 activate
  neighbor 192.168.43.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CE4

```
!
ip cef distributed
!
interface Loopback0
 ip address 192.168.23.23 255.255.255.255
!
!
interface Ethernet0/0/7
 ip address 192.168.44.2 255.255.255.0
!
router bgp 600
 bgp router-id 192.168.23.23
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
neighbor 192.168.44.1 remote-as 400
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.44.1 activate
  neighbor 192.168.44.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example PE1

```
!
ip cef distributed
!
ip vrf vpn3
 rd 300:3
 route-target export 300:3
 route-target import 300:3
!
mpls ldp graceful-restart
```

```
!
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.192.10 255.255.255.255
!
interface Ethernet3/1
 ip vrf forwarding vpn3
 ip address 192.168.4.1 255.255.255.0
!
interface Ethernet5/3
 ip address 192.168.3.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 network 192.168.192.10 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
!
router bgp 300
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.19.19 remote-as 300
 neighbor 192.168.19.19 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.19.19 activate
  neighbor 192.168.19.19 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn3
  redistribute connected
  neighbor 192.168.41.2 remote-as 500
  neighbor 192.168.41.2 activate
  neighbor 192.168.41.2 as-override
  neighbor 192.168.41.2 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-CE1

```
!
ip cef distributed
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.11.11 255.255.255.255
```

```
!
!
interface Ethernet3/4
 ip address 192.168.30.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 300 metric 3 subnets
 passive-interface FastEthernet1/0
 network 192.168.11.11 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
 distance ospf intra-area 19 inter-area 19
!
router bgp 300
 bgp router-id 192.168.11.11
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.13.1 remote-as 100
 !
 address-family ipv4
  redistribute ospf 300 metric 4 match internal external 1 external 2
  neighbor 192.168.13.1 activate
  neighbor 192.168.13.1 send-label
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-PE1

```
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 route-target import 100:5
 route-target import 200:1
!
ip vrf vpn2
 rd 100:2
 route-target export 100:2
 route-target import 100:2
 route-target import 100:6
 route-target import 200:2
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.12.12 255.255.255.255
!
!
interface FastEthernet4/0/0
 ip address 192.168.34.1 255.255.255.0
```

```
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet4/0/1
 ip vrf forwarding vpn1
 ip address 192.168.13.1 255.255.255.0
 mpls bgp forwarding
!
!
interface FastEthernet4/1/0
 ip vrf forwarding vpn2
 ip address 192.168.33.1 255.255.255.0
 mpls bgp forwarding
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.12.12 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100
!
router bgp 100
 bgp router-id 192.168.12.12
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.15.15 remote-as 100
 neighbor 192.168.15.15 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.15.15 activate
  neighbor 192.168.15.15 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn2
  neighbor 192.168.33.2 remote-as 400
  neighbor 192.168.33.2 update-source FastEthernet4/1/0
  neighbor 192.168.33.2 activate
  neighbor 192.168.33.2 as-override
  neighbor 192.168.33.2 advertisement-interval 5
  neighbor 192.168.33.2 send-label
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn1
  neighbor 192.168.31.2 remote-as 300
  neighbor 192.168.31.2 update-source FastEthernet4/0/1
  neighbor 192.168.31.2 activate
  neighbor 192.168.31.2 as-override
  neighbor 192.168.31.2 advertisement-interval 5
  neighbor 192.168.31.2 send-label
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example PE 2

```
ip cef distributed
!
ip vrf vpn4
 rd 400:4
 route-target export 400:4
 route-target import 400:4
!
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.13.13 255.255.255.255
!
!
interface Ethernet4/1/2
 ip vrf forwarding vpn4
 ip address 192.168.42.1 255.255.255.0
!
!
interface Ethernet4/1/6
 ip address 192.168.32.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 400
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.13.13 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
!
router bgp 400
 bgp router-id 192.168.13.13
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.25.25 remote-as 400
 neighbor 192.168.25.25 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.25.25 activate
  neighbor 192.168.25.25 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn4
  redistribute connected
  neighbor 192.168.42.2 remote-as 600
  neighbor 192.168.42.2 activate
  neighbor 192.168.42.2 as-override
  neighbor 192.168.42.2 advertisement-interval 5
  no auto-summary
  no synchronization
```

```
 exit-address-family
!
```

Example CSC-CE2

```
!
ip cef distributed
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
interface Loopback0
 ip address 192.168.14.14 255.255.255.255
!
!
interface GigabitEthernet8/16
 ip address 192.168.33.2 255.255.255.0
 mpls bgp forwarding
!
!
interface GigabitEthernet8/24
 ip address 192.168.32.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 400
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 redistribute bgp 400 metric 3 subnets
 passive-interface GigabitEthernet8/16
 network 192.168.14.14 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
 distance ospf intra-area 19 inter-area 19
!
router bgp 400
 bgp router-id 192.168.14.14
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.33.1 remote-as 100
 !
 address-family ipv4
  no synchronization
  redistribute connected
  redistribute ospf 400 metric 4 match internal external 1 external 2
  neighbor 192.168.33.1 activate
  neighbor 192.168.33.1 advertisement-interval 5
  neighbor 192.168.33.1 send-label
  no auto-summary
 exit-address-family
!
```

Example ASBR1

```
!
```

```
ip vrf vpn5
 rd 100:5
 route-target export 100:5
 route-target import 100:5
 route-target import 100:1
 route-target import 200:5
 inter-as-hybrid csc next-hop 192.168.35.2
!
ip vrf vpn6
 rd 100:6
 route-target export 100:6
 route-target import 100:6
 route-target import 100:2
 route-target import 200:6
 inter-as-hybrid csc next-hop 192.168.36.2
!
mpls ldp graceful-restart
mpls label protocol ldp
!
!
interface Loopback0
 ip address 192.168.15.15 255.255.255.255
!
interface GigabitEthernet2/3
 ip vrf forwarding vpn5
 ip address 192.168.35.1 255.255.255.0
 mpls bgp forwarding
!
interface GigabitEthernet2/4
 ip vrf forwarding vpn6
 ip address 192.168.36.1 255.255.255.0
 mpls bgp forwarding
!
!
interface GigabitEthernet2/5
 ip address 192.168.34.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
interface GigabitEthernet2/16
 ip address 192.168.37.1 255.255.255.0
 mpls bgp forwarding
!
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.15.15 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100
!
router bgp 100
 bgp router-id 192.168.15.15
 no bgp default ipv4-unicast
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.12.12 remote-as 100
 neighbor 192.168.12.12 update-source Loopback0
 neighbor 192.168.0.2 remote-as 200
```

```
 neighbor 192.168.0.2 disable-connected-check
 !
 address-family ipv4
  no synchronization
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.12.12 activate
  neighbor 192.168.12.12 send-community extended
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 send-community extended
  neighbor 192.168.0.2 inter-as-hybrid
 exit-address-family
 !
 address-family ipv4 vrf vpn5
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn6
  no synchronization
 exit-address-family
!
ip route 192.168.16.16 255.255.255.255 GigabitEthernet2/16 192.168.0.2
ip route vrf vpn5 192.168.16.16 255.255.255.255 GigabitEthernet2/3 192.168.35.2
ip route vrf vpn6 192.168.16.16 255.255.255.255 GigabitEthernet2/4 192.168.36.2
!
ip vrf vpn5
 rd 200:5
 route-target export 200:5
 route-target import 200:5
 route-target import 200:1
 route-target import 100:1
 route-target import 100:5
 inter-as-hybrid csc next-hop 192.168.35.1
!
ip vrf vpn6
 rd 200:6
 route-target export 200:6
 route-target import 200:6
 route-target import 200:2
 route-target import 100:2
 route-target import 100:6
 inter-as-hybrid csc next-hop 192.168.36.1
!
mpls ldp graceful-restart
mpls label protocol ldp
!
!
interface Loopback0
 ip address 192.168.16.16 255.255.255.255
!
!
interface GigabitEthernet3/1
 ip vrf forwarding vpn5
 ip address 192.168.35.2 255.255.255.0
 mpls bgp forwarding
!
interface GigabitEthernet3/2
 ip vrf forwarding vpn6
 ip address 192.168.36.2 255.255.255.0
 mpls bgp forwarding
!      !
interface GigabitEthernet3/14
```

```
 ip address 192.168.0.2 255.0.0.0
 mpls bgp forwarding
!
interface GigabitEthernet3/15
 ip address 192.168.38.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.16.16 0.0.0.0 area 200
 network 192.168.0.0 0.0.255.255 area 200
!
router bgp 200
 bgp router-id 192.168.16.16
 no bgp default ipv4-unicast
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.17.17 remote-as 200
 neighbor 192.168.17.17 update-source Loopback0
 neighbor 192.168.37.1 remote-as 100
 neighbor 192.168.37.1 disable-connected-check
 !
 address-family ipv4
  no synchronization
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.17.17 activate
  neighbor 192.168.17.17 send-community extended
  neighbor 192.168.37.1 activate
  neighbor 192.168.37.1 send-community extended
  neighbor 192.168.37.1 inter-as-hybrid
 exit-address-family
 !
 address-family ipv4 vrf vpn5
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn6
  no synchronization
 exit-address-family
!
ip route 192.168.15.15 255.255.255.255 GigabitEthernet3/14 192.168.37.1
ip route vrf vpn5 192.168.15.15 255.255.255.255 GigabitEthernet3/1 192.168.35.1
ip route vrf vpn6 192.168.15.15 255.255.255.255 GigabitEthernet3/2 192.168.36.1
!
```

Example CSC-PE 3

```
ip vrf vpn1
 rd 200:1
 route-target export 200:1
 route-target import 200:1
 route-target import 200:5
 route-target import 100:1
```

```
!
ip vrf vpn2
 rd 200:2
 route-target export 200:2
 route-target import 200:2
 route-target import 200:6
 route-target import 100:2
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.17.17 255.255.255.255
!
interface FastEthernet4/0/2
 ip vrf forwarding vpn2
 ip address 192.168.5.1 255.255.255.0
 mpls bgp forwarding
!
!
interface FastEthernet4/0/4
 ip vrf forwarding vpn1
 ip address 192.168.9.1 255.255.255.0
 mpls bgp forwarding
!
!
interface FastEthernet4/0/7
 ip address 192.168.38.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.17.17 0.0.0.0 area 200
 network 192.168.0.0 0.0.255.255 area 200
!
router bgp 200
 bgp router-id 192.168.17.17
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.16.16 remote-as 200
 neighbor 192.168.16.16 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.16.16 activate
  neighbor 192.168.16.16 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn2
  neighbor 192.168.55.0 remote-as 400
  neighbor 192.168.55.0 update-source FastEthernet4/0/2
  neighbor 192.168.55.0 activate
  neighbor 192.168.55.0 as-override
  neighbor 192.168.55.0 advertisement-interval 5
  neighbor 192.168.55.0 send-label
```

```
 no auto-summary
 no synchronization
exit-address-family
!
address-family ipv4 vrf vpn1
 neighbor 192.168.39.2 remote-as 300
 neighbor 192.168.39.2 update-source FastEthernet4/0/4
 neighbor 192.168.39.2 activate
 neighbor 192.168.39.2 as-override
 neighbor 192.168.39.2 advertisement-interval 5
 neighbor 192.168.39.2 send-label
 no auto-summary
 no synchronization
exit-address-family
!
```

Example CSC-CE3

```
!
interface Loopback0
 ip address 192.168.18.18 255.255.255.255
!
!
interface Ethernet3/3
 ip address 192.168.40.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
interface FastEthernet5/0
 ip address 192.168.39.2 255.255.255.0
 mpls bgp forwarding
!
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 300 metric 3 subnets
 network 192.168.18.18 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
 distance ospf intra-area 19 inter-area 19
!
router bgp 300
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.9.1 remote-as 200
 !
 address-family ipv4
  redistribute connected
  redistribute ospf 300 metric 4 match internal external 1 external 2
  neighbor 192.168.9.1 activate
  neighbor 192.168.9.1 advertisement-interval 5
  neighbor 192.168.9.1 send-label
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-CE 4

```
!
ip cef distributed
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.24.24 255.255.255.255
!
!
interface FastEthernet1/1
 ip address 192.168.55.0 255.255.255.0
 mpls bgp forwarding
!
!
interface Ethernet3/5
 ip address 192.168.56.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 400
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 400 metric 3 subnets
 network 192.168.24.24 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
!
router bgp 400
 bgp log-neighbor-changes
 neighbor 192.168.5.1 remote-as 200
 !
 address-family ipv4
  redistribute connected
  redistribute ospf 400 metric 4 match internal external 1 external 2
  neighbor 192.168.5.1 activate
  neighbor 192.168.5.1 advertisement-interval 5
  neighbor 192.168.5.1 send-label
  no auto-summary
  no synchronization
 exit-address-family
```

Example PE 3

```
!
ip cef distributed
!
ip vrf vpn3
 rd 300:3
 route-target export 300:3
 route-target import 300:3
 mpls ip
!
!
mpls ldp graceful-restart
mpls label protocol ldp
```

```
!
!
interface Loopback0
 ip address 192.168.19.19 255.255.255.255
!
!
interface Ethernet5/1/1
 ip vrf forwarding vpn3
 ip address 192.168.43.1 255.255.255.0
!
!
interface Ethernet5/1/4
 ip address 192.168.40.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.19.19 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
 network 192.168.0.0 0.0.255.255 area 300
!
router bgp 300
 bgp router-id 192.168.19.19
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.192.10 remote-as 300
 neighbor 192.168.192.10 update-source Loopback0
 !
 address-family ipv4
  no neighbor 192.168.192.10 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.192.10 activate
  neighbor 192.168.192.10 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn3
  neighbor 192.168.43.2 remote-as 500
  neighbor 192.168.43.2 activate
  neighbor 192.168.43.2 as-override
  neighbor 192.168.43.2 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
```

Example PE 4

```
!
ip cef distributed
!
ip vrf vpn4
 rd 400:4
```

```
 route-target export 400:4
 route-target import 400:4
!
mpls ldp graceful-restart
mpls ldp protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.25.25 255.255.255.255
!
!
interface Ethernet5/0/4
 ip address 192.168.56.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
interface Ethernet5/0/7
 ip vrf forwarding vpn4
 ip address 192.168.44.1 255.255.255.0
!
!
router ospf 400
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.25.25 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
!
router bgp 400
 bgp router-id 192.168.25.25
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.13.13 remote-as 400
 neighbor 192.168.13.13 ebgp-multihop 7
 neighbor 192.168.13.13 update-source Loopback0
 !
 address-family ipv4
  no neighbor 192.168.13.13 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.13.13 activate
  neighbor 192.168.13.13 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn4
  neighbor 192.168.44.2 remote-as 600
  neighbor 192.168.44.2 activate
  neighbor 192.168.44.2 as-override
  neighbor 192.168.44.2 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN interautonomous systems	• MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses • MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2283	*Multiprotocol Extensions for BGP-4*
RFC 4364	*BGP/MPLS IP Virtual Private Networks*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN--Inter-AS Option AB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 149: Feature Information for MPLS VPN--Inter-AS Option AB

Feature Name	Release	Feature Information
MPLS VPN--Inter-AS Option AB	12.2(33)SRC 15.0(1)M 15.0(1)S 15.0(1)SY Cisco IOS XE Release 2.4	This feature combines the best functionality of an Inter-AS Option 10 A and Inter-AS Option 10 B network to allow an MPLS VPN service provider to interconnect different autonomous systems to provide VPN services. In Cisco IOS Release 12.2(33)SRC, this feature was introduced. In Cisco IOS Release 15.0(1)M, this feature was implemented on Cisco 1900, 2900, 3800, and 3900 series routers. In Cisco IOS XE Release 2.4, this feature was implemented on the Cisco ASR 1000 Series Routers. These commands were introduced or modified: **neighbor inter-as-hybrid**, **inter-as-hybrid**.

Feature Name	Release	Feature Information
MPLS VPN--Inter-AS Option AB+	15.0(1)SY	The MPLS VPN—Inter-AS Option AB+ feature addresses the scalability concerns of MPLS VPN—Inter-AS Option A by using a single BGP session to signal VPN prefixes (as described in Inter-AS Option B). In an Inter-AS AB+ deployment, the forwarding connections between the ASBRs are maintained on a per-VRF basis while the control plane information is exchanged by a single Multiprotocol BGP session. In Cisco IOS Release 15.0(1)SY, this feature was introduced. These commands were introduced or modified: **inter-as-hybrid**.

Glossary

ASBR -- Autonomous System Boundary router. A router that connects one autonomous system to another.

autonomous system --A collection of networks under a common administration sharing a common routing strategy.

BGP --Border Gateway Protocol. An interdomain routing protocol that exchanges network reachability information with other BGP systems (which may be within the same autonomous system or between multiple autonomous systems).

CE router--customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers do not recognize associated MPLS VPNs.

CSC --Carrier Supporting Carrier. A hierarchical VPN model that allows small service providers, or customer carriers, to interconnect their IP or MPLS networks over an MPLS backbone. This eliminates the need for customer carriers to build and maintain their own MPLS backbone.

eBGP --external Border Gateway Protocol. A BGP between routers located within different autonomous systems. When two routers, located in different autonomous systems, are more than one hop away from one another, the eBGP session between the two routers is considered a multihop BGP.

edge router--A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

iBGP --internal Border Gateway Protocol. A BGP between routers within the same autonomous system.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within a single autonomous system. Examples of common Internet IGP protocols include IGRP, OSPF, IS-IS, and RIP.

IP --Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

LFIB --Label Forwarding Information Base. Data structure used in MPLS to hold information about incoming and outgoing labels and associated Forwarding Equivalence Class (FEC) packets.

MP-BGP --Multiprotocol BGP.

MPLS --Multiprotocol Label Switching. The name of the IETF working group responsible for label switching, and the name of the label switching approach it has standardized.

NLRI --Network Layer Reachability Information. The BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and extended community values.

NSF --Nonstop forwarding enables routers to continuously forward IP packets following a Route Processor takeover or switchover to another Route Processor. NSF maintains and updates Layer 3 routing and forwarding information in the backup Route Processor to ensure that IP packets and routing protocol information are forwarded continuously during the switchover and route convergence process.

PE router--provider edge router. A router that is part of a service provider's network. It is connected to a customer edge (CE) router. All MPLS VPN processing occurs in the PE router.

QoS --quality of service. Measure of performance for a transmission system that indicates its transmission quality and service availability.

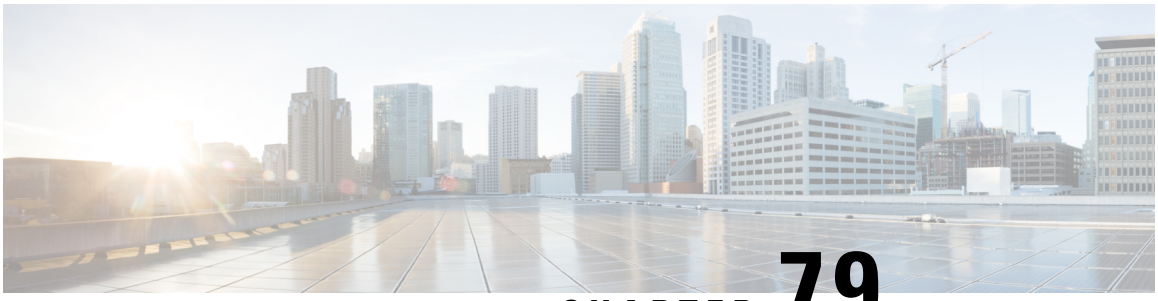
RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is imported.

SLA --Service Level Agreement given to VPN subscribers.

VPN --Virtual Private Network. A secure MPLS-based network that shares resources on one or more physical networks (typically implemented by one or more service providers). A VPN contains geographically dispersed sites that can communicate securely over a shared backbone network.

VRF --VPN routing and forwarding instance. Routing information that defines a VPN site that is attached to a PE router. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.



CHAPTER 79

MPLS VPN Carrier Supporting Carrier Using LDP and an IGP

Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. This module explains how to configure the MPLS VPN CSC network using MPLS Label Distribution Protocol (LDP) to distribute MPLS labels and an Interior Gateway Protocol (IGP) to distribute routes.

Prerequisites for MPLS VPN CSC with LDP and IGP

- The provider edge (PE) routers of the backbone carrier require 128 MB of memory.
- The backbone carrier must enable the PE router to check that the packets it receives from the customer edge (CE) router contain only the labels that the PE router advertised to the CE router. This prevents data spoofing, which occurs when a packet from an unrecognized IP address is sent to a router.

Restrictions for MPLS VPN CSC with LDP and IGP

The following features are not supported with this feature:

- ATM MPLS
- Carrier supporting carrier traffic engineering
- Carrier supporting carrier quality of service (QoS)
- RSVP aggregation

- VPN Multicast between the customer carrier and the backbone carrier network

The following router platforms are supported on the edge of the MPLS VPN:

- Cisco 7200 series
- Cisco 7500 series
- Cisco 12000 series

See the table below for Cisco 12000 series line card support added for Cisco IOS releases.

Table 150: Cisco12000 Series Line Card Support Added for Cisco IOS Releases

Type	Line Cards	Cisco IOS Release Added
Packet over SONET (POS)	4-Port OC-3 POS 1-Port OC-12 POS 8-Port OC-3 POS 16-Port OC-3 POS 4-Port OC-12 POS 1-Port OC-48 POS 4-Port OC-3 POS ISE 8-Port OC-3 POS ISE 16 x OC-3 POS ISE 4 Port OC-12 POS ISE 1-Port OC-48 POS ISE	12.0(16)ST 12.0(21)ST 12.0(22)S
Electrical Interface	6- Port DS3 12- Port DS3 6-Port E3	12.0(16)ST 12.0(21)ST
ATM	4-Port OC-3 ATM 1-Port OC12 ATM 4-Port OC-12 ATM	12.0(22)S
Channelized Interface	2-Port CHOC-3 6-Port Ch T3 (DS1) 1-Port CHOC-12 (DS3) 1-Port CHOC-12 (OC-3) 4-Port CHOC-12 ISE 1-Port CHOC-48 ISE	12.0(22)S

Information About MPLS VPN CSC with LDP and IGP

MPLS VPN CSC Introduction

Carrier supporting carrier is where one service provider allows another service provider to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

A backbone carrier offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services. The customer carrier can be either:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

Benefits of Implementing MPLS VPN CSC

The MPLS VPN CSC network provides the following benefits to service providers who are backbone carriers and to customer carriers.

Benefits to the Backbone Carrier

- The backbone carrier can accommodate many customer carriers and give them access to its backbone. The backbone carrier does not need to create and maintain separate backbones for its customer carriers. Using one backbone network to support multiple customer carriers simplifies the backbone carrier's VPN operations. The backbone carrier uses a consistent method for managing and maintaining the backbone network. This is also cheaper and more efficient than maintaining separate backbones.
- The MPLS VPN carrier supporting carrier feature is scalable. Carrier supporting carrier can change the VPN to meet changing bandwidth and connectivity needs. The feature can accommodate unplanned growth and changes. The carrier supporting carrier feature enables tens of thousands of VPNs to be set up over the same network, and it allows a service provider to offer both VPN and Internet services.
- The MPLS VPN carrier supporting carrier feature is a flexible solution. The backbone carrier can accommodate many types of customer carriers. The backbone carrier can accept customer carriers who are ISPs or VPN service providers or both. The backbone carrier can accommodate customer carriers that require security and various bandwidths.

Benefits to the Customer Carriers

- The MPLS VPN carrier supporting carrier feature removes from the customer carrier the burden of configuring, operating, and maintaining its own backbone. The customer carrier uses the backbone network of a backbone carrier, but the backbone carrier is responsible for network maintenance and operation.
- Customer carriers who use the VPN services provided by the backbone carrier receive the same level of security that Frame Relay or ATM-based VPNs provide. Customer carriers can also use IPSec in their VPNs for a higher level of security; it is completely transparent to the backbone carrier.

- Customer carriers can use any link layer technology (SONET, DSL, Frame Relay, and so on) to connect the CE routers to the PE routers and the PE routers to the P routers. The MPLS VPN carrier supporting carrier feature is link layer independent. The CE routers and PE routers use IP to communicate, and the backbone carrier uses MPLS.
- The customer carrier can use any addressing scheme and still be supported by a backbone carrier. The customer address space and routing information are independent of the address space and routing information of other customer carriers or the backbone provider.

Configuration Options for MPLS VPN CSC with LDP and IGP

The backbone carrier offers BGP and MPLS VPN services. The customer carrier can be one of the two types of service providers described in the following sections, which explain how the backbone and customer carriers distribute IPv4 routes and MPLS labels.

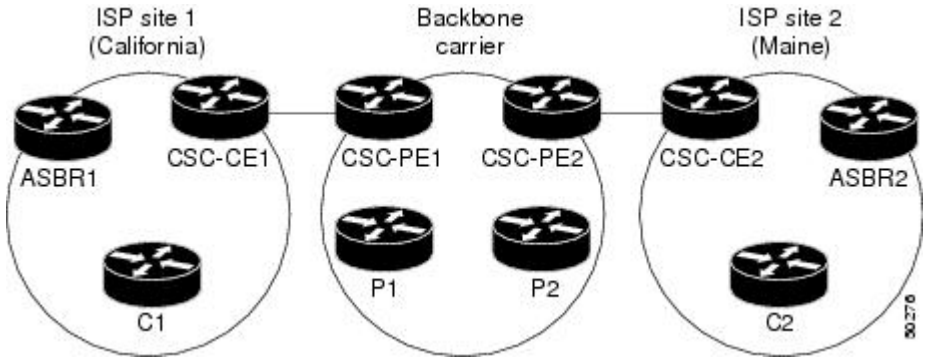
Customer Carrier Is an ISP

This section explains how a BGP/MPLS VPN service provider (backbone carrier) can provide a segment of its backbone network to a customer who is an ISP.

Consider the following example:

An ISP has two sites: one in California, the other in Maine. Each site is a point of presence (POP). The ISP wants to connect these sites using a VPN service provided by a backbone carrier. The figure below illustrates this situation.

Figure 125: Sample BGP/MPLS Backbone Carrier Supporting an ISP



Note The CE routers in the figures are CE routers to the backbone carrier. However, they are PE routers to the customer carrier.

In this example, only the backbone carrier uses MPLS. The customer carrier (ISP) uses only IP. As a result, the backbone carrier must carry all the Internet routes of the customer carrier, which could be as many as 100,000 routes. This poses a scalability problem for the backbone carrier. To solve the scalability problem, the backbone carrier is configured as follows:

- The backbone carrier allows only internal routes of the customer carrier (IGP routes) to be exchanged between the CE routers of the customer carrier and the PE routers of the backbone carrier.

- MPLS is enabled on the interface between the CE router of the customer carrier and the PE router of the backbone carrier.

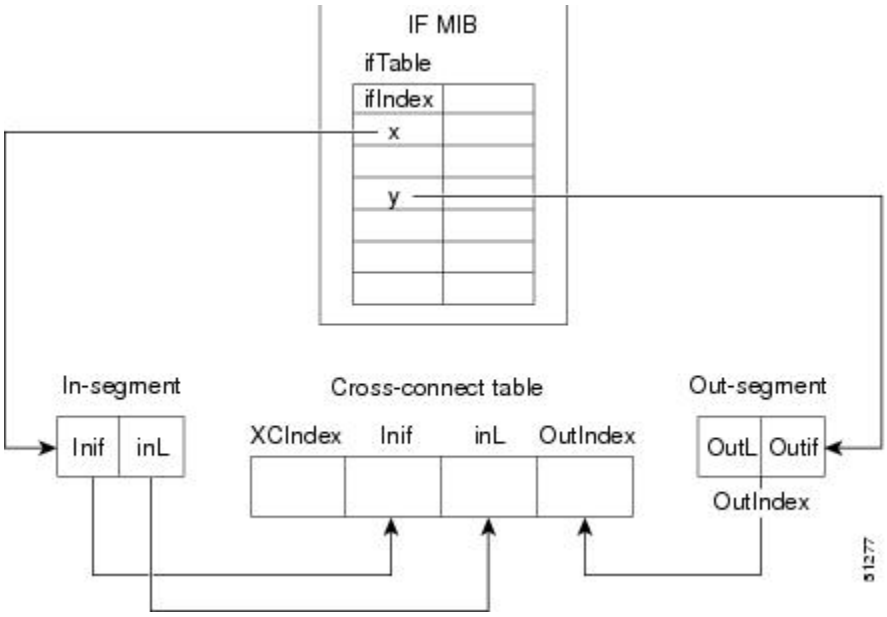
Internal and external routes are differentiated this way:

- Internal routes go to any of the routers within the ISP.
- External routes go to the Internet.

The number of internal routes is much lower than the number of external routes. Restricting the routes between the CE routers of the customer carrier and the PE routers of the backbone carrier significantly reduces the number of routes that the PE router needs to maintain.

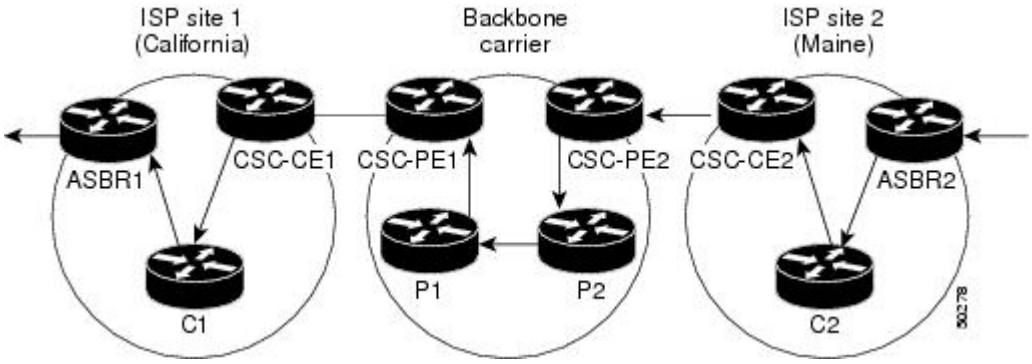
Because the PE routers do not have to carry external routes in the VRF routing table, they can use the incoming label in the packet to forward the customer carrier Internet traffic. Adding MPLS to the routers provides a consistent method of transporting packets from the customer carrier to the backbone carrier. MPLS allows the exchange of an MPLS label between the PE and the CE routers for every internal customer carrier route. The routers in the customer carrier have all the external routes either through internal Border Gateway Protocol (iBGP) or route redistribution to provide Internet connectivity. The figure below shows how information is exchanged when the network is configured in this manner.

Figure 126: Backbone Carrier Exchanging Routing Information with a Customer Carrier Who Is an ISP



In the figure below, routes are created between the backbone carrier and the customer carrier sites. ASBR2 receives an Internet route that originated outside the network. All routers in the ISP sites have all the external routes through IBGP connections among them.

Figure 127: Establishing a Route Between a Backbone Carrier and a Customer Carrier Who Is an ISP



The table below describes the process of establishing the route, which can be divided into two distinct steps:

- The backbone carrier propagates the IGP information of the customer carrier, which enables the customer carrier routers to reach all the customer carrier routers in the remote sites.
- Once the routers of the customer carriers in different sites are reachable, external routes can be propagated in the customer carrier sites, using IBGP without using the backbone carrier routers.

Table 151: Establishing a Route Between the Backbone Carrier and the Customer Carrier ISP

Step	Description
1	CSC-CE2 sends the internal routes within site 2 to CSC-PE2. The routes include the route to ASBR2.
2	CSC-PE2 sends the routing information for site 2 to CSC-PE1, using MPLS VPN processes. CSC-PE1 gets one label (called L3), which is associated with the route to the VPN-IP address for ASBR2. CSC-PE1 gets another label (called L2), which is associated with the route to CSC-PE2.
3	CSC-PE1 sends the routing information associated with internal routes from site 2 to CSC-CE1. CSC-PE1 also sends the label binding information. As a result, CSC-CE1 gets the route to ASBR2 with CSC-PE1 as the next hop. The label associated with that route is called L1.
4	CSC-CE1 distributes the routing information through site 1. Every router in site 1 gets a route for every internal destination in site 2. Therefore, every router in site 1 can reach routers in site 2 and learn external routes through IBGP.
5	ASBR2 receives an Internet route.
6	The IBGP sessions exchange the external routing information of the ISP, including a route to the Internet. Every router in site 1 knows a route to the Internet, with ASBR2 as the next hop of that route.

Customer Carrier Is a BGP MPLS VPN Service Provider

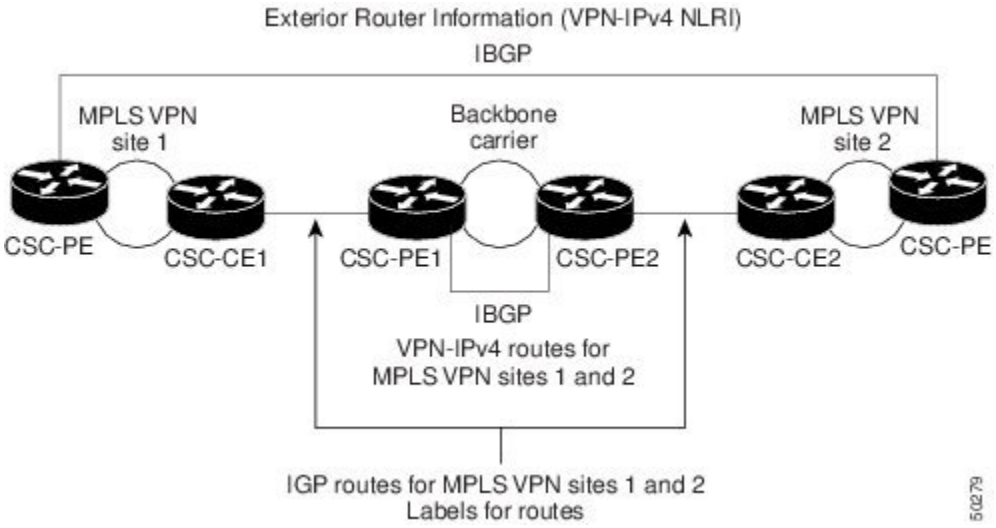
When a backbone carrier and the customer carrier both provide BGP/MPLS VPN services, the method of transporting data is different from when a customer carrier provides only ISP services. The following list highlights those differences:

- When a customer carrier provides BGP/MPLS VPN services, its external routes are VPN-IPv4 routes. When a customer carrier is an ISP, its external routes are IP routes.

- When a customer carrier provides BGP/MPLS VPN services, every site within the customer carrier must use MPLS. When a customer carrier is an ISP, the sites do not need to use MPLS.

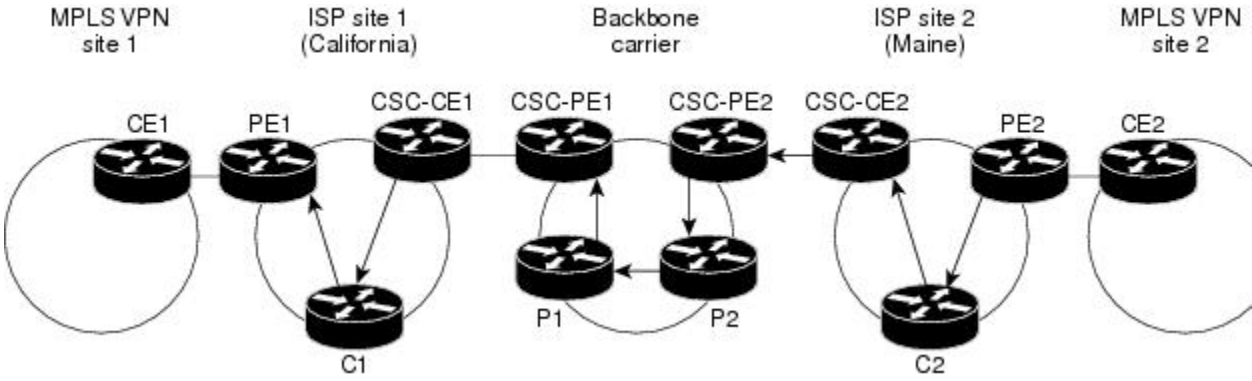
The figure below shows how information is exchanged when MPLS VPN services reside on all customer carrier sites and on the backbone carrier.

Figure 128: Backbone Carrier Exchanging Information with a Customer Carrier Who Is an MPLS VPN Service Provider



In the example shown in the figure below, routes are created between the backbone carrier and the customer carrier sites.

Figure 129: Establishing a Route Between a Backbone Carrier and a Customer Carrier Who Is an MPLS VPN Service Provider



The table below describes the process of establishing the route.

Table 152: Establishing a Route Between the Backbone Carrier and Customer Carrier Site

Step	Description
1	CE2 sends all the internal routes within site 2 to CSC-PE2.
2	CSC-PE2 sends the routing information for site 2 to CSC-PE1, using MPLS VPN processes. CSC-PE1 gets one label (called L3), which is associated with the route to the VPN-IP address for PE2. CSC-PE1 gets another label (called L2), which is associated with the route to CSC-PE2.

Step	Description
3	CSC-PE1 sends the routing information associated with internal routes from site 2 to CSC-CE1. CSC-PE1 also sends the label binding information. As a result, CSC-CE1 gets the route to PE2 with CSC-PE1 as the next hop. The label associated with that route is called L1.
4	CE1 distributes the routing and labeling information through site 1. Every router in site 1 gets a route for every internal destination in site 2. Therefore, PE1 can establish an MP-IBGP session with PE2.
5	CE2 advertises the internal routes of MPLS VPN site 2 to PE2.
6	PE2 allocates labels for all the VPN routes (regular MPLS VPN functionality) and advertises the labels to PE1, using MP-IBGP.
7	PE1 can forward traffic from VPN site 1 that is destined for VPN site 2.

How to Configure MPLS VPN CSC with LDP and IGP

Configuring the Backbone Carrier Core

Configuring the backbone carrier core requires configuring connectivity and routing functions for the CSC core and the CSC-PE routers.

Configuring and verifying the CSC core (backbone carrier) involves the following tasks:

Prerequisites

Before you configure a backbone carrier core, configure the following on the CSC core routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on. For information, see Configuring a Basic BGP Network, Configuring OSPF, Configuring a Basic IS-IS Network, and Configuring EIGRP.
- Label Distribution Protocol (LDP). For information, see MPLS Label Distribution Protocol.

Verifying IP Connectivity and LDP Configuration in the CSC Core

Perform this task to verify IP connectivity and LDP configuration in the CSC core.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
5. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**]
6. **show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**]
7. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
8. **show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] |**all**]

9. **show ip route**
10. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name* | *system-address*} **Example:** `Router# ping ip 10.0.0.1`	(Optional) Diagnoses basic network connectivity on AppleTalk, Connectionless Network Service (CLNS), IP, Novell, Apollo, VINES, DECnet, or Xerox Network System (XNS) networks. • Use the **ping ip** command to verify the connectivity from one CSC core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.0.0.1`	(Optional) Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to verify the path that a packet goes through before reaching the final destination. The **trace** command can help isolate a trouble spot if two routers cannot communicate.
Step 4	**show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Router# show mpls forwarding-table`	(Optional) Displays the contents of the MPLS label forwarding information base (LFIB). • Use the **show mpls forwarding-table** command to verify that MPLS packets are being forwarded.
Step 5	**show mpls ldp discovery** [**vrf** *vrf-name* | **all**] **Example:** `Router# show mpls ldp discovery`	(Optional) Displays the status of the LDP discovery process. • Use the **show mpls ldp discovery** command to verify that LDP is operational in the CSC core.
Step 6	**show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**] **Example:** `Router# show mpls ldp neighbor`	(Optional) Displays the status of LDP sessions. • Use the**show mpls ldp neighbor** command to verify LDP configuration in the CSC core.
Step 7	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:**	(Optional) Displays entries in the forwarding Information Base (FIB). • Use the **show ip cef** command to check the forwarding table (prefixes, next hops, and interfaces).

	Command or Action	Purpose
	`Router# show ip cef`	
Step 8	**show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] |**all**] **Example:** `Router# show mpls interfaces`	(Optional) Displays information about one or more or all interfaces that are configured for label switching. • Use the**show mpls interfaces** command to verify that the interfaces are configured to use LDP.
Step 9	**show ip route** **Example:** `Router# show ip route`	(Optional) Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, and interface.
Step 10	**disable** **Example:** `Router# disable`	(Optional) Returns to privileged EXEC mode.

Troubleshooting Tips

You can use the **ping** and **trace** commands to verify complete MPLS connectivity in the core. You also get useful troubleshooting information from the additional **show** commands.

Configuring VRFs for CSC-PE Routers

Perform this task to configure VPN routing and forwarding (VRF) instances for the backbone carrier edge (CSC-PE) routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **exit**
8. **interface** *type number*
9. **ip vrf forwarding** *vrf-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	Example:
	`Router(config-vrf)# rd 100:1`
		Creates routing and forwarding tables.
		• The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN-IPv4 prefix. You can enter an RD in either of these formats:
		  • 16-bit AS number: your 32-bit number, for example, 101:3
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	(Optional) Exits to global configuration mode.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	Specifies the interface to configure and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip vrf detail** command and make sure the MPLS VPN is up and associated with the right interfaces.

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier

Perform this task to configure Multiprotocol BGP (MP-BGP) connectivity in the backbone carrier.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type*
7. **address-family vpnv4** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
9. **neighbor** {*ip-address* | *peer-group-name*} **activate**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	(Optional) Disables the IPv4 unicast address family on all neighbors. • Use the **no bgp default-unicast** command if you are using this neighbor for MPLS routes only.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.5.5.5 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type* **Example:** `Router(config-router)# neighbor 10.2.0.0 update-source loopback0`	Allows BGP sessions to use a specific operational interface for TCP connections. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *interface-type* argument specifies the interface to be used as the source.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.

	Command or Action	Purpose
		• The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.4.0.0 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 10	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command generates an error message, enter a **debug ip bgp** *x.x.x.x* **events** command, where *x.x.x.x* is the IP address of the neighbor.

Configuring the CSC-PE and CSC-CE Routers

To enable the CSC-PE and CSC-CE routers to distribute routes and MPLS labels, perform the following tasks:

Prerequisites

Before you configure the CSC-PE and CSC-CE routers, you must configure an IGP on the CSC-PE and CSC-CE routers. A routing protocol is required between the PE and CE routers that connect the backbone carrier to the customer carrier. The routing protocol enables the customer carrier to exchange IGP routing information with the backbone carrier. Use the same routing protocol that the customer carrier uses. You can choose RIP, OSPF, or static routing as the routing protocol. BGP is not supported. For the configuration steps, see Configuring MPLS Layer 3 VPNs .

Configuring LDP on the CSC-PE and CSC-CE Routers

MPLS LDP is required between the PE and CE routers that connect the backbone carrier to the customer carrier. You can configure LDP as the default label distribution protocol for the entire router or just for the PE-to-CE interface for VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **mpls label protocol ldp**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Specifies MPLS LDP as the default label distribution protocol for the router.
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	(Optional) Specifies the interface to configure and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 5	**mpls label protocol ldp** **Example:** `Router(config-if)# mpls label protocol ldp`	(Optional) Specifies MPLS LDP as the default label distribution protocol for the interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits to privileged EXEC mode.

Enabling MPLS Encapsulation on the CSC-PE and CSC-CE Routers

Every packet that crosses the backbone carrier must be encapsulated, so that the packet includes MPLS labels. You can enable MPLS encapsulation for the entire router or just on the interface of the PE or CE router. To enable the encapsulation of packets, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **interface** *type number*
5. **mpls ip**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Router(config)# mpls ip`	Enables MPLS encapsulation for the router.
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	(Optional) Specifies the interface to configure and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 5	**mpls ip** **Example:** `Router(config-if)# mpls ip`	(Optional) Enables MPLS encapsulation for the specified interface.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits to privileged EXEC mode.

Verifying the Carrier Supporting Carrier Configuration

The following commands verify the status of LDP sessions that were configured between the backbone carrier and customer carrier. Now the customer carrier ISP sites appear as a VPN customer to the backbone carrier.

SUMMARY STEPS

1. **show mpls ldp discovery vrf** *vrf-name*
2. **show mpls ldp discovery all**

DETAILED STEPS

Step 1 **show mpls ldp discovery vrf** *vrf-name*

Use this command to show that the LDP sessions are in VRF VPN1 of the PE router of the backbone carrier, for example:

Example:

```
Router# show mpls ldp discovery vrf vpn1
Local LDP Identifier:
    10.0.0.0:0
Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            LDP Id: 10.0.0.1:0
POS6/0 (ldp): xmit
```

Step 2 **show mpls ldp discovery all**

Use this command to list all LDP sessions in a router, for example:

Example:

```
Router# show mpls ldp discovery all
Local LDP Identifier:
    10.10.10.10:0
Discovery Sources:
    Interfaces:
        Ethernet1/5 (ldp): xmit/recv
            LDP Id: 10.5.5.5:0
VRF vpn1: Local LDP Identifier:
    10.0.0.1:0
Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            LDP Id: 10.0.0.1:0
POS6/0 (ldp): xmit
```

The Local LDP Identifier field shows the LDP identifier for the local label switching router for this session. The Interfaces field displays the interfaces engaging in LDP discovery activity:

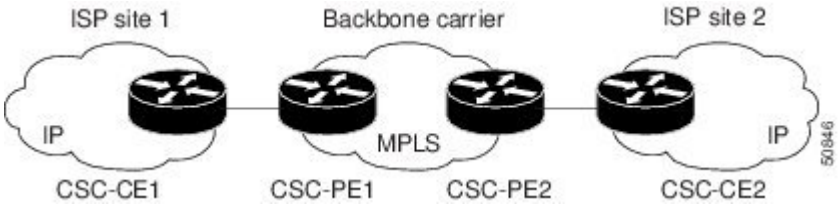
- xmit indicates that the interface is transmitting LDP discovery hello packets.
- recv indicates that the interface is receiving LDP discovery hello packets.

Configuration Examples for MPLS VPN CSC with LDP and IGP

MPLS VPN CSC Network with a Customer Who Is an ISP Example

The figure below shows a carrier supporting carrier network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a POP. The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS. The ISP sites use IP. To enable packet transfer between the ISP sites and the backbone carrier, the CE routers that connect the ISPs to the backbone carrier run MPLS.

Figure 130: Carrier Supporting Carrier Network with a Customer Carrier Who Is an ISP



The following examples show the configuration of each router in the carrier supporting carrier network. OSPF is used to connect the customer carrier to the backbone carrier.

CSC-CE1 Configuration

```
mpls label protocol ldp
!
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
```

```
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM2/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.14.14.14 0.0.0.0 area 200
 network 10.15.0.0 0.255.255.255 area 200
 network 10.16.0.0 0.255.255.255 area 200
```

CSC-PE1 Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.11.11.11 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.19.19.19 255.255.255.255
 no ip directed-broadcast
!
interface ATM1/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1/0.1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
```

```
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.11.11.11 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 10.19.19.19 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.12.12.12 remote-as 100
 neighbor 10.12.12.12 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-PE2 Configuration

```
ip cef distributed
!
ip vrf vpn1
```

```
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.20.20.20 255.255.255.255
 no ip directed-broadcast
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.12.12.12 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
```

```
 network 10.20.20.20 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.11.11.11 remote-as 100
 neighbor 10.11.11.11 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-CE2 Configuration

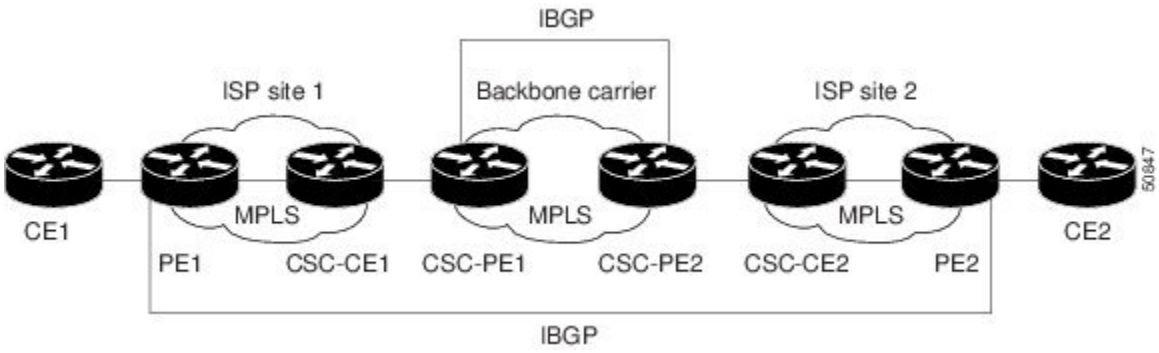
```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
```

```
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.16.16.16 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
 network 10.0.0.0 0.255.255.255 area 200
```

MPLS VPN CSC Network with a Customer Who Is an MPLS VPN Provider Example

The figure below shows a carrier supporting carrier network configuration where the customer carrier is an MPLS VPN provider. The customer carrier has two sites. The backbone carrier and the customer carrier use MPLS. The IBGP sessions exchange the external routing information of the ISP.

Figure 131: Carrier Supporting Carrier Network with a Customer Carrier Who Is an MPLS VPN Provider



The following configuration examples show the configuration of each router in the carrier supporting carrier network. OSPF is the protocol used to connect the customer carrier to the backbone carrier.

CE1 Configuration

```
ip cef
!
interface Loopback0
 ip address 10.17.17.17 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
router ospf 300
 log-adjacency-changes
 redistribute bgp 300 subnets
 passive-interface Ethernet0/1
```

```
 network 10.17.17.17 0.0.0.0 area 300
!
router bgp 300
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected
 redistribute ospf 300 match internal external 1 external 2
 neighbor 10.0.0.1 remote-as 200
 neighbor 10.0.0.1 advertisement-interval 5
 no auto-summary
```

PE1 Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address 10.13.13.13 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/0
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 passive-interface Ethernet3/0
 network 10.13.13.13 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.15.15.15 remote-as 200
```

```
 neighbor 10.15.15.15 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 neighbor 10.0.0.2 advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-CE1 Configuration

```
mpls label protocol ldp
!
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM2/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
```

```
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.14.14.14 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
 network 10.0.0.0 0.255.255.255 area 200
```

CSC-PE1 Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.19.19.19 255.255.255.255
 no ip directed-broadcast
!
interface ATM1/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
```

```
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.11.11.11 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 10.19.19.19 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.12.12.12 remote-as 100
 neighbor 10.12.12.12 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-PE2 Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.20.20.20 255.255.255.255
 no ip directed-broadcast
!
interface ATM0/1/0
 no ip address
```

```
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.12.12.12 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 10.20.20.20 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.11.11.11 remote-as 100
 neighbor 10.11.11.11 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 exit-address-family
```

```
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-CE2 Configuration

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.16.16.16 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
 network 10.0.0.0 0.255.255.255 area 200
```

PE2 Configuration

```
ip cef
ip cef accounting non-recursive
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address 10.15.15.15 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet3/0
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 passive-interface Ethernet3/0
 network 10.15.15.15 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.13.13.13 remote-as 200
 neighbor 10.13.13.13 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
```

```
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 neighbor 10.0.0.2 advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

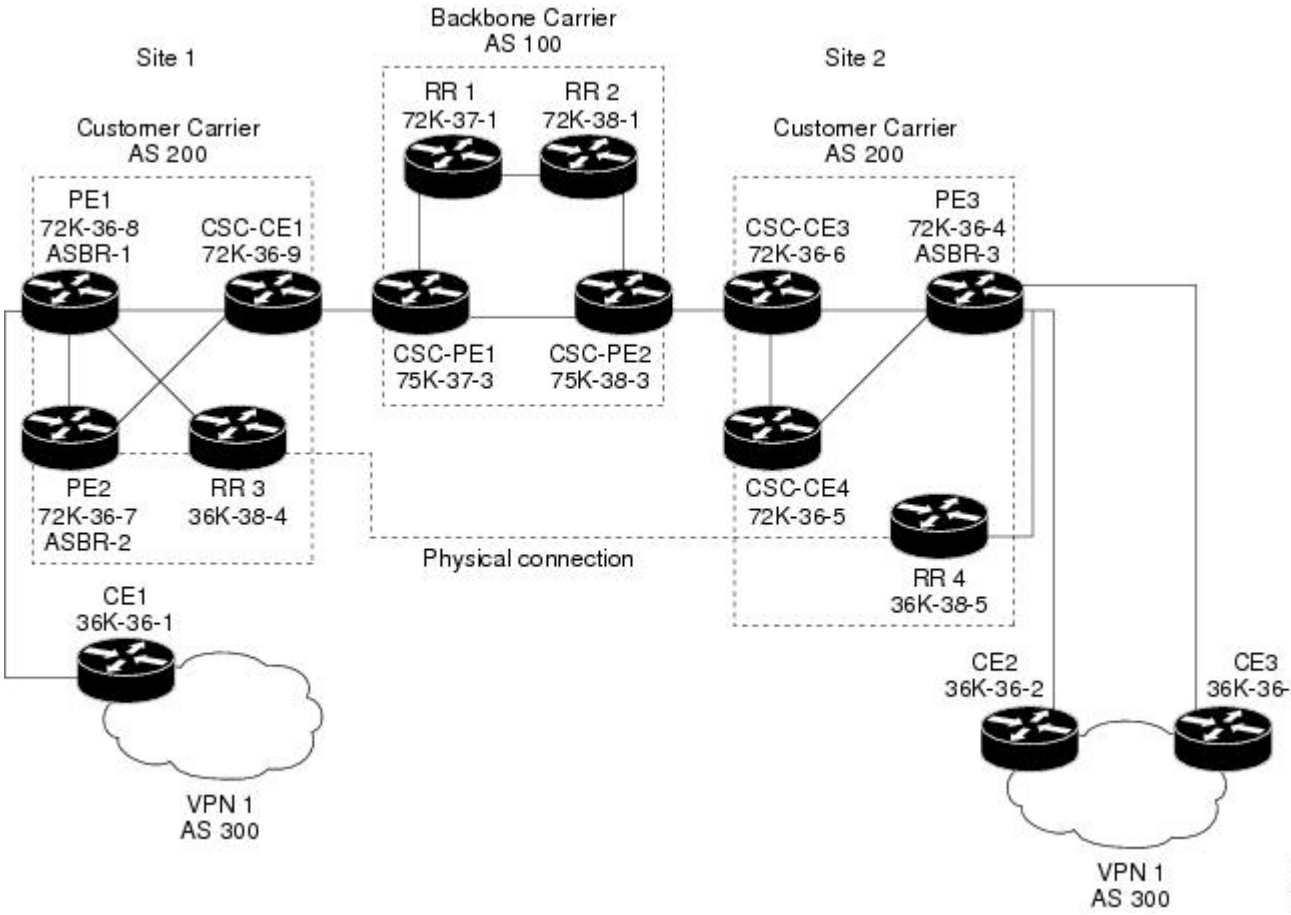
CE2 Configuration

```
ip cef
!
interface Loopback0
 ip address 10.18.18.18 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
router ospf 300
 log-adjacency-changes
 redistribute bgp 300 subnets
 passive-interface Ethernet0/1
 network 10.18.18.18 0.0.0.0 area 300
!
router bgp 300
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected
 redistribute ospf 300 match internal external 1 external 2
 neighbor 10.0.0.1 remote-as 200
 neighbor 10.0.0.1 advertisement-interval 5
 no auto-summary
```

MPLS VPN CSC Network That Contains Route Reflectors Example

The figure below shows a carrier supporting carrier network configuration that contains route reflectors. The customer carrier has two sites.

Figure 132: Carrier Supporting Carrier Network that Contains Route Reflectors



Note A connection between route reflectors (RRs) is not necessary.

The following configuration examples show the configuration of each router in the carrier supporting carrier network. Note the following:

- The router IP addresses are abbreviated for ease of reading. For example, the loopback address for PE 1 is 25, which is equivalent to 10.25.25.25.
- The following list shows the loopback addresses for the CSC-PE routers:
  - CSC-PE1 (75K-37-3): loopback 0 = 10.15.15.15, loopback 1 = 10.18.18.18
  - CSC-PE2 (75K-38-3): loopback 0 = 10.16.16.16, loopback 1 = 10.20.20.20

Backbone Carrier Configuration

Route Reflector 1 (72K-37-1) Configuration

```
interface Loopback0
 ip address 10.13.13.13 255.255.255.255
```

```
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 mpls
ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM1/1
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1.1 mpls
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
 network 10.2.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 bgp cluster-id 1
 redistribute static
 neighbor 10.15.15.15 remote-as 100
 neighbor 10.15.15.15 update-source Loopback0
 neighbor 10.16.16.16 remote-as 100
 neighbor 10.16.16.16 update-source Loopback0
 !
 address-family ipv4 vrf vpn1
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 route-reflector-client
 neighbor 10.15.15.15 send-community extended
 neighbor 10.16.16.16 activate
 neighbor 10.16.16.16 route-reflector-client
 neighbor 10.16.16.16 send-community extended
 bgp scan-time import 5
 exit-address-family
```

Route Reflector 2 (72K-38-1) Configuration

```
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 mpls
 ip address 10.0.0.1 255.0.0.0
no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM1/1
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1.1 mpls
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0 0.255.255.255 area 100
 network 10.2.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 bgp cluster-id 1
 redistribute static
 neighbor 10.15.15.15 remote-as 100
 neighbor 10.15.15.15 update-source Loopback0
 neighbor 10.16.16.16 remote-as 100
 neighbor 10.16.16.16 update-source Loopback0
 !
 address-family ipv4 vrf vpn1
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 route-reflector-client
 neighbor 10.15.15.15 send-community extended
 neighbor 10.16.16.16 activate
```

```
 neighbor 10.16.16.16 route-reflector-client
 neighbor 10.16.16.16 send-community extended
 bgp scan-time import 5
 exit-address-family
```

CSC-PE1 (75K-37-3) Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
interface Loopback0
 ip address 10.15.15.15 255.255.255.255
 no ip directed-broadcast
!
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.18.18.18 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/0/1
 ip vrf forwarding vpn1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip route-cache distributed
 mpls label protocol ldp
 mpls ip
!
interface ATM1/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1/0.1 mpls
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
 no atm enable-ilmi-trap
```

```
 mpls label protocol ldp
 mpls ip
!
interface ATM3/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/1/0.1 mpls
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
 network 10.2.0.0 0.255.255.255 area 100
 network 10.3.0.0 0.255.255.255 area 100
 network 10.4.0.0 0.255.255.255 area 100
!
router ospf 1 vrf vpn1
 redistribute bgp 100 metric-type 1 subnets
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.13.13.13 remote-as 100
 neighbor 10.13.13.13 update-source Loopback0
 neighbor 10.14.14.14 remote-as 100
 neighbor 10.14.14.14 update-source Loopback0
 !
 address-family ipv4
 redistribute static
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 neighbor 10.14.14.14 activate
 neighbor 10.14.14.14 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 1 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-PE2 (75K-38-3) Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
!
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.20.20.20 255.255.255.255
 no ip directed-broadcast
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 mpls
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM2/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/1/0.1 mpls
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
```

```
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/1/0
 no ip address
 no ip directed-broadcast
no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/1/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 6 33 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 1 vrf vpn1
 redistribute bgp 100 metric-type 1 subnets
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.13.13.13 remote-as 100
 neighbor 10.13.13.13 update-source Loopback0
 neighbor 10.14.14.14 remote-as 100
 neighbor 10.14.14.14 update-source Loopback0
 !
 address-family ipv4
 redistribute static
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 neighbor 10.14.14.14 activate
 neighbor 10.14.14.14 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 1 match internal external 1 external 2
 no auto-summary
```

```
 no synchronization
 exit-address-family
```

Customer Carrier Site 1 Configuration

PE1 (72K-36-8) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
no mpls ip propagate-ttl
!
interface Loopback0
 ip address 10.25.25.25 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/0
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 200
```

```
 neighbor 10.22.22.22 remote-as 200
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 redistribute connected
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

CSC-CE1 (72K-36-9) Configuration

```
ip cef
no ip domain-lookup
!
interface Loopback0
 ip address 10.11.11.11 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
mpls label protocol ldp
mpls ip
!
interface ATM2/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
mpls label protocol ldp
mpls ip
!
interface Ethernet3/0
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
```

```
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
```

PE2 (72K-36-7) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
no mpls ip propagate-ttl
!
interface Loopback0
 ip address 10.24.24.24 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet3/0
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/1
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet3/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/3
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
```

```
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 200
 neighbor 10.22.22.22 remote-as 200
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

Route Reflector 3 (36K-38-4) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.23.23.23 255.255.255.255
!
interface Ethernet1/1
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
interface Ethernet1/2
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0
 no ip address
 no ip mroute-cache
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM3/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 atm pvc 100 0 55 aal5snap
 mpls label protocol ldp
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
!
router bgp 200
```

```
 no synchronization
 no bgp default ipv4-unicast
 bgp cluster-id 2
 redistribute static
 neighbor 10.21.21.21 remote-as 200
 neighbor 10.21.21.21 update-source Loopback0
 neighbor 10.24.24.24 remote-as 200
 neighbor 10.24.24.24 update-source Loopback0
 neighbor 10.25.25.25 remote-as 200
 neighbor 10.25.25.25 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.21.21.21 activate
 neighbor 10.21.21.21 route-reflector-client
 neighbor 10.21.21.21 send-community extended
 neighbor 10.24.24.24 activate
 neighbor 10.24.24.24 route-reflector-client
 neighbor 10.24.24.24 send-community extended
 neighbor 10.25.25.25 activate
 neighbor 10.25.25.25 route-reflector-client
 neighbor 10.25.25.25 send-community extended
 exit-address-family
```

CE1 (36K-36-1) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.28.28.28 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet0/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
router bgp 300
 network 10.0.0.0
 network 10.0.0.0
 network 10.0.0.0
 neighbor 10.0.0.1 remote-as 200
 neighbor 10.0.0.1 remote-as 200
```

Customer Carrier Site 2 Configuration

CSC-CE3 (72K-36-6) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
```

```
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
mpls label protocol ldp
 mpls ip
!
interface POS2/0
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
mpls label protocol ldp
mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 40 aal5snap
 mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
```

PE3 (72K-36-4) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
!
!
interface Loopback0
 ip address 10.21.21.21 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet3/0
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet3/1
 ip vrf forwarding vpn2
```

```
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet3/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
mpls label protocol ldp
mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 40 aal5snap
mpls label protocol ldp
mpls ip
!
interface ATM6/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 20 aal5snap
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
!
router bgp 200
 neighbor 10.22.22.22 remote-as 200
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 redistribute connected
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

CSC-CE4 (72K-36-5) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
 no ip directed-broadcast
!
interface POS4/0
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
mpls label protocol ldp
mpls ip
 clock source internal
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 20 aal5snap
mpls label protocol ldp
mpls ip
!
interface ATM6/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 33 aal5snap
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
```

Route Reflector 4 (36K-38-5) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.22.22.22 255.255.255.255
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
interface ATM2/0
 no ip address
```

```
 no ip mroute-cache
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 atm pvc 100 0 55 aal5snap
 mpls label protocol ldp
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
!
router bgp 200
 no synchronization
 no bgp default ipv4-unicast
 bgp cluster-id 2
 redistribute static
 neighbor 10.21.21.21 remote-as 200
 neighbor 10.21.21.21 update-source Loopback0
 neighbor 10.24.24.24 remote-as 200
 neighbor 10.24.24.24 update-source Loopback0
 neighbor 10.25.25.25 remote-as 200
 neighbor 10.25.25.25 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.21.21.21 activate
 neighbor 10.21.21.21 route-reflector-client
 neighbor 10.21.21.21 send-community extended
 neighbor 10.24.24.24 activate
 neighbor 10.24.24.24 route-reflector-client
 neighbor 10.24.24.24 send-community extended
 neighbor 10.25.25.25 activate
 neighbor 10.25.25.25 route-reflector-client
 neighbor 10.25.25.25 send-community extended
 exit-address-family
```

CE2 (36K-36-2) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.26.26.26 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet0/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
router ospf 300
```

```
 redistribute bgp 300
 network 10.0.0.0 0.255.255.255 area 300
 network 10.0.0.0 0.255.255.255 area 300
!
router bgp 300
 network 10.0.0.0
 network 10.1.0.0
 network 10.2.0.0
 neighbor 10.0.0.1 remote-as 200
```

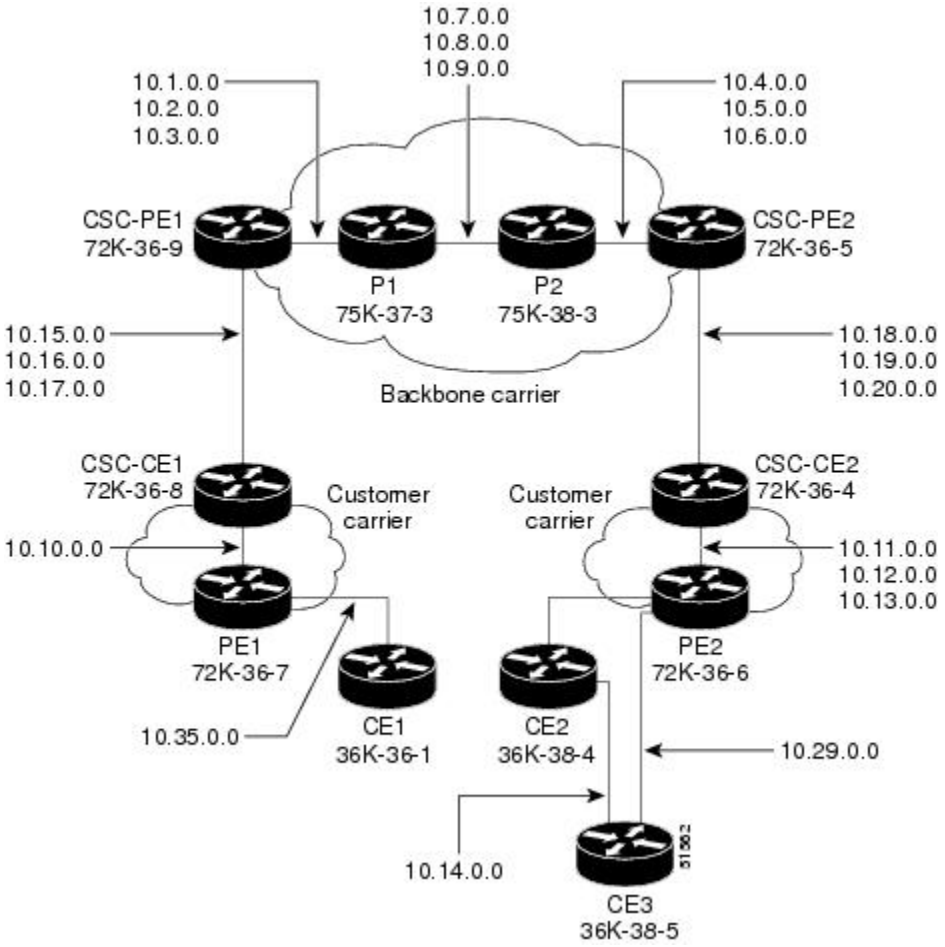
CE3 (36K-36-3) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.27.27.27 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet1/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet1/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
router ospf 300
 redistribute bgp 300
 network 10.0.0.0 0.255.255.255 area 300
 network 10.0.0.0 0.255.255.255 area 300
!
router bgp 300
 network 10.0.0.0
 network 10.1.0.0
 network 10.2.0.0
 neighbor 10.0.0.1 remote-as 200
```

MPLS VPN CSC Network with a Customer Who Has VPNs at the Network Edge Example

The figure below shows a carrier supporting carrier network configuration where the customer carrier has VPNs at the network edge.

Figure 133: Carrier Supporting Carrier Network



Backbone Carrier Configuration

CSC-PE1 (72K-36-9) Configuration

```
ip cef
!
ip vrf vpn1
rd 100:0
route-target export 100:0
route-target import 100:0
mpls label protocol ldp
!
!
interface Loopback0
ip address 10.14.14.14 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Loopback100
ip vrf forwarding vpn1
ip address 10.22.22.22 255.255.255.255
```

```
no ip directed-broadcast
!
interface ATM1/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
ip address 10.1.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.2 point-to-point
ip address 10.2.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.3 point-to-point
ip address 10.3.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM2/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
ip vrf forwarding vpn1
ip address 10.15.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM2/0.2 point-to-point
ip vrf forwarding vpn1
ip address 10.16.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM2/0.3 point-to-point
ip vrf forwarding vpn1
ip address 10.17.0.2 255.255.0.0
no ip directed-broadcast
```

```
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 100
log-adjacency-changes
redistribute connected subnets
passive-interface ATM2/0.1
passive-interface ATM2/0.2
passive-interface ATM2/0.3
passive-interface Loopback100
network 10.14.14.14 0.0.0.0 area 100
network 10.1.0.0 0.0.255.255 area 100
network 10.2.0.0 0.0.255.255 area 100
network 10.3.0.0 0.0.255.255 area 100
!
router ospf 200 vrf vpn1
log-adjacency-changes
redistribute connected subnets
redistribute bgp 100 metric-type 1 subnets
network 10.22.22.22 0.0.0.0 area 200
network 10.15.0.0 0.0.255.255 area 200
network 10.16.0.0 0.0.255.255 area 200
network 10.17.0.0 0.0.255.255 area 200
!
router bgp 100
bgp log-neighbor-changes
timers bgp 10 30
neighbor 10.11.11.11 remote-as 100
neighbor 10.11.11.11 update-source Loopback0
!
address-family ipv4
neighbor 10.11.11.11 activate
neighbor 10.11.11.11 send-community extended
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 10.11.11.11 activate
neighbor 10.11.11.11 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1
redistribute ospf 200 match internal external 1 external 2
no auto-summary
no synchronization
exit-address-family
```

P1 (75K-37-3) Configuration

```
ip cef distributed
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.12.12.12 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface ATM1/1/0
no ip address
```

```
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM1/1/0.1 point-to-point
ip address 10.7.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 103 0 53 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/1/0.2 point-to-point
ip address 10.8.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 104 0 54 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/1/0.3 point-to-point
ip address 10.9.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 105 0 55 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/0/0
no ip address
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
ip address 10.1.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
mpls accounting experimental input
tag-switching ip
!
interface ATM3/0/0.2 point-to-point
ip address 10.2.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/0/0.3 point-to-point
ip address 10.3.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
```

```
router ospf 100
log-adjacency-changes
redistribute connected subnets
network 10.12.12.12 0.0.0.0 area 100
network 10.1.0.0 0.0.255.255 area 100
network 10.2.0.0 0.0.255.255 area 100
network 10.3.0.0 0.0.255.255 area 100
network 10.7.0.0 0.0.255.255 area 100
network 10.8.0.0 0.0.255.255 area 100
network 10.9.0.0 0.0.255.255 area 100
```

P2 (75K-38-3) Configuration

```
ip cef distributed
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.13.13.13 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface ATM0/1/0
no ip address
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM0/1/0.1 point-to-point
ip address 10.7.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 103 0 53 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM0/1/0.2 point-to-point
ip address 10.8.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 104 0 54 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM0/1/0.3 point-to-point
ip address 10.9.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 105 0 55 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/1/0
no ip address
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
```

```
no atm ilmi-keepalive
!
interface ATM3/1/0.1 point-to-point
ip address 10.4.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/1/0.2 point-to-point
ip address 10.5.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/1/0.3 point-to-point
ip address 10.6.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 100
log-adjacency-changes
redistribute connected subnets
network 10.13.13.13 0.0.0.0 area 100
network 10.4.0.0 0.0.255.255 area 100
network 10.5.0.0 0.0.255.255 area 100
network 10.6.0.0 0.0.255.255 area 100
network 10.7.0.0 0.0.255.255 area 100
network 10.8.0.0 0.0.255.255 area 100
network 10.9.0.0 0.0.255.255 area 100
!
```

CSC-PE2 (72K-36-5) Configuration

```
ip cef
!
ip vrf vpn1
rd 100:0
route-target export 100:0
route-target import 100:0
mpls label protocol ldp
!
interface Loopback0
ip address 10.11.11.11 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Loopback100
ip vrf forwarding vpn1
ip address 10.23.23.23 255.255.255.255
no ip directed-broadcast
!
interface ATM5/0
no ip address
no ip directed-broadcast
no ip mroute-cache
```

```
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
ip vrf forwarding vpn1
ip address 10.18.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.2 point-to-point
ip vrf forwarding vpn1
ip address 10.19.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.3 point-to-point
ip vrf forwarding vpn1
ip address 10.20.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
ip address 10.4.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.2 point-to-point
ip address 10.5.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.3 point-to-point
ip address 10.6.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
```

```
!
router ospf 100
log-adjacency-changes
redistribute connected subnets
passive-interface ATM5/0.1
passive-interface ATM5/0.2
passive-interface ATM5/0.3
passive-interface Loopback100
network 10.11.11.11 0.0.0.0 area 100
network 10.4.0.0 0.0.255.255 area 100
network 10.5.0.0 0.0.255.255 area 100
network 10.6.0.0 0.0.255.255 area 100
!
router ospf 200 vrf vpn1
log-adjacency-changes
redistribute connected subnets
redistribute bgp 100 metric-type 1 subnets
network 10.23.23.23 0.0.0.0 area 200
network 10.18.0.0 0.0.255.255 area 200
network 10.19.0.0 0.0.255.255 area 200
network 10.20.0.0 0.0.255.255 area 200
!
router bgp 100
bgp log-neighbor-changes
timers bgp 10 30
neighbor 10.14.14.14 remote-as 100
neighbor 10.14.14.14 update-source Loopback0
!
address-family ipv4
neighbor 10.14.14.14 activate
neighbor 10.14.14.14 send-community extended
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 10.14.14.14 activate
neighbor 10.14.14.14 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1
redistribute ospf 200 match internal external 1 external 2
no auto-summary
no synchronization
exit-address-family
```

Customer Carrier Site 1 Configuration

CSC-CE1 (72K-36-8) Configuration

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.15.15.15 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface ATM1/0
no ip address
no ip directed-broadcast
```

```
no ip mroute-cache
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
ip address 10.15.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.2 point-to-point
ip address 10.16.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.3 point-to-point
ip address 10.17.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface Ethernet3/1
ip address 10.10.0.2 255.255.0.0
no ip directed-broadcast
no ip mroute-cache
mpls label protocol ldp
tag-switching ip
!
router ospf 200
log-adjacency-changes
redistribute connected subnets
network 10.15.15.15 0.0.0.0 area 200
network 10.10.0.0 0.0.255.255 area 200
network 10.15.0.0 0.0.255.255 area 200
network 10.16.0.0 0.0.255.255 area 200
network 10.17.0.0 0.0.255.255 area 200
```

PE2 (72K-36-7) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
no mpls ip propagate-ttl
!
interface Loopback0
 ip address 10.24.24.24 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet3/0
```

```
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/1
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet3/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/3
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 200
 neighbor 10.22.22.22 remote-as 200
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

CE1 (36K-36-1) Configuration

```
ip cef
!
interface Loopback0
ip address 10.19.19.19 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0/2
ip address 30.35.0.1 255.255.0.0
```

```
no ip directed-broadcast
!
router ospf 300
log-adjacency-changes
redistribute connected subnets
redistribute bgp 300 subnets
passive-interface Ethernet0/2
network 10.19.19.19 0.0.0.0 area 300
!
router bgp 300
no synchronization
bgp log-neighbor-changes
timers bgp 10 30
redistribute connected
redistribute ospf 300 match internal external 1 external 2
neighbor 10.35.0.2 remote-as 200
neighbor 10.35.0.2 advertisement-interval 5
no auto-summary
```

Customer Carrier Site 2 Configuration

CSC-CE2 (72K-36-4) Configuration

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.17.17.17 255.255.255.255
no ip directed-broadcast
!
interface ATM5/0
no ip address
no ip directed-broadcast
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
ip address 10.11.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.2 point-to-point
ip address 10.12.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.3 point-to-point
ip address 10.13.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
```

```
!
interface ATM6/0
no ip address
no ip directed-broadcast
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
ip address 10.18.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.2 point-to-point
ip address 10.19.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.3 point-to-point
ip address 10.20.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 200
log-adjacency-changes
redistribute connected subnets
network 10.17.17.17 0.0.0.0 area 200
network 10.11.0.0 0.0.255.255 area 200
network 10.12.0.0 0.0.255.255 area 200
network 10.13.0.0 0.0.255.255 area 200
network 10.18.0.0 0.0.255.255 area 200
network 10.19.0.0 0.0.255.255 area 200
network 10.20.0.0 0.0.255.255 area 200
```

PE2 (72K-36-6) Configuration

```
ip cef
!
ip vrf customersite
rd 200:1
route-target export 200:1
route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
ip address 10.18.18.18 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Ethernet3/0
ip vrf forwarding customersite
ip address 10.29.0.2 255.255.0.0
```

```
no ip directed-broadcast
!
interface Ethernet3/1
ip vrf forwarding customersite
ip address 10.30.0.2 255.255.0.0
no ip directed-broadcast
!
interface ATM5/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
ip address 10.11.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.2 point-to-point
ip address 10.12.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.3 point-to-point
ip address 10.13.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 200
log-adjacency-changes
redistribute connected subnets
passive-interface Ethernet3/0
passive-interface Ethernet3/1
network 10.18.18.18 0.0.0.0 area 200
network 10.11.0.0 0.0.255.255 area 200
network 10.12.0.0 0.0.255.255 area 200
network 10.13.0.0 0.0.255.255 area 200
!
router bgp 200
no bgp default ipv4-unicast
bgp log-neighbor-changes
timers bgp 10 30
neighbor 10.16.16.16 remote-as 200
neighbor 10.16.16.16 update-source Loopback0
!
address-family ipv4
neighbor 10.16.16.16 activate
neighbor 10.16.16.16 send-community extended
no synchronization
exit-address-family
!
address-family vpnv4
```

```
neighbor 10.16.16.16 activate
neighbor 10.16.16.16 send-community extended
exit-address-family
!
address-family ipv4 vrf customersite
neighbor 10.29.0.1 remote-as 300
neighbor 10.29.0.1 activate
neighbor 10.29.0.1 as-override
neighbor 10.29.0.1 advertisement-interval 5
neighbor 10.30.0.1 remote-as 300
neighbor 10.30.0.1 activate
neighbor 10.30.0.1 as-override
neighbor 10.30.0.1 advertisement-interval 5
no auto-summary
no synchronization
exit-address-family
```

CE2 (36K-38-4) Configuration

```
ip cef
!
interface Loopback0
ip address 10.21.21.21 255.255.255.255
!
interface Ethernet1/3
ip address 10.29.0.1 255.255.0.0
!
interface Ethernet5/0
ip address 10.14.0.1 255.255.0.0
!
router ospf 300
log-adjacency-changes
redistribute connected subnets
redistribute bgp 300 subnets
passive-interface Ethernet1/3
network 10.21.21.21 0.0.0.0 area 300
network 10.14.0.0 0.0.255.255 area 300
!
router bgp 300
no synchronization
timers bgp 10 30
redistribute connected
redistribute ospf 300 match internal external 1 external 2
neighbor 10.29.0.2 remote-as 200
neighbor 10.29.0.2 advertisement-interval 5
no auto-summary
```

CE3 (36K-38-5) Configuration

```
ip cef
!
interface Loopback0
ip address 10.20.20.20 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0/2
ip address 10.30.0.1 255.255.0.0
no ip directed-broadcast
!
interface Ethernet0/3
ip address 10.14.0.2 255.255.0.0
no ip directed-broadcast
```

```
!
router ospf 300
log-adjacency-changes
redistribute connected subnets
redistribute bgp 300 subnets
passive-interface Ethernet0/2
network 10.20.20.20 0.0.0.0 area 300
network 10.14.0.0 0.0.255.255 area 300
!
router bgp 300
no synchronization
bgp log-neighbor-changes
timers bgp 10 30
redistribute connected
redistribute ospf 300 match internal external 1 external 2
neighbor 10.30.0.2 remote-as 200
neighbor 10.30.0.2 advertisement-interval 5
no auto-summary
```

Additional References for MPLS VPN Carrier Supporting Carrier Using LDP and an IGP

Related Documents

Related Topic	Document Title
MPLS	MPLS Product Literature

RFCs

RFC	Title
RFC 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN CSC with LDP and IGP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 153: Feature Information for MPLS VPN CSC with LDP and IGP

Feature Name	Releases	Feature Configuration Information
MPLS VPN Carrier Supporting Carrier	12.0(14)ST 12.0(16)ST 12.2(8)T 12.0(21)ST 12.0(22)S 12.0(23)S Cisco IOS XE Release 2.2	This feature enables you to set up and create an MPLS VPN CSC network that uses LDP to transport MPLS labels and an IGP to transport routes. In 12.0(14)ST, this feature was introduced. In 12.0(16)ST, this feature was integrated. In 12.2(8)T, this feature was integrated. In 12.0(21)ST, this feature was integrated. In 12.0(22)S, this feature was integrated. In 12.0(23)S, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.

Glossary

ASBR -- Autonomous System Boundary router. A router that connects one autonomous system to another.

autonomous system --A collection of networks under a common administration sharing a common routing strategy.

BGP --Border Gateway Protocol. An interdomain routing protocol that exchanges network reachability information with other BGP systems (which may be within the same autonomous system or between multiple autonomous systems).

CE router--customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers do not recognize associated MPLS VPNs.

CSC --Carrier Supporting Carrier. A hierarchical VPN model that allows small service providers, or customer carriers, to interconnect their IP or MPLS networks over an MPLS backbone. This eliminates the need for customer carriers to build and maintain their own MPLS backbone.

eBGP --external Border Gateway Protocol. A BGP between routers located within different autonomous systems. When two routers, located in different autonomous systems, are more than one hop away from one another, the eBGP session between the two routers is considered a multihop BGP.

edge router--A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

iBGP --internal Border Gateway Protocol. A BGP between routers within the same autonomous system.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within a single autonomous system. Examples of common Internet IGP protocols include IGRP, OSPF, IS-IS, and RIP.

IP --Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

LFIB --Label Forwarding Information Base. Data structure used in MPLS to hold information about incoming and outgoing labels and associated Forwarding Equivalence Class (FEC) packets.

MP-BGP --Multiprotocol BGP.

MPLS --Multiprotocol Label Switching. The name of the IETF working group responsible for label switching, and the name of the label switching approach it has standardized.

NLRI --Network Layer Reachability Information. The BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and extended community values.

NSF --Nonstop forwarding enables routers to continuously forward IP packets following a Route Processor takeover or switchover to another Route Processor. NSF maintains and updates Layer 3 routing and forwarding information in the backup Route Processor to ensure that IP packets and routing protocol information are forwarded continuously during the switchover and route convergence process.

PE router--provider edge router. A router that is part of a service provider's network. It is connected to a customer edge (CE) router. All MPLS VPN processing occurs in the PE router.

QoS --quality of service. Measure of performance for a transmission system that indicates its transmission quality and service availability.

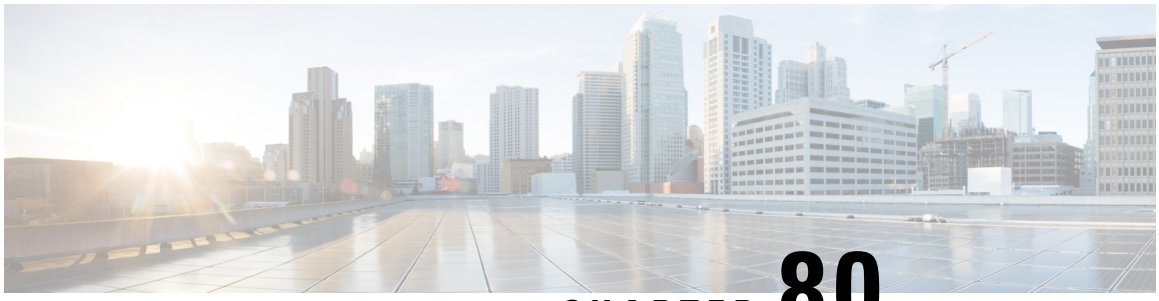
RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is imported.

SLA --Service Level Agreement given to VPN subscribers.

VPN --Virtual Private Network. A secure MPLS-based network that shares resources on one or more physical networks (typically implemented by one or more service providers). A VPN contains geographically dispersed sites that can communicate securely over a shared backbone network.

VRF --VPN routing and forwarding instance. Routing information that defines a VPN site that is attached to a PE router. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.



CHAPTER 80

MPLS VPN Carrier Supporting Carrier with BGP

Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. This module explains how to configure an MPLS VPN CSC network that uses Border Gateway Protocol (BGP) to distribute routes and MPLS labels.

Prerequisites for MPLS VPN CSC with BGP

- You should be able to configure MPLS VPNs with end-to-end (CE-to-CE router) pings working. To accomplish this, you need to know how to configure Interior Gateway Protocols (IGPs), MPLS Label Distribution Protocol (LDP), and Multiprotocol Border Gateway Protocol (MP-BGP).
- Make sure that the CSC-PE routers and the CSC-CE routers run images that support BGP label distribution. Otherwise, you cannot run external BGP (EBGP) between them. Ensure that connectivity between the customer carrier and the backbone carrier. EBGP-based label distribution is configured on these links to enable MPLS between the customer and backbone carriers.

Restrictions for MPLS VPN CSC with BGP

On a provider edge (PE) router, you can configure an interface for either BGP with labels or LDP. You cannot enable both types of label distribution on the same interface. If you switch from one protocol to the other, then you must disable the existing protocol on all interfaces before enabling the other protocol.

This feature does not support the following:

- EBGP multihop between CSC-PE and CSC-CE routers
- EIBGP multipath load sharing

The physical interfaces that connect the BGP speakers must support Cisco Express Forwarding or distributed Cisco Express Forwarding and MPLS.

Information About MPLS VPN CSC with BGP

MPLS VPN CSC Introduction

Carrier supporting carrier is where one service provider allows another service provider to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

A backbone carrier offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services. The customer carrier can be either:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

Benefits of Implementing MPLS VPN CSC

The MPLS VPN CSC network provides the following benefits to service providers who are backbone carriers and to customer carriers.

Benefits to the Backbone Carrier

- The backbone carrier can accommodate many customer carriers and give them access to its backbone. The backbone carrier does not need to create and maintain separate backbones for its customer carriers. Using one backbone network to support multiple customer carriers simplifies the backbone carrier's VPN operations. The backbone carrier uses a consistent method for managing and maintaining the backbone network. This is also cheaper and more efficient than maintaining separate backbones.
- The MPLS VPN carrier supporting carrier feature is scalable. Carrier supporting carrier can change the VPN to meet changing bandwidth and connectivity needs. The feature can accommodate unplanned growth and changes. The carrier supporting carrier feature enables tens of thousands of VPNs to be set up over the same network, and it allows a service provider to offer both VPN and Internet services.
- The MPLS VPN carrier supporting carrier feature is a flexible solution. The backbone carrier can accommodate many types of customer carriers. The backbone carrier can accept customer carriers who are ISPs or VPN service providers or both. The backbone carrier can accommodate customer carriers that require security and various bandwidths.

Benefits to the Customer Carriers

- The MPLS VPN carrier supporting carrier feature removes from the customer carrier the burden of configuring, operating, and maintaining its own backbone. The customer carrier uses the backbone network of a backbone carrier, but the backbone carrier is responsible for network maintenance and operation.

- Customer carriers who use the VPN services provided by the backbone carrier receive the same level of security that Frame Relay or ATM-based VPNs provide. Customer carriers can also use IPSec in their VPNs for a higher level of security; it is completely transparent to the backbone carrier.
- Customer carriers can use any link layer technology (SONET, DSL, Frame Relay, and so on) to connect the CE routers to the PE routers and the PE routers to the P routers. The MPLS VPN carrier supporting carrier feature is link layer independent. The CE routers and PE routers use IP to communicate, and the backbone carrier uses MPLS.
- The customer carrier can use any addressing scheme and still be supported by a backbone carrier. The customer address space and routing information are independent of the address space and routing information of other customer carriers or the backbone provider.

Benefits of Implementing MPLS VPN CSC with BGP

You can configure your CSC network to enable BGP to transport routes and MPLS labels between the backbone carrier PE routers and the customer carrier CE routers using multiple paths. The benefits of using BGP to distribute IPv4 routes and MPLS label routes are:

- BGP takes the place of an IGP and LDP in a VPN forwarding/routing instance (VRF) table. You can use BGP to distribute routes and MPLS labels. Using a single protocol instead of two simplifies the configuration and troubleshooting.
- BGP is the preferred routing protocol for connecting two ISPs, mainly because of its routing policies and ability to scale. ISPs commonly use BGP between two providers. This feature enables those ISPs to use BGP.

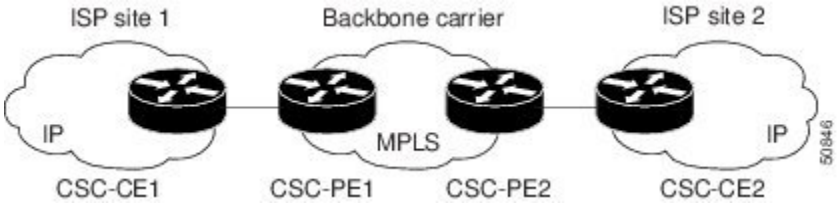
Configuration Options for MPLS VPN CSC with BGP

The following sections explain how the backbone and customer carriers distribute IPv4 routes and MPLS labels. The backbone carrier offers BGP and MPLS VPN services. The customer carrier can be either of the following:

Customer Carrier Is an ISP with an IP Core

The figure below shows a network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a point of presence (POP). The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS. The ISP sites use IP.

Figure 134: Network Where the Customer Carrier Is an ISP



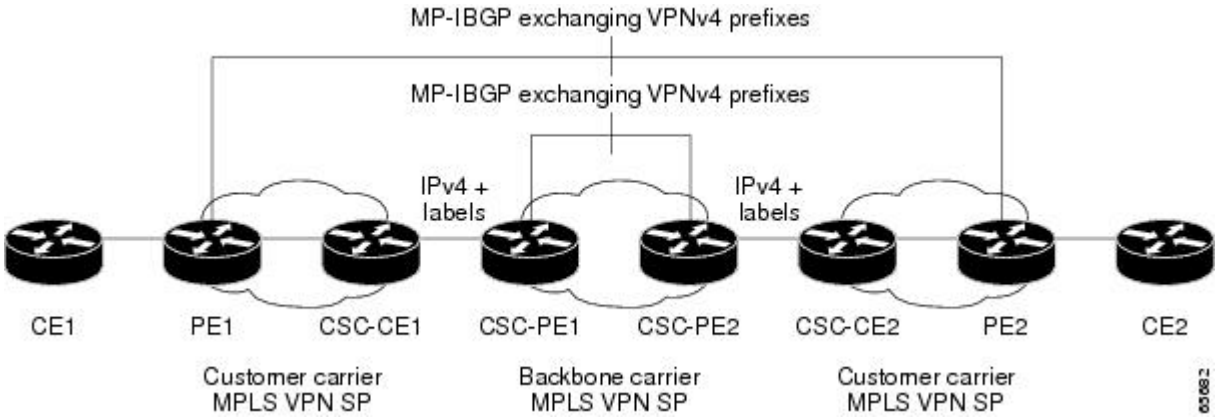
The links between the CE and PE routers use EBGP to distribute IPv4 routes and MPLS labels. Between the links, the PE routers use multiprotocol IBGP to distribute VPNv4 routes.

Note If a router other than a Cisco router is used as a CSC-PE or CSC-CE, that router must support IPv4 BGP label distribution (RFC 3107). Otherwise, you cannot run EBGP with labels between the routers.

Customer Carrier Is an MPLS Service Provider With or Without VPN Services

The figure below shows a network configuration where the backbone carrier and the customer carrier are BGP/MPLS VPN service providers. This is known as hierarchical VPNs. The customer carrier has two sites. Both the backbone carrier and the customer carrier use MPLS in their networks.

Figure 135: Network Where the Customer Carrier Is an MPLS VPN Service Provider



In this configuration, the customer carrier can configure its network in one of the following ways:

- The customer carrier can run IGP and LDP in its core network. In this case, the CSC-CE1 router in the customer carrier redistributes the EBGP routes it learns from the CSC-PE1 router of the backbone carrier to IGP.
- The CSC-CE1 router of the customer carrier system can run an IPv4 and labels IBGP session with the PE1 router.

How to Configure MPLS VPN CSC with BGP

Identifying the Carrier Supporting Carrier Topology

Before you configure the MPLS VPN CSC with BGP, you need to identify both the backbone and customer carrier topology.

For hierarchical VPNs, the customer carrier of the MPLS VPN network provides MPLS VPN services to its own customers. In this instance, you need to identify the type of customer carrier as well as the topology of the customer carriers. Hierarchical VPNs require extra configuration steps, which are noted in the configuration sections.

Note You can connect multiple CSC-CE routers to the same PE, or you can connect a single CSC-CE router to CSC-PEs using more than one interface to provide redundancy and multiple path support in CSC topology.

Perform this task to identify the carrier supporting carrier topology.

SUMMARY STEPS

1. Identify the type of customer carrier, ISP or MPLS VPN service provider.
2. (For hierarchical VPNs only) Identify the CE routers.
3. (For hierarchical VPNs only) Identify the customer carrier core router configuration.
4. Identify the customer carrier edge (CSC-CE) routers.
5. Identify the backbone carrier router configuration.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the type of customer carrier, ISP or MPLS VPN service provider.	Sets up requirements for configuration of carrier supporting carrier network. • For an ISP, customer site configuration is not required. • For an MPLS VPN service provider, the customer site needs to be configured, as well as any task or step designated "for hierarchical VPNs only."
Step 2	(For hierarchical VPNs only) Identify the CE routers.	Sets up requirements for configuration of CE to PE connections.
Step 3	(For hierarchical VPNs only) Identify the customer carrier core router configuration.	Sets up requirements for connection configuration between core (P) routers and between P routers and edge routers (PE and CSC-CE routers).
Step 4	Identify the customer carrier edge (CSC-CE) routers.	Sets up requirements for configuration of CSC-CE to CSC-PE connections.
Step 5	Identify the backbone carrier router configuration.	Sets up requirements for connection configuration between CSC core routers and between CSC core routers and edge routers (CSC-CE and CSC-PE routers).

What to Do Next

Set up your carrier supporting carrier networks.

Configuring the Backbone Carrier Core

Configuring the backbone carrier core requires setting up connectivity and routing functions for the CSC core and the CSC-PE routers.

Configuring and verifying the CSC core (backbone carrier) involves the following tasks:

Prerequisites

Before you configure a backbone carrier core, configure the following on the CSC core routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on.
- Label Distribution Protocol (LDP). For information, see How to Configure MPLS LDP.

Verifying IP Connectivity and LDP Configuration in the CSC Core

Perform this task to verify IP connectivity and LDP configuration in the CSC core.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
5. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**]
6. **show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**]
7. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
8. **show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] | **all**]
9. **show ip route**
10. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name*	*system-address*} **Example:** `Router# ping ip 10.1.0.0`	(Optional) Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. • Use the **ping ip** command to verify the connectivity from one CSC core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.2.0.0`	(Optional) Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to verify the path that a packet goes through before reaching the final destination. The **trace** command can help isolate a trouble spot if two routers cannot communicate.

	Command or Action	Purpose
Step 4	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table`	(Optional) Displays the contents of the MPLS label forwarding information base (LFIB). • Use the **show mpls forwarding-table** command to verify that MPLS packets are being forwarded.
Step 5	**show mpls ldp discovery** [**vrf** *vrf-name* | **all**] **Example:** `Router# show mpls ldp discovery`	(Optional) Displays the status of the LDP discovery process. • Use the **show mpls ldp discovery** command to verify that LDP is operational in the CSC core.
Step 6	**show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**] **Example:** `Router# show mpls ldp neighbor`	(Optional) Displays the status of LDP sessions. • Use the **show mpls ldp neighbor** command to verify LDP configuration in the CSC core.
Step 7	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef`	(Optional) Displays entries in the forwarding information base (FIB). • Use the **show ip cef** command to check the forwarding table (prefixes, next hops, and interfaces).
Step 8	**show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] | **all**] **Example:** `Router# show mpls interfaces`	(Optional) Displays information about one or more or all interfaces that are configured for label switching. • Use the **show mpls interfaces** command to verify that the interfaces are configured to use LDP.
Step 9	**show ip route** **Example:** `Router# show ip route`	(Optional) Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, interface, and so forth.
Step 10	**disable** **Example:** `Router# disable`	(Optional) Returns to privileged EXEC mode.

Troubleshooting Tips

You can use the **ping** and **trace** commands to verify complete MPLS connectivity in the core. You also get useful troubleshooting information from the additional **show** commands.

Configuring VRFs for CSC-PE Routers

Perform this task to configure VPN forwarding/routing instances (VRFs) for the backbone carrier edge (CSC-PE) routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **exit**
8. **interface** *type number*
9. **ip vrf forwarding** *vrf-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	Example:
	`Router(config-vrf)# rd 100:1`
		Creates routing and forwarding tables.
		• The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		  • 16-bit AS number: your 32-bit number, for example, 101:3
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community*	Creates a route-target extended community for a VRF.

	Command or Action	Purpose
	Example: `Router(config-vrf)# route-target import 100:1`	• The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	(Optional) Exits to global configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	Specifies the interface to configure. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip vrf detail** command and make sure the MPLS VPN is up and associated with the right interfaces.

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier

Perform this task to configure Multiprotocol BGP (MP-BGP) connectivity in the backbone carrier.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type*
7. **address-family vpnv4** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
9. **neighbor** {*ip-address* | *peer-group-name*} **activate**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	(Optional) Disables the IPv4 unicast address family on all neighbors. • Use the **no bgp default-unicast** command if you are using this neighbor for MPLS routes only.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:**	Adds an entry to the BGP or multiprotocol BGP neighbor table.

	Command or Action	Purpose
	`Router(config-router)# neighbor 10.5.5.5 remote-as 100`	• The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type* **Example:** `Router(config-router)# neighbor 10.2.0.0 update-source loopback0`	Allows BGP sessions to use a specific operational interface for TCP connections. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *interface-type* argument specifies the interface to be used as the source.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.4.0.0 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 10	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp x.x.x.x events** command, where *x.x.x.x* is the IP address of the neighbor.

Configuring the CSC-PE and CSC-CE Routers

Perform the following tasks to configure and verify links between a CSC-PE router and the carrier CSC-CE router for an MPLS VPN CSC network that uses BGP to distribute routes and MPLS labels.

The figure below shows the configuration for the peering with directly connected interfaces between CSC-PE and CSC-CE routers. This configuration is used as the example in the tasks that follow.

Figure 136: Configuration for Peering with Directly Connected Interfaces Between CSC-PE and CSC-CE Routers

Configuring CSC-PE Routers

Perform this task to configure the CSC-PE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** *ip-address* **as-override**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** *ip-address* **as-override** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 as-override`	Configures a PE router to override the autonomous system number (ASN) of a site with the ASN of a provider. • The *ip-address* argument specifies the IP address of the router that is to be overridden with the ASN provided.

	Command or Action	Purpose
Step 8	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. Make sure you see the following line in the command output under Neighbor capabilities:

```
IPv4 MPLS Label capability:advertised and received
```

Configuring CSC-CE Routers

Perform this task to configure the CSC-CE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **redistribute** *protocol*
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.

Step 5	**redistribute** *protocol*
	Example:
	`Router(config-router-af)# redistribute static`
		Redistributes routes from one routing domain into another routing domain.
		• The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **egp**, **igrp**, **isis**, **ospf**, **mobile**, **static** [**ip**], **connected**, and **rip**.
		  • The **static** [**ip**] keyword redistributes IP static routes. The optional **ip** keyword is used when you redistribute static routes into IS-IS.
		  • The **connected** keyword refers to routes which are established automatically when IP is enabled on an interface. For routing protocols such as OSPF and IS-IS, these routes are redistributed as external to the autonomous system.

Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.5.0.2 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor.

	Command or Action	Purpose
		• The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.3.0.2 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from the address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Verifying Labels in the CSC-PE Routers

Perform this task to verify the labels in the CSC-PE routers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
3. **show mpls interfaces** [**all**]
4. **show ip route vrf** *vrf-name* [*prefix*]
5. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
6. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
7. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
8. **traceroute vrf** [*vrf-name*] *ip-address*
9. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all summary`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4 all summary** command to check that the BGP session is up and running between the CSC-PE routers and the CSC-CE routers. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.
Step 3	**show mpls interfaces** [**all**] **Example:** `Router# show mpls interfaces all`	(Optional) Displays information about one or more interfaces that have been configured for label switching. • Use the **show mpls interfaces all** command to check that MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. Check that LDP is turned off on the VRF because EBGP distributes the labels.
Step 4	**show ip route vrf** *vrf-name* [*prefix*] **Example:** `Router# show ip route vrf vpn1 10.5.5.5`	(Optional) Displays the IP routing table associated with a VRF. • Use the **show ip route vrf** command to check that the prefixes for the PE routers are in the routing table of the CSC-PE routers. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.
Step 5	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 vrf vpn1 labels`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4 vrf** *vrf-name* **labels** command to check that the prefixes for the customer carrier MPLS service provider networks are in the BGP table and have the appropriate labels. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.

	Command or Action	Purpose
Step 6	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef vrf vpn1 10.1.0.0 detail`	(Optional) Displays entries in the forwarding information base (FIB) or displays a summary of the FIB. • Use the **show ip cef vrf** and the **show ip cef vrf detail** commands to check that the prefixes of the PE routers are in the CEF table.
Step 7	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn1 10.1.0.0 detail`	(Optional) Displays the contents of the MPLS lable forwarding information base (LFIB). • Use the **show mpls forwarding-table** command with the **vrf** keyword and both the **vrf** and **detail** keywords to check that the prefixes for the PE routers in the local customer MPLS VPN service provider are in the LFIB. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 8	**traceroute vrf** [*vrf-name*] *ip-address* **Example:** `Router# traceroute vrf vpn2 10.2.0.0`	Shows the routes that packets follow traveling through a network to their destination. • Use the **traceroute vrf** command to check the data path and transport labels from a PE to a destination CE router. **Note** This command works with MPLS-aware traceroute only if the backbone routers are configured to propagate and generate IP Time to Live (TTL) information. For more information, see the documentation on the **mpls ip propagate-ttl** command. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 9	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying Labels in the CSC-CE Routers

Perform this task to verify the labels in the CSC-CE routers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp summary**
3. **show ip route** [*address*]
4. **show mpls ldp bindings** [*network* {*mask* | *length*}]
5. **show ip cef** [*network* [*mask*]] [**longer-prefixes**] [**detail**]
6. **show mpls forwarding table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
7. **show ip bgp labels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp summary** **Example:** `Router# show ip bgp summary`	(Optional) Displays the status of all BGP connections. • Use the **show ip bgp summary** command to check that the BGP session is up and running on the CSC-CE routers.
Step 3	**show ip route** [*address*] **Example:** `Router# show ip route 10.1.0.0`	(Optional) Displays IP routing table entries. • Use the **show ip route** to check that the loopback address of the local and remote PE routers are in the routing table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 4	**show mpls ldp bindings** [*network* {*mask*	*length*}] **Example:** `Router# show mpls ldp bindings 10.2.0.0 255.255.255.255`	(Optional) Displays the contents of the label information base (LIB). • Use the **show mpls ldp bindings** command to check that the prefix of the local PE router is in the MPLS LDP bindings.
Step 5	**show ip cef** [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef 10.5.0.0 detail`	(Optional) Displays entries in the forwarding information base (FIB) or a summary of the FIB. • Use the **show ip cef** and the **show ip cef detail** commands to check that the prefixes of the local and remote PE routers are in the Cisco Express Forwarding table.

	Command or Action	Purpose
		Note If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 6	**show mpls forwarding table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table 10.2.0.0 detail`	(Optional) Displays the contents of the MPLS LFIB. • Use the **show mpls forwarding-table** and **show mpls forwarding-table detail** commands to check that the prefixes of the local and remote PE routers are in the MPLS forwarding table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.
Step 7	**show ip bgp labels** **Example:** `Router# show ip bgp labels`	(Optional) Displays information about MPLS labels from the EBGP route table. • Use the **show ip bgp labels** command to check that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks.

Configuring the Customer Carrier Network

Perform the following tasks to configure and verify the customer carrier network. This requires setting up connectivity and routing functions for the customer carrier core (P) routers and the customer carrier edge (PE) routers.

Prerequisites

Before you configure an MPLS VPN CSC network that uses BGP to distribute routes and MPLS labels, you must configure the following on your customer carrier routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on. For information, see Configuring a Basic BGP Network, Configuring OSPF, Configuring a Basic IS-IS Network, and Configuring EIGRP.
- MPLS VPN functionality on the PE routers (for hierarchical VPNs only).
- Label Distribution Protocol (LDP) on P and PE routers (for hierarchical VPNs only). For information, see How to Configure MPLS LDP.

Note You must configure the items in the preceding list before performing the tasks in this section.

Verifying IP Connectivity in the Customer Carrier

Perform this task to verify IP connectivity in the customer carrier.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route**
5. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name* | *system-address*} **Example:** `Router# ping ip 10.2.0.0`	Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. • Use the **ping** command to verify the connectivity from one customer carrier core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.1.0.0`	Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to verify the path that a packet goes through before reaching the final destination. The **trace** command can help isolate a trouble spot if two routers cannot communicate.
Step 4	**show ip route** **Example:** `Router# show ip route`	Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, interface, and so forth.
Step 5	**disable** **Example:** `Router# disable`	Returns to user mode.

Configuring a Customer Carrier Core Router as a Route Reflector

Perform this task to configure a customer carrier core (P) router as a route reflector of multiprotocol BGP prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** *ip-address* **route-reflector-client**
8. **exit-address-family**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and labels the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.1.1.1 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.

	Command or Action	Purpose
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.1.1.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor 10.1.1.1 route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client. • The *ip-address* argument specifies the IP address of the BGP neighbor being identified as a client.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. For neighbors to exchange other address prefix types, such as multicast and VPNv4, you must also activate neighbors using the **neighbor activate** command in address family configuration mode, as shown.

Route reflectors and clients (neighbors or internal BGP peer groups) that are defined in router configuration mode using the **neighbor route-reflector-client** command reflect unicast address prefixes to and from those clients by default. To cause them to reflect prefixes for other address families, such as multicast, define the reflectors and clients in address family configuration mode, using the **neighbor route-reflector-client** command, as shown.

Configuring the Customer Site for Hierarchical VPNs

Note This section applies only to customer carrier networks that use BGP to distribute routes and MPLS labels.

Perform the following tasks to configure and verify the customer site for hierarchical VPNs:

Note This section applies to hierarchical VPNs only.

Defining VPNs on PE Routers for Hierarchical VPNs

Perform this task to define VPNs on PE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **ip vrf forwarding** *vrf-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn2`	Creates a VRF routing table and a Cisco Express Forwarding table and enters VRF configuration mode. • The *vrf-name* argument is a name you assign to a VRF.
Step 4	**rd** *route-distinguisher* **Example:**	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.

	Command or Action	Purpose
	`Router(config-vrf)# rd 200:1`	
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 200:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map map23`	Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-vrf)# ip vrf forwarding vpn2`	Associates a VPN VRF instance with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.

Configuring BGP Routing Sessions on the PE Routers for Hierarchical VPNs

Perform this task to configure BGP routing sessions on the PE routers for PE-to-CE router communication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures the router to run a BGP process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 multicast`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.5.5.5 remote-as 300`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:**	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor 10.1.0.0 activate`	• The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Verifying Labels in Each PE Router for Hierarchical VPNs

Perform this task to verify labels in each PE router for hierarchical VPNs.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show mpls forwarding-table** [**vrf** *vrf-name*] [*prefix*] [**detail**]
4. **show ip cef** [*network* [*mask* [**longer-prefix**]]] [**detail**]
5. **show ip cef vrf** *vrf-name* [*ip-prefix*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route vrf** *vrf-name* [*prefix*] **Example:** `Router# show ip route vrf vpn2 10.5.5.5`	(Optional) Displays the IP routing table associated with a VRF. • Use the **show ip route vrf** command to check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.
Step 3	**show mpls forwarding-table** [**vrf** *vrf-name*] [*prefix*] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn2 10.1.0.0`	(Optional) Displays the contents of the LFIB. • Use the **show mpls forwarding-table** command to check that the prefixes for the local and remote CE routers are in the MPLS forwarding table, and that the specified prefix is untagged.
Step 4	**show ip cef** [*network* [*mask* [**longer-prefix**]]] [**detail**] **Example:**	(Optional) Displays specific entries in the FIB based on IP address information.

	Command or Action	Purpose
	`Router# show ip cef 10.2.0.0`	• Use the **show ip cef** command to check that the prefixes of the local and remote PE routers are in the Cisco Express Forwarding table.
Step 5	**show ip cef vrf** *vrf-name* [*ip-prefix*] **Example:** `Router# show ip cef vrf vpn2 10.3.0.0`	(Optional) Displays the Cisco Express Forwarding table associated with a VRF. • Use the **show ip cef vrf** command to check that the prefix of the remote CE router is in the Cisco Express Forwarding table.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits to user EXEC mode.

Configuring CE Routers for Hierarchical VPNs

Perform this task to configure CE routers for hierarchical VPNs. This configuration is the same as that for an MPLS VPN that is not in a hierarchical topology.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface** *type number*
5. **ip addres** *ip-address mask* [**secondary**]
6. **exit**
7. **router bgp** *as-number*
8. **redistribute** *protocol*
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip cef** [**distributed**] **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding on the route processor card. • The **distributed** keyword enables distributed Cisco Express Forwarding operation. Cisco Express Forwarding information is distributed to the line cards. Line cards perform express forwarding. **Note** For the Cisco ASR 1000 Series Aggregation Services Router, the **distributed** keyword is required.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 0`	Configures an interface type and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • A loopback interface indicates a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. • The *number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.
Step 5	**ip addres** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.8.0.0 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private

	Command or Action	Purpose
		autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 8	**redistribute** *protocol* **Example:** `Router(config-router)# redistribute connected`	Redistributes routes from one routing domain into another routing domain. • The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **mobile**, **ospf**, **static** [**ip**], or **rip**. The **connected** keyword refers to routes that are established automatically when IP is enabled on an interface. For routing protocols such as Open Shortest Path First (OSPF) and IS-IS, these routes are redistributed as external to the autonomous system.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.8.0.0 remote-as 100`	Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Verifying IP Connectivity in the Customer Site

Perform this task to verify IP connectivity in the customer site.

SUMMARY STEPS

1. **enable**
2. **show ip route** [*ip-address* [*mask*]] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**
3. **ping** [*protocol*] {*host-name* | *system-address*}
4. **trace** [*protocol*] [*destination*]
5. **disable**

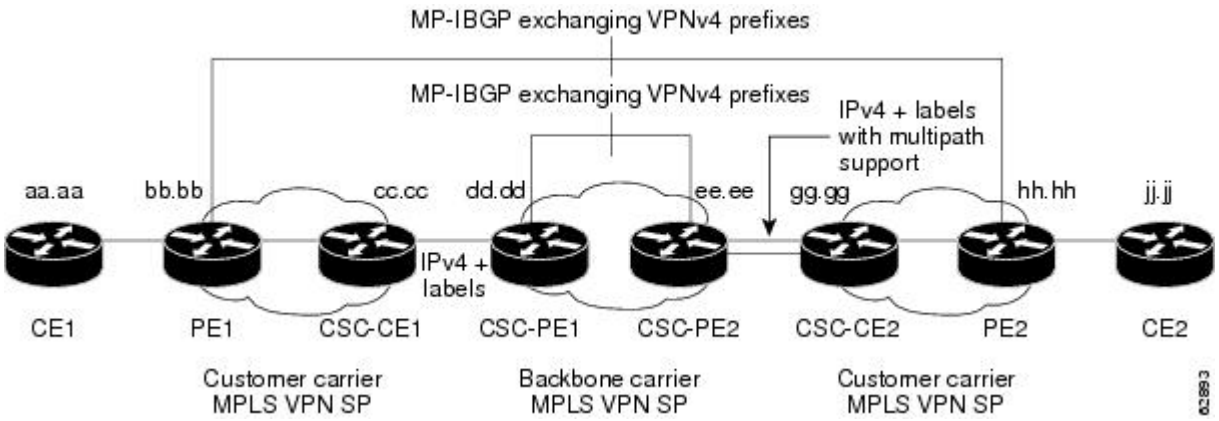
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route** [*ip-address* [*mask*]] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download** **Example:** `Router# show ip route 10.5.5.5`	(Optional) Displays the current state of the routing table. • Use the **show ip route ip-address** command to check that the loopback addresses of the remote CE routers learned through the PE router are in the routing table of the local CE routers.
Step 3	**ping** [*protocol*] {*host-name* | *system-address*} **Example:** `Router# ping 10.5.5.5`	Diagnoses basic network connectivity on Apollo, AppleTalk, Connectionless Network Service (CLNS), DECnet, IP, Novell IPX, VINES, or XNS networks. • Use the **ping** command to check connectivity between customer site routers.
Step 4	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.5.5.5`	Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to follow the path of the packets in the customer site. • To use nondefault parameters and invoke an extended **trace** test, enter the **trace** command without a destination argument. You will be stepped through a dialog to select the desired parameters.
Step 5	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Configuration Examples for MPLS VPN CSC with BGP

The figure below shows a sample CSC topology for exchanging IPv4 routes and MPLS labels. Use this figure as a reference for configuring and verifying carrier supporting carrier routers to exchange IPv4 routes and MPLS labels.

Figure 137: Sample CSC Topology for Exchanging IPv4 Routes and MPLS Labels



The table below describes the sample configuration shown in the figure above.

Table 154: Description of Sample Configuration Shown in figure 1

Routers	Description
CE1 and CE2	Belong to an end customer. CE1 and CE2 routers exchange routes learned from PE routers. The end customer is purchasing VPN services from a customer carrier.
PE1 and PE2	Part of a customer carrier network that is configured to provide MPLS VPN services. PE1 and PE2 are peering with a VPNv4 IBGP session to form an MPLS VPN network.
CSC-CE1 and CSC-CE2	Part of a customer carrier network. CSC-CE1 and CSC-CE2 routers exchange IPv4 BGP updates with MPLS labels and redistribute PE loopback addressees to and from the IGP (OSPF in this example). The customer carrier is purchasing carrier supporting carrier VPN services from a backbone carrier.
CSC-PE1 and CSC-PE2	Part of the backbone carrier's network configured to provide carrier supporting carrier VPN services. CSC-PE1 and CSC-PE2 are peering with a VPNv4 IP BGP session to form the MPLS VPN network. In the VRF, CSC-PE1 and CSC-PE2 are peering with the CSC-CE routers, which are configured for carrying MPLS labels with the routes, with an IPv4 EBGP session.

Configuring the Backbone Carrier Core Examples

Configuration and verification examples for the backbone carrier core included in this section are as follows:

Verifying IP Connectivity and LDP Configuration in the CSC Core Example

Check that CSC-PE2 is reachable from CSC-PE1 by entering the following command on CSC-CE1:

```
Router# ping 10.5.5.5
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Verify the path from CSC-PE1 to CSC-PE2 by entering the following command on CSC-CE1:

```
Router# trace 10.5.5.5
Type escape sequence to abort.
Tracing the route to 10.5.5.5
  1 10.5.5.5 0 msec 0 msec *
```

Check that CSC-PE router prefixes are in the MPLS forwarding table:

```
Router# show mpls forwarding-table
Local    Outgoing     Prefix or         Bytes tag    Outgoing      Next Hop
tag      tag or VC    Tunnel Id         switched     interface
16       2/nn         dd.dd.dd.dd/32      0          AT2/1/0.1     point2point
17       16           bb.bb.bb.bb/32[V]   30204      Et1/0         pp.0.0.1
21       Pop tag      cc.cc.cc.cc/32[V]   0          Et1/0         pp.0.0.1
22       Pop tag      nn.0.0.0/8[V]       570        Et1/0         pp.0.0.1
23       Aggregate    pp.0.0.0/8[V]       0
2        2/nn         gg.gg.gg.gg/32[V]   0          AT3/0.1       point2point
8        2/nn         hh.hh.hh.hh/32[V]   15452      AT3/0.1       point2point
29       2/nn         qq.0.0.0/8[V]       0          AT3/0.1       point2point
30       2/nn         ss.0.0.0/8[V]       0          AT3/0.1       point2point
```

Check the status of LDP discovery processes in the core:

```
Router# show mpls ldp discovery
Local LDP Identifier:
  ee.ee.ee.ee:0
  Discovery Sources:
  Interfaces:
    ATM2/1/0.1 (ldp): xmit/recv
      TDP Id: dd.dd.dd.dd:1
```

Check the status of LDP sessions in the core:

```
Router# show mpls ldp neighbor
Peer LDP Ident: dd.dd.dd.dd:1; Local LDP Ident ee.ee.ee.ee:1
       TCP connection: dd.dd.dd.dd.646 - ee.ee.ee.ee.11007
       State: Oper; Msgs sent/rcvd: 20/21; Downstream on demand
       Up time: 00:14:56
       LDP discovery sources:
         ATM2/1/0.1, Src IP addr: dd.dd.dd.dd
```

Check the forwarding table (prefixes, next-hops, and interfaces):

```
Router# show ip cef
Prefix              Next Hop          Interface
0.0.0.0/0           drop              Null0 (default route handler entry)
0.0.0.0/32          receive
dd.dd.dd.dd/32      dd.dd.dd.dd       ATM2/1/0.1
ee.ee.ee.ee/32      receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32 receive
```

Note Also see the Verifying Labels in the CSC-CE Routers Examples, on page 1673.

Verify that interfaces are configured to use LDP:

```
Router# show mpls interfaces
Interface              IP            Tunnel   Operational
Ethernet0/1            Yes (ldp)     No       Yes
```

Display the entire routing table, including host IP address, next hop, interface, and so forth:

```
Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
Gateway of last resort is not set
     dd.0.0.0/32 is subnetted, 1 subnets
O       dd.dd.dd.dd [110/7] via dd.dd.dd.dd, 00:16:42, ATM2/1/0.1
     ee.0.0.0/32 is subnetted, 1 subnets
C       ee.ee.ee.ee is directly connected, Loopback0
```

Configuring VRFs for CSC-PE Routers Example

The following example shows how to configure a VPN routing and forwarding (VRF) instance for a CSC-PE router:

```
ip cef distributed
ip vrf vpn1
rd 100:1
route target both 100:1
!
```

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier Example

The following example shows how to configure Multiprotocol BGP (MP-BGP) for VPN connectivity in the backbone carrier:

```
ip cef distributed
ip vrf vpn1
rd 100:1
route target both 100:1
hostname csc-pe1
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
```

```
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee send-community extended
 bgp dampening 30
 exit-address-family
 !
router bgp 100
. . .
! (BGP IPv4 to CSC-CE router from CSC-PE router)
!
 address-family ipv4 vrf vpn1
 neighbor ss.0.0.2 remote-as 200
 neighbor ss.0.0.2 activate
 neighbor ss.0.0.2 as-override
 neighbor ss.0.0.2 advertisement-interval 5
 neighbor ss.0.0.2 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
!
```

Configuring the Links Between CSC-PE and CSC-CE Routers Examples

This section contains the following examples:

Configuring the CSC-PE Routers Examples

The following example shows how to configure a CSC-PE router:

```
ip cef
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
mpls label protocol ldp
!
interface Loopback0
 ip address dd.dd.dd.dd 255.255.255.255
!
interface Ethernet3/1
 ip vrf forwarding vpn1
 ip address pp.0.0.2 255.0.0.0
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 mpls
 ip unnumbered Loopback0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
```

```
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/1
 network dd.dd.dd.dd 0.0.0.0 area 100
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
!
address-family vpnv4                                !VPNv4 session with CSC-PE2
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee send-community extended
 bgp dampening 30
 exit-address-family
!
 address-family ipv4 vrf vpn1
 neighbor pp.0.0.1 remote-as 200
 neighbor pp.0.0.1 activate
 neighbor pp.0.0.1 as-override
 neighbor pp.0.0.1 advertisement-interval 5
 neighbor pp.0.0.1 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Configuring the CSC-CE Routers Examples

The following example shows how to configure a CSC-CE router:

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address cc.cc.cc.cc 255.255.255.255
!
interface Ethernet3/0
 ip address pp.0.0.1 255.0.0.0
!
interface Ethernet4/0
 ip address nn.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets                     !Exchange routes
 redistribute bgp 200 metric 3 subnets              !learned from PE1
 passive-interface ATM1/0
 passive-interface Ethernet3/0
 network cc.cc.cc.cc 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
```

```
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor pp.0.0.2 remote-as 100
 neighbor pp.0.0.2 update-source Ethernet3/0
 no auto-summary
!
address-family ipv4
 redistribute connected
 redistribute ospf 200 metric 4 match internal
 neighbor pp.0.0.2 activate
 neighbor pp.0.0.2 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Verifying Labels in the CSC-PE Routers Examples

The following examples show how to verify the configurations of the CSC-PE routers.

Verify that the BGP session is up and running between the CSC-PE router and the CSC-CE router. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.

```
Router# show ip bgp vpnv4 all summary
BBGP router identifier 10.5.5.5, local AS number 100
BGP table version is 52, main routing table version 52
12 network entries and 13 paths using 2232 bytes of memory
6 BGP path attribute entries using 336 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 16/4 prefixes, 27/14 paths, scan interval 5 secs
Neighbor        V   AS     MsgRcvd MsgSent  TblVer InQ  OutQ Up/Down  State/PfxRcd
10.5.5.5        4  100     7685    7686      52    0       0 21:17:04       6
10.0.0.2        4  200     7676    7678      52    0       0 21:16:43       7
```

Verify that the MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. LDP is turned off on the VRF because EBGP distributes the labels.

```
Router# show mpls interfaces all
Interface              IP            Tunnel   Operational
GigabitEthernet6/0     Yes (ldp)     No       Yes
VRF vpn1:
Ethernet3/1            No            No       Yes
```

Verify that the prefix for the local PE router is in the routing table of the CSC-PE router:

```
Router# show ip route vrf vpn2 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 100", distance 20, metric 4
  Tag 200, type external
  Last update from pp.0.0.2 21:28:39 ago
  Routing Descriptor Blocks:
  * pp.0.0.2, from pp.0.0.2, 21:28:39 ago
      Route metric is 4, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix for the remote PE router is in the routing table of the CSC-PE router:

```
Router# show ip route vrf vpn2 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 100", distance 200, metric 4
  Tag 200, type internal
  Last update from 10.1.0.0 21:27:39 ago
  Routing Descriptor Blocks:
  * 10.1.0.0 (Default-IP-Routing-Table), from 10.1.0.0, 21:27:39 ago
      Route metric is 4, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefixes for the customer carrier MPLS VPN service provider networks are in the BGP table, and have appropriate labels:

```
Router# show ip bgp vpnv4 vrf vpn2 labels

   Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (vpn1)
   cc.cc.cc.cc/32   pp.0.0.2        22/imp-null
   bb.bb.bb.bb/32   pp.0.0.2        27/20
   hh.hh.hh.hh/32   ee.ee.ee.ee     34/35
   gg.gg.gg.gg/32   ee.ee.ee.ee     30/30
   nn.0.0.0         pp.0.0.2        23/imp-null
   ss.0.0.0         ee.ee.ee.ee     33/34
   pp.0.0.0         pp.0.0.2        25/aggregate(vpn1)
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.1.0.0
10.1.0.0/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 27
    fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
  via pp.0.0.2, 0 dependencies, recursive
    next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
    valid cached adjacency
    tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}

Router# show ip cef vrf vpn2 10.1.0.0 detail
10.1.0.0/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 27
    fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
  via pp.0.0.2, 0 dependencies, recursive
    next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
    valid cached adjacency
    tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 10.1.0.0
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
27     20          10.1.0.0/32[V]   958048     Et3/1       pp.0.0.2

Router# show mpls forwarding-table vrf vpn2 10.1.0.0 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
```

```
27     20 10.1.0.0/32[V]             958125     Et3/1      pp.0.0.2
        MAC/Encaps=14/18, MTU=1500, Tag Stack{20}
        00B04A74A05400B0C26E10558847 00014000
        VPN route: vpn1
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.3.0.0
10.3.0.0/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 34
    fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
  via ee.ee.ee.ee, 0 dependencies, recursive
    next hop rr.0.0.2, GigabitEthernet6/0 via ee.ee.ee.ee/32
    valid cached adjacency
    tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}

Router# show ip cef vrf vpn2 10.3.0.0 detail
hh.hh.hh.hh/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 34
    fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
  via ee.ee.ee.ee, 0 dependencies, recursive
    next hop rr.0.0.2, GigabitEthernet6/0 via ee.ee.ee.ee/32
    valid cached adjacency
    tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 10.3.0.0
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
34     35          hh.hh.hh.hh/32[V] 139034     Gi6/0      rr.0.0.2

Router# show mpls forwarding-table vrf vpn2 10.3.0.0 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
34     35          hh.hh.hh.hh/32[V] 139034     Gi6/0      rr.0.0.2
        MAC/Encaps=14/18, MTU=1500, Tag Stack{35}
        00B0C26E447000B0C26E10A88847 00023000
        VPN route: vpn1
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verifying Labels in the CSC-CE Routers Examples

The following examples show how to verify the configurations of the CSC-CE routers.

Verify that the BGP session is up and running:

```
Router# show ip bgp summary
BGP router identifier cc.cc.cc.cc, local AS number 200
BGP table version is 35, main routing table version 35
14 network entries and 14 paths using 2030 bytes of memory
3 BGP path attribute entries using 168 bytes of memory
```

```
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 1 history paths, 0 dampened paths
BGP activity 17/67 prefixes, 29/15 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
pp.0.0.1        4   100    7615    7613       35    0    0 21:06:19        5
```

Verify that the loopback address of the local PE router is in the routing table:

```
Router# show ip route 10.1.0.0
Routing entry for 10.1.0.0/32
  Known via "ospf 200", distance 110, metric 101, type intra area
  Redistributing via bgp 200
  Advertised by bgp 200 metric 4 match internal
  Last update from nn.0.0.1 on Ethernet4/0, 00:34:08 ago
  Routing Descriptor Blocks:
  * nn.0.0.1, from bb.bb.bb.bb, 00:34:08 ago, via Ethernet4/0
      Route metric is 101, traffic share count is 1
```

Verify that the loopback address of the remote PE router is in the routing table:

```
Router# show ip route 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 200", distance 20, metric 0
  Tag 100, type external
  Redistributing via ospf 200
  Advertised by ospf 200 metric 3 subnets
  Last update from pp.0.0.1 00:45:16 ago
  Routing Descriptor Blocks:
  * pp.0.0.1, from pp.0.0.1, 00:45:16 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2, BGP network version 0
```

Verify that the prefix of the local PE router is in the MPLS LDP bindings:

```
Router# show mpls ldp bindings 10.1.0.0 255.255.255.255
  tib entry: 10.1.0.0/32, rev 20
        local binding:  tag: 20
        remote binding: tsr: 10.1.0.0:0, tag: imp-null
```

Verify that the prefix of the local PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.1.0.0
10.1.0.0/32, version 46, cached adjacency nn.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 20
  via nn.0.0.1, Ethernet4/0, 0 dependencies
    next hop nn.0.0.1, Ethernet4/0
    unresolved
    valid cached adjacency
    tag rewrite with Et4/0, nn.0.0.1, tags imposed {}
```

Verify that the prefix of the local PE router is in the MPLS forwarding table:

```
Router# show mpls forwarding-table 10.1.0.0
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
20     Pop tag     bb.bb.bb.bb/32    893397     Et4/0      nn.0.0.1

Router# show mpls forwarding-table 10.1.0.0 detail
```

```
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
20     Pop tag     bb.bb.bb.bb/32    893524     Et4/0      nn.0.0.1
       MAC/Encaps=14/14, MTU=1504, Tag Stack{}
       00074F83685400B04A74A0708847
       No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verify that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks:

```
Router# show ip bgp labels
Network          Next Hop         In Label/Out Label
cc.cc.cc.cc/32   0.0.0.0          imp-null/exp-null
bb.bb.bb.bb/32   nn.0.0.1         20/exp-null
hh.hh.hh.hh/32   pp.0.0.1         26/34
gg.gg.gg.gg/32   pp.0.0.1         23/30
nn.0.0.0         0.0.0.0          imp-null/exp-null
ss.0.0.0         pp.0.0.1         25/33
pp.0.0.0         0.0.0.0          imp-null/exp-null
pp.0.0.1/32      0.0.0.0          16/exp-null
```

Verify that the prefix of the remote PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.5.5.5
10.5.5.5/32, version 54, cached adjacency pp.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 26
    fast tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}
  via pp.0.0.1, 0 dependencies, recursive
    next hop pp.0.0.1, Ethernet3/0 via pp.0.0.1/32
    valid cached adjacency
    tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}
```

Verify that the prefix of the remote PE router is in the MPLS forwarding table:

```
Router# show mpls forwarding-table 10.5.5.5
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     34          hh.hh.hh.hh/32    81786      Et3/0      pp.0.0.1

Router# show mpls forwarding-table 10.5.5.5 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     34          hh.hh.hh.hh/32    81863      Et3/0      pp.0.0.1
       MAC/Encaps=14/18, MTU=1500, Tag Stack{34}
       00B0C26E105500B04A74A0548847 00022000
       No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Configuring the Customer Carrier Network Examples

Customer carrier configuration and verification examples in this section include:

Verifying IP Connectivity in the Customer Carrier Example

Verify the connectivity from one customer carrier core router to another (from CE1 to CE2) by entering the following command:

```
Router# ping 10.2.0.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to jj.jj.jj.jj, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
```

Verify the path that a packet goes through on its way to its final destination from CE1 to CE2:

```
Router# trace 10.2.0.0
Type escape sequence to abort.
Tracing the route to 10.2.0.0
  1 mm.0.0.2 0 msec 0 msec 4 msec
  2 nn.0.0.2 [MPLS: Labels 20/21 Exp 0] 8 msec 8 msec 12 msec
  3 pp.0.0.2 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 12 msec
  4 ss.0.0.1 [MPLS: Labels 17/21 Exp 0] 8 msec 8 msec 12 msec
  5 ss.0.0.2 [MPLS: Labels 16/21 Exp 0] 8 msec 8 msec 12 msec
  6 tt.0.0.1 [AS 200] [MPLS: Label 21 Exp 0] 8 msec 8 msec 8 msec
  7 tt.0.0.2 [AS 200] 8 msec 4 msec *
```

Verify the path that a packet goes through on its way to its final destination from CE2 to CE1:

```
Router# trace 10.1.0.0
Type escape sequence to abort.
Tracing the route to 10.1.0.0
  1 tt.0.0.1 0 msec 0 msec 0 msec
  2 qq.0.0.2 [MPLS: Labels 18/21 Exp 0] 8 msec 12 msec 12 msec
  3 ss.0.0.1 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 8 msec
  4 pp.0.0.2 [MPLS: Labels 17/21 Exp 0] 12 msec 8 msec 8 msec
  5 pp.0.0.1 [MPLS: Labels 16/21 Exp 0] 12 msec 12 msec 8 msec
  6 mm.0.0.2 [AS 200] [MPLS: Label 21 Exp 0] 12 msec 8 msec 12 msec
  7 mm.0.0.1 [AS 200] 4 msec 4 msec *
```

Configuring a Customer Carrier Core Router as a Route Reflector Example

The following example shows how to use an address family to configure internal BGP peer 10.1.1.1 as a route-reflector client for both unicast and multicast prefixes:

```
router bgp 200
 address-family vpnv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client

router bgp 100
 address-family vpnv4
  neighbor xx.xx.xx.xx activate
  neighbor xx.xx.xx.xx route-reflector-client
   ! xx.xx.xx,xx is a PE router
  neighbor xx.xx.xx.xx send-community extended
 exit address-family
! You need to configure your peer BGP neighbor.
```

Configuring the Customer Site for Hierarchical VPNs Examples

This section contains the following configuration and verification examples for the customer site:

Configuring PE Routers for Hierarchical VPNs Examples

This example shows how to configure a PE router:

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Ethernet3/0
 ip address nn.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/3
 ip vrf forwarding vpn2
 ip address mm.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/3
 network bb.bb.bb.bb 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor hh.hh.hh.hh remote-as 200
 neighbor hh.hh.hh.hh update-source Loopback0
!
address-family vpnv4                                    !VPNv4 session with PE2
 neighbor hh.hh.hh.hh activate
 neighbor hh.hh.hh.hh send-community extended
 bgp dampening 30
 exit-address-family
!
 address-family ipv4 vrf vpn2
 neighbor mm.0.0.1 remote-as 300
 neighbor mm.0.0.1 activate
 neighbor mm.0.0.1 as-override
 neighbor mm.0.0.1 advertisement-interval 5
no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Verifying Labels in Each PE Router for Hierarchical VPNs Examples

The following examples show how to verify the configuration of PE router in hierarchical VPNs.

Verify that the loopback address of the local CE router is in the routing table of the PE1 router:

```
Router# show ip route vrf vpn2 10.2.2.2
```

```
Routing entry for 10.2.2.2/32
  Known via "bgp 200", distance 20, metric 0
  Tag 300, type external
  Last update from mm.0.0.2 20:36:59 ago
  Routing Descriptor Blocks:
  * mm.0.0.2, from mm.0.0.2, 20:36:59 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix for the local CE router is in the MPLS forwarding table, and that the prefix is untagged:

```
Router# show mpls forwarding-table vrf vpn2 10.2.2.2
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
23     Untagged    aa.aa.aa.aa/32[V] 0          Et3/3      mm.0.0.2
```

Verify that the prefix of the remote PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.5.5.5

10.5.5.5/32, version 31, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 31
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
  via nn.0.0.2, Ethernet3/0, 2 dependencies
    next hop nn.0.0.2, Ethernet3/0
    unresolved
    valid cached adjacency
    tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
```

Verify that the loopback address of the remote CE router is in the routing table:

```
Router# show ip route vrf vpn2 10.2.0.0
Routing entry for 10.2.0.0/32
  Known via "bgp 200", distance 200, metric 0
  Tag 300, type internal
  Last update from hh.hh.hh.hh 20:38:49 ago
  Routing Descriptor Blocks:
  * hh.hh.hh.hh (Default-IP-Routing-Table), from hh.hh.hh.hh, 20:38:49 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix of the remote CE router is in the MPLS forwarding table, and that an outgoing interface exists:

```
Router# show mpls forwarding-table vrf vpn2 10.2.0.0
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
None   26          jj.jj.jj.jj/32    0          Et3/0      nn.0.0.2
```

Verify that the prefix of the remote CE router is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.2.0.0
10.2.0.0/32, version 12, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: VPN route head
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
  via hh.hh.hh.hh, 0 dependencies, recursive
    next hop nn.0.0.2, Ethernet3/0 via hh.hh.hh.hh/32
```

```
    valid cached adjacency
    tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
```

Verify that the prefix of the local PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.1.0.0
10.1.0.0/32, version 9, connected, receive
  tag information set
    local tag: implicit-null
```

Configuring CE Routers for Hierarchical VPNs Examples

The following example shows how to configure a CE router:

```
ip cef distributed
interface Loopback0
ip address 10.3.0.0 255.255.255.255
!
interface FastEthernet0/3/3
 ip address mm.0.0.1 255.0.0.0
!
router bgp 300
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected                          !Redistributing routes into BGP
 neighbor mm.0.0.2 remote-as 200                 !to send to PE1
 neighbor mm.0.0.2 advertisement-interval 5
 no auto-summary
```

Verifying IP Connectivity in the Customer Site Examples

The following examples show how to verify IP connectivity at the customer site.

Verify that the loopback address of the remote CE router, learned from the PE router, is in the routing table of the local router:

```
Router# show ip route 10.2.0.0
Routing entry for 10.2.0.0/32
  Known via "bgp 300", distance 20, metric 0
  Tag 200, type external
  Redistributing via ospf 300
  Advertised by ospf 300 subnets
  Last update from mm.0.0.1 20:29:35 ago
  Routing Descriptor Blocks:
  * mm.0.0.1, from mm.0.0.1, 20:29:35 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
```

Additional References

Related Documents

Related Topic	Document Title
LDP	MPLS Label Distribution Protocol

Related Topic	Document Title
MPLS	MPLS Product Literature

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 1171	*A Border Gateway Protocol 4*
RFC 1700	*Assigned Numbers*
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh IBGP*
RFC 2283	*Multiprotocol Extensions for BGP-4*
RFC 2547	*BGP/MPLS VPNs*
RFC 2842	*Capabilities Advertisement with BGP-4*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3107	*Carrying Label Information in BGP-4*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS VPN CSC with BGP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 155: Feature Information for MPLS VPN CSC with BGP

Feature Name	Releases	Feature Information
MPLS VPN--Carrier Supporting Carrier--IPv4 BGP Label Distribution	12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(27)S 12.0(29)S Cisco IOS XE Release 2.2	This feature enables you to create an MPLS VPN CSC network that uses BGP to transport routes and MPLS labels. In 12.0(21)ST, this feature was introduced. In 12.0(22)S, this feature was integrated. In 12.0(23)S, this feature was integrated. In 12.2(13)T, this feature was integrated. 12.0(24)S, this feature was integrated. In 12.2(14)S, this feature was integrated. In 12.0(27)S, this feature was integrated. In 12.0(29)S, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.

Glossary

ASBR -- Autonomous System Boundary router. A router that connects one autonomous system to another.

autonomous system --A collection of networks under a common administration sharing a common routing strategy.

BGP --Border Gateway Protocol. An interdomain routing protocol that exchanges network reachability information with other BGP systems (which may be within the same autonomous system or between multiple autonomous systems).

CE router--customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers do not recognize associated MPLS VPNs.

CSC --Carrier Supporting Carrier. A hierarchical VPN model that allows small service providers, or customer carriers, to interconnect their IP or MPLS networks over an MPLS backbone. This eliminates the need for customer carriers to build and maintain their own MPLS backbone.

eBGP --external Border Gateway Protocol. A BGP between routers located within different autonomous systems. When two routers, located in different autonomous systems, are more than one hop away from one another, the eBGP session between the two routers is considered a multihop BGP.

edge router--A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

iBGP --internal Border Gateway Protocol. A BGP between routers within the same autonomous system.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within a single autonomous system. Examples of common Internet IGP protocols include IGRP, OSPF, IS-IS, and RIP.

IP --Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

LFIB --Label Forwarding Information Base. Data structure used in MPLS to hold information about incoming and outgoing labels and associated Forwarding Equivalence Class (FEC) packets.

MP-BGP --Multiprotocol BGP.

MPLS --Multiprotocol Label Switching. The name of the IETF working group responsible for label switching, and the name of the label switching approach it has standardized.

NLRI --Network Layer Reachability Information. The BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and extended community values.

NSF --Nonstop forwarding enables routers to continuously forward IP packets following a Route Processor takeover or switchover to another Route Processor. NSF maintains and updates Layer 3 routing and forwarding information in the backup Route Processor to ensure that IP packets and routing protocol information are forwarded continuously during the switchover and route convergence process.

PE router--provider edge router. A router that is part of a service provider's network. It is connected to a customer edge (CE) router. All MPLS VPN processing occurs in the PE router.

QoS --quality of service. Measure of performance for a transmission system that indicates its transmission quality and service availability.

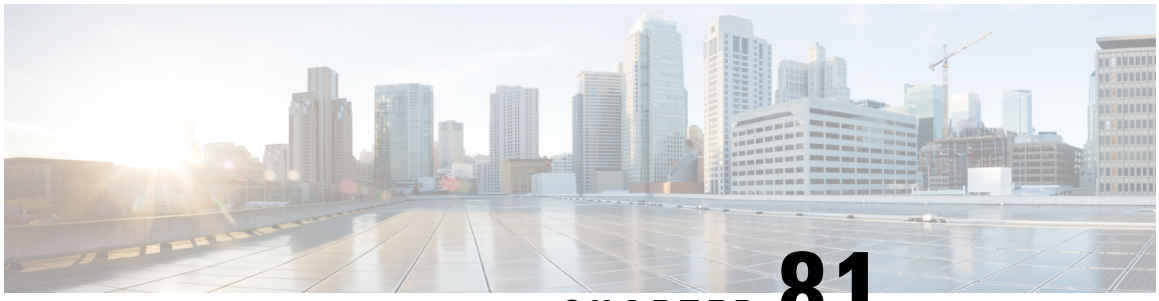
RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is imported.

SLA --Service Level Agreement given to VPN subscribers.

VPN --Virtual Private Network. A secure MPLS-based network that shares resources on one or more physical networks (typically implemented by one or more service providers). A VPN contains geographically dispersed sites that can communicate securely over a shared backbone network.

VRF --VPN routing and forwarding instance. Routing information that defines a VPN site that is attached to a PE router. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.



CHAPTER 81

MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

The MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs feature allows MPLS VPN interautonomous (Inter-AS) and MPLS VPN Carrier Supporting Carrier (CSC) networks to load share traffic between adjacent label switch routers (LSRs) that are connected by multiple links. The LSRs can be a pair of Autonomous System Boundary Routers (ASBRs) or a CSC-provider edge (PE) and a CSC-customer edge (CE) device. Using directly connected loopback peering allows load sharing at the Interior Gateway Protocol (IGP) level so only one Border Gateway Protocol (BGP) session is needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs except BGP.

Prerequisites for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Ensure that your Multiprotocol Label Switching (MPLS) virtual private network (VPN) network, including MPLS VPN interautonomous system (Inter-AS) or Carrier Supporting Carrier (CSC), is configured and working properly.

Restrictions for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Load sharing using directly connected loopback peering does not apply to Carrier Supported Carrier (CSC) networks that use the Label Distribution Protocol (LDP) and an Interior Gateway Protocol (IGP) to distribute routes and Multiprotocol Label Switching (MPLS) labels.

The software does not support load balancing in interautonomous system (Inter-AS) and CSC when there are multiple links between provider edge (PE) or Autonomous System Boundary Router (ASBR) devices.

When you configure static routes in an MPLS or MPLS virtual private network (VPN) environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in Cisco software releases that support the Tag Forwarding Information Base (TFIB. The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in Cisco software releases that support the MPLS Forwarding Infrastructure (MFI). Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in an MPLS environment:

- **ip route** *destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask interface1 next-hop1*
- **ip route** *destination-prefix mask interface2 next-hop2*

Unsupported Static Routes in an MPLS Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment and enable load sharing where the next hop can be reached through two paths:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment and enable load sharing where the destination can be reached through two next hops:

- **ip route** *destination-prefix mask next-hop1*
- **ip route** *destination-prefix mask next-hop2*

Use the *interface* and *next-hop* arguments when specifying static routes.

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are associated with the same virtual routing and forwarding (VRF) instance:

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface1 next-hop1*
- **ip route vrf** *vrf-name destination-prefix mask interface2 next-hop2*

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the internet gateway.

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address* **global**
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and the interface are in the core.)

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interfaces:

- **ip route** *destination-prefix mask interface1 next-hop1*
- **ip route** *destination-prefix mask interface2 next-hop2*

Unsupported Static Routes in an MPLS VPN Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

- **ip route vrf** *destination-prefix mask next-hop-address* **global**

The following **ip route** commands are not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

- **ip route vrf** *destination-prefix mask next-hop1* **global**
- **ip route vrf** *destination-prefix mask next-hop2* **global**

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

- **ip route vrf** *vrf-name destination-prefix mask next-hop1*
- **ip route vrf** *vrf-name destination-prefix mask next-hop2*

Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Device

The following **ip route vrf** command is supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table on the customer edge (CE) side. For example, the following command is supported when the destination-prefix is the CE device's loopback address, as in external Border Gateway Protocol (eBGP) multihop cases.

- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static nonrecursive routes and a specific outbound interfaces:

- **ip route** *destination-prefix mask interface1 nexthop1*
- **ip route** *destination-prefix mask interface2 nexthop2*

Information About MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Load Sharing Using Directly Connected Loopback Peering

You use the MPLS VPN Load Balancing Support for Inter-AS and CSC VPN feature to load share traffic between adjacent label switched routers (LSRs) that are connected by multiple links. The LSRs could be a pair of Autonomous System Boundary Routers (ASBRs) or a carrier supporting carrier provider edge (CSC-PE) and a CSC-customer edge (CE).

Using directly connected loopback peering allows load sharing at the Interior Gateway Protocol (IGP) level so only one Border Gateway Protocol (BGP) session is needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs except BGP.

Directly connected loopback peering enables load sharing of traffic as follows:

- A BGP session is established, using the loopback addresses of the LSRs.
- Multiprotocol Label Switching (MPLS) is enabled on the connecting links.
- Multiple static routes to the loopback address of the adjacent LSR allow IGP load sharing.
- The outgoing label to the loopback address of the adjacent LSR is an implicit null label and is inferred by the LSR.
- Because IGP load sharing is enabled on the loopback address of the adjacent LSR, any traffic destined to a prefix that is learned over the BGP session (and recurses over the loopback) is load shared.

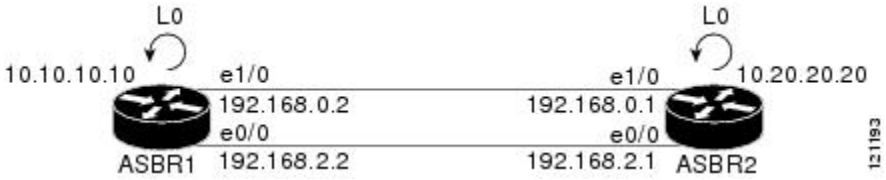
How to Configure MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS using ASBRs to Exchange VPN-IPv4 Addresses

This section describes the following tasks you need to do to configure peering of loopback interfaces of directly connected Autonomous System Boundary Routers (ASBRs):

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2. This configuration is used as the example in the tasks that follow.

Figure 138: Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2



Configuring Loopback Interface Addresses for Directly Connected ASBRs

Perform this task to configure loopback interface addresses for directly connected Autonomous System Boundary Routers (ASBRs).

Note Loopback addresses need to be configured for each directly connected ASBR. That is, configure a loopback address for ASBR1 and for ASBR2 in the example shown in the figure above.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface- number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface- number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up and enters interface configuration mode. • The **interface-number** argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address mask* [**secondary**]	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
	Example: `Device(config-if)# ip address 10.10.10.10 255.255.255.255`	• The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the external Border Gateway Protocol (eBGP) neighbor loopback.

Note You need to configure /32 static routes on each of the directly connected ASBRs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]	Establishes static routes.

	Command or Action	Purpose
	Example: `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	• The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the specified network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a “match” value for controlling redistribution through the use of route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on Connecting Loopback Interfaces

Perform this task to configure forwarding on the connecting loopback interfaces.

This task is required for sessions between loopbacks. In the “Configuring /32 Static Routes to the eBGP Neighbor Loopback" section, Ethernet 1/0 and Ethernet 0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Perform this task to configure an external Border Gateway Protocol (eBGP) session between the loopbacks.

Note You need to configure an eBGP session between loopbacks on each directly connected Autonomous System Boundary Router (ASBR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family vpnv4** [**unicast**]
9. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
10. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard extended**]
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the BGP routing process. • The *as-number* indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**no bgp default route-target filter** **Example:** `Device(config)# no bgp default route-target filter`	Disables BGP route-target filtering, and enters router configuration mode. • All received BGP VPN-IPv4 routes are accepted by the device.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.

	Command or Action	Purpose
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **unicast** keyword specifies unicast prefixes.
Step 9	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of a BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard extended**]	Specifies that a communities attribute should be sent to a BGP neighbor.

	Command or Action	Purpose
	Example: `Device(config-router-af)# neighbor 10.20.20.20 send-community extended`	• The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 11	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

Perform this task to verify that load sharing occurs between loopbacks. You need to ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** {*mask* | *length*} | **labels** *label* [*network label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** {*mask*	*length*}	**labels** *label* [*network label*]	**interface** *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter an optional keyword or argument if desired.

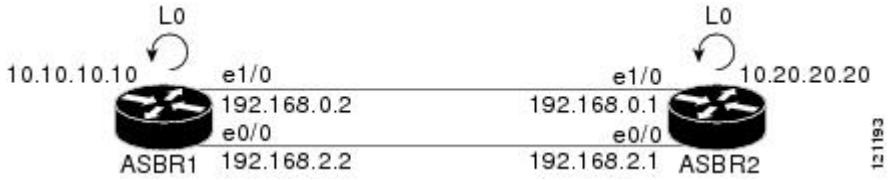
	Command or Action	Purpose
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS Using ASBRs to Exchange IPv4 Routes and Labels

The following sections describe how to configure peering of loopback interfaces of directly connected Autonomous System Boundary Routers (ASBRs) to achieve load sharing in an interautonomous system network:

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2. This configuration is used as the example in the tasks that follow.

Figure 139: Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2



Configuring Loopback Interface Addresses for Directly Connected ASBRs

Note Loopback addresses need to be configured for each directly connected Autonomous System Boundary Router (ASBR). That is, configure a loopback address for ASBR1 and for ASBR2 as in the example shown in the figure above.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface number*
4. **ip address** *ip-address* [*mask* [**secondary**]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up and enters interface configuration mode. • The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address* [*mask* [**secondary**]] **Example:** `Device(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the external Border Gateway Protocol (eBGP) neighbor loopback.

Note You need to configure /32 static routes on each of the directly connected Autonomous System Boundary Routers (ASBRs).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]

4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	Establishes static routes. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the specified network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a "match" value for controlling redistribution through the use of route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on Connecting Loopback Interfaces

This task is required for sessions between loopbacks. In the "Configuring /32 Static Routes to the eBGP Neighbor Loopback" task, Ethernet1/0 and Ethernet0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Note You need to configure an external Border Gateway Protocol (eBGP) session between loopbacks on each directly connected Autonomous System Boundary Router (ASBR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
9. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
10. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the BGP routing process, and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other

	Command or Action	Purpose
		BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the number of the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `Device(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *ttl* argument the time-to-live in the range from 1 to 255 hops.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.

	Command or Action	Purpose
		• The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 9	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a VPN routing/forwarding instance (VRF) to associate with submode commands.
Step 10	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.

	Command or Action	Purpose
Step 12	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

To verify that load sharing can occur between loopbacks, ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [*network* {*mask* |*length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
3. **disable**

DETAILED STEPS

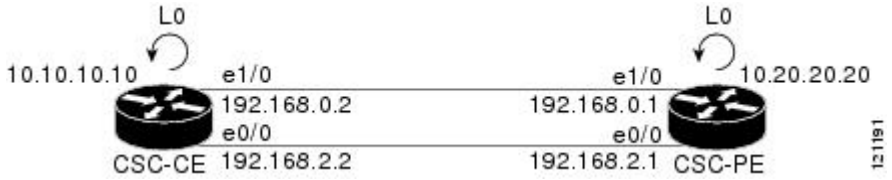
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** [*network* {*mask* |*length*}	**labels** *label* [*label*]	**interface** *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter a keyword or argument, if desired.
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuring Directly Connected Loopback Peering on MPLS VPN Carrier Supporting Carrier

The following sections explain how to load balance Carrier Supporting Carrier (CSC) traffic by peering loopback interfaces of directly connected CSC-provider edge (PE) and CSC-customer edge (CE) devices:

The figure below shows the loopback configuration for directly connected CSC-PE and CSC-CE devices. This configuration is used as the example in the tasks that follow.

Figure 140: Loopback Interface Configuration for Directly Connected CSC-PE and CSC-CE Devices



Configuring Loopback Interface Addresses on CSC-PE Devices

Note Configuration of a loopback interface address on the Carrier Supporting Carrier (CSC)-provider edge (PE) device requires the enabling of a virtual routing and forwarding (VRF) instance. The CSC-customer edge (CE) device loopback interface does not require enabling a VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface number*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up, and enters interface configuration mode. • The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.

	Command or Action	Purpose
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.20.20.20 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Loopback Interface Addresses for CSC-CE Routers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface-number* **Example:**	Configures a software-only virtual interface that emulates an interface that is always up.

	Command or Action	Purpose
	`Device(config)# interface loopback 0`	• The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route vrf** *vrf-name prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**global**] [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route vrf** *vrf-name prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**global**] [*distance*] [*name*] [**permanent**] [**tag** *tag*]	Establishes static routes for a virtual routing and forwarding (VRF) instance.

	Command or Action	Purpose
	Example: `Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet 1/0 172.16.0.2`	• The *vrf-name* argument is the name of the VRF for the static route. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the destination network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The **global** keyword specifies that the given next hop address is in the nonVRF routing table. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a "match" value for controlling redistribution via route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-CE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	Establishes static routes. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the destination network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword and argument name a tag value that can be used as a "match" value for controlling redistribution via route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-PE Interfaces That Connect to the CSC-CE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **mpls bgp forwarding**
7. **exit**
8. Repeat Steps 3 through 6 for another connecting interface (Ethernet 0/0).

9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a virtual routing and forwarding (VRF) instance with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.16.0.1 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	Repeat Steps 3 through 6 for another connecting interface (Ethernet 0/0).	
Step 9	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-CE Interfaces That Connect to the CSC-PE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *typeslot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *typeslot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information.

	Command or Action	Purpose
		• The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the CSC-PE Device and the CSC-CE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
9. **ip vrf forwarding** *vrf-name*
10. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
11. **neighbor** *ip-address* **send-label**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the Border Gateway Protocol (BGP) routing process. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.10.10.10 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.10.10.10 disable-connected-check`	Allows peering between loopbacks. • The *ip-address*argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.10.10.10 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.

	Command or Action	Purpose
		• The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **ipv4** keyword configures sessions that carry standard IPv4 address prefixes. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a virtual routing and forwarding (VRF) instance to associate with submode commands.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-router-af)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 10	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 11	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 send-label`	Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device. • The *ip-address* argument is the IP address of the neighboring device.
Step 12	**end** **Example:**	Exits to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configuring an eBGP Session Between the CSC-CE Device and the CSC-PE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family ipv4** [**unicast**] [**vrf** *vrf-name*]
9. **neighbor** {*ip-address* | *peer-group-name*|*ipv6-address*] **activate**
10. **neighbor** *ip-address* **send-label**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the Border Gateway Protocol (BGP) routing process. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*	Adds an entry to the BGP or multiprotocol BGP neighbor table.

	Command or Action	Purpose
	Example: `Device(config-router)# neighbor 10.20.20.20 remote-as 100`	• The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family ipv4** [**unicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing protocols such as BGP, RIP, and static routing. • The **ipv4** keyword configures sessions that carry standard IPv4 address prefixes. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a virtual routing and forwarding (VRF) instance to associate with submode commands.
Step 9	**neighbor** {*ip-address* | *peer-group-name*|*ipv6-address*] **activate** **Example:**	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 10.20.20.20 activate`	• The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 10	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 send-label`	Enables a BGP device to send Multiprotocol Label Switching (MPLS) labels with BGP routes to a neighboring BGP device. • The *ip-address* argument is the IP address of the neighboring device.
Step 11	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

To verify that load sharing occurs between loopbacks, ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | [**interface**] *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask*	*length*}	**labels** *label* [*-label*]	[**interface**] *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:**	Displays the contents of the MPLS LFIB.

	Command or Action	Purpose
	`Device# show mpls forwarding-table`	
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuration Examples for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Examples: Configuring a 32 Static Route from an ASBR to the Loopback Address of Another ASBR

The following example configures a /32 static route from ASBR1 to the loopback address of ASBR2:

```
Device# configure terminal
Device(config)# ip route 10.20.20.20 255.255.255.255 e1/0 168.192.0.1
Device(config)# ip route 10.20.20.20 255.255.255.255 e0/0 168.192.2.1
```

The following example configures a /32 static route from ASBR2 to the loopback address of ASBR1:

```
Device# configure terminal
Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e1/0 168.192.0.2
Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e0/0 168.192.2.2
```

Example: Configuring BGP MPLS Forwarding on the Interfaces Connecting ASBRs

The following example configures the Border Gateway Protocol (BGP) and Multiprotocol Label Switching (MPLS) forwarding on the interfaces connecting ASBR2 with ASBR1:

```
Device# configure terminal
Device(config)# interface ethernet 1/0
Device(config-if)# ip vrf forwarding vpn1
Device(config-if)# ip address 168.192.0.1 255.255.255.255
Device(config-if)# mpls bgp forwarding
Device(config-if)# exit
Device(config)# interface ethernet 0/0
Device(config-if)# ip vrf forwarding vpn1
Device(config-if)# ip address 168.192.2.1 255.255.255.255
Device(config-if)# mpls bgp forwarding
Device(config-if)# exit
```

Example: Configuring VPNv4 Sessions on an ASBR

The following example configures VPNv4 sessions on ASBR2:

```
Device# configure terminal
Device(config)# router bgp 200
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# neighbor 10.10.10.10 remote-as 100
Device(config-router)# neighbor 10.10.10.10 disable-connected-check
Device(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255
Device(config-router)# neighbor 10.10.10.10 update-source Loopback0
!
Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor 10.10.10.10 activate
Device(config-router-af)# neighbor 10.10.10.10 send-community extended
Device(config-router-af)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring MPLS VPN CSC with BGP	"MPLS VPN Carrier Supporting Carrier with BGP" module in the *MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide*
Configuring BGP	"Configuring BGP" module in the *IP Routing: BGP Configuration Guide*
Configuring BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN" module in the *IP Routing: BGP Configuration Guide*

RFCs

RFC	Title
RFC 1164	Application of the Border Gateway Protocol in the Internet
RFC 1171	A Border Gateway Protocol 4
RFC 1700	Assigned Numbers
RFC 1966	BGP Route Reflection: An Alternative to Full Mesh IBGP
RFC 2283	Multiprotocol Extensions for BGP-4
RFC 2373	IP Version 6 Addressing Architecture

RFC	Title
RFC 2547	BGP/MPLS VPNs
RFC 2842	Capabilities Advertisement with BGP-4
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3107	Carrying Label Information in BGP-4

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

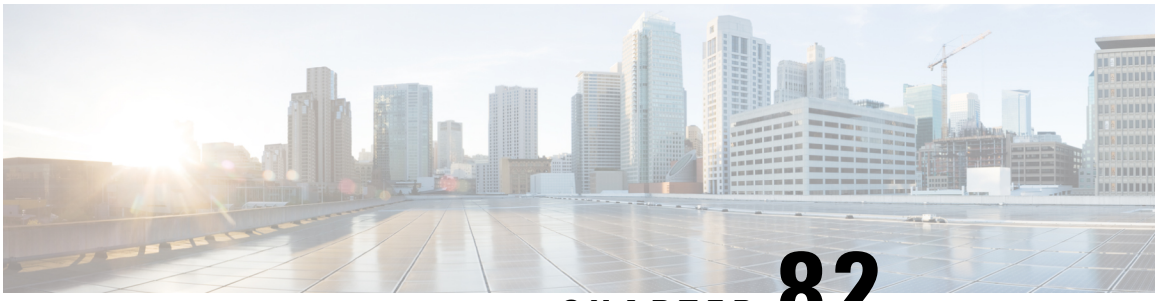
Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 156: Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Feature Name	Releases	Feature Information
MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs	12.0(29)S 12.4(20)T 12.2(33)SRA 12.2(33)SXH Cisco IOS XE Release 2.2	The MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs feature allows MPLS VPN Inter-AS and MPLS VPN CSC networks to load share traffic between adjacent LSRs that are connected by multiple links. The LSRs can be a pair of ASBRs or a CSC-PE and a CSC-CE. Using directly connected loopback peering allows load sharing at the IGP level, so more than one BGP session is not needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs than BGP. In Cisco IOS Release 12.0(29)S, this feature was introduced. In Cisco IOS Release 12.4(20)T, 12.2(33)SRA, and 12.2(33)SXH, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. No commands were introduced or modified.



CHAPTER 82

MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

The MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs feature enables you to configure external Border Gateway Protocol (eBGP) multipath with IPv4 labels. This creates an entry in the Multiprotocol Label Switching (MPLS) forwarding table with label information for each outgoing path installed in the routing table thereby allowing redundant connectivity and load balancing. Without this feature, the MPLS forwarding table contains the labels only for the BGP best path even though the routing table has more than one path for the prefix.

Prerequisites for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Ensure that your Multiprotocol Label Switching (MPLS) virtual private network (VPN) network, including MPLS VPN interautonomous system (Inter-AS) or Carrier Supporting Carrier (CSC), is configured and working properly.

Restrictions for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

The MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs feature is not supported on Multiprotocol Label Switching (MPLS) virtual private network (VPN) interautonomous system (Inter-AS) with Autonomous System Boundary Routers (ASBRs) that exchange VPNv4 routes.

When you configure static routes in an MPLS or MPLS virtual private network (VPN) environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in Cisco software releases that support the Tag Forwarding Information Base (TFIB. The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in Cisco software releases that support the MPLS Forwarding Infrastructure (MFI). Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in an MPLS environment:

- **ip route** *destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask interface1 next-hop1*
- **ip route** *destination-prefix mask interface2 next-hop2*

Unsupported Static Routes in an MPLS Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment and enable load sharing where the next hop can be reached through two paths:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment and enable load sharing where the destination can be reached through two next hops:

- **ip route** *destination-prefix mask next-hop1*
- **ip route** *destination-prefix mask next-hop2*

Use the *interface* and *next-hop* arguments when specifying static routes.

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are associated with the same virtual routing and forwarding (VRF) instance:

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface1 next-hop1*
- **ip route vrf** *vrf-name destination-prefix mask interface2 next-hop2*

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the internet gateway.

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address* **global**
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and the interface are in the core.)

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interfaces:

- **ip route** *destination-prefix mask interface1 next-hop1*
- **ip route** *destination-prefix mask interface2 next-hop2*

Unsupported Static Routes in an MPLS VPN Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

- **ip route vrf** *destination-prefix mask next-hop-address* **global**

The following **ip route** commands are not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

- **ip route vrf** *destination-prefix mask next-hop1* **global**
- **ip route vrf** *destination-prefix mask next-hop2* **global**

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

- **ip route vrf** *vrf-name destination-prefix mask next-hop1*
- **ip route vrf** *vrf-name destination-prefix mask next-hop2*

Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Device

The following **ip route vrf** command is supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table on the customer edge (CE) side. For example, the following command is supported when the destination-prefix is the CE device's loopback address, as in external Border Gateway Protocol (eBGP) multihop cases.

- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static nonrecursive routes and a specific outbound interfaces:

- **ip route** *destination-prefix mask interface1 nexthop1*
- **ip route** *destination-prefix mask interface2 nexthop2*

Information About MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Overview of MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

When a device learns two identical external Border Gateway Protocol (eBGP) paths for a prefix from a neighboring autonomous system, it chooses the path with the lower route ID as the best path. This best path is installed in the IP routing table. You can enable eBGP multipath, which installs multiple paths in the IP routing table (instead of picking one best path) when the eBGP paths are learned from a neighboring autonomous system.

During packet switching, depending on the switching mode, either per-packet or per-destination load sharing is performed among the multiple paths. The **maximum-paths** router configuration command controls the number of paths allowed. By default, BGP installs only one path to the IP routing table.

How to Configure MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Configuring MPLS VPN eBGP Multipath Load Sharing with Inter-AS MPLS VPNs

Perform this task on the Autonomous System Boundary Routers (ASBRs) to configure external Border Gateway Protocol (eBGP) multipath for Multiprotocol Label Switching (MPLS) virtual private network (VPN) interautonomous systems with ASBRs exchanging IPv4 routes and MPLS labels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **maximum-paths** *number-paths*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**

8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a BGP routing process and places the device in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 6	**maximum-paths** *number-paths* **Example:** `Device(config-router-af)# maximum-paths 2`	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. • The *number-paths* argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring device. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 send-label`	Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighboring device.
Step 9	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring MPLS VPN eBGP Multipath Load Sharing with Carrier Supporting Carrier on the CSC-PE Devices

Perform this task to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-provider edge (CSC-PE) devices that distribute BGP routes with Multiprotocol Label Switching (MPLS) labels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **maximum-paths** *number-paths*
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*

7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** *ip-address* **as-override**
9. **neighbor** *ip-address* **send-label**
10. **exit-address-family**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**maximum-paths** *number-paths* **Example:** `Device(config-router-af)# maximum-paths 2`	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. • On the CSC-PE device, this command is enabled in address family configuration mode. • The *number-paths* argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table.

	Command or Action	Purpose
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** *ip-address* **as-override** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 as-override`	Configures a PE device to override the autonomous system number (ASN) of a site with the ASN of a provider. • The *ip-address* argument specifies the IP address of the device that is to be overridden with the ASN provided.
Step 9	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 send-label`	Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighboring device.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**end** **Example:** `Device(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring MPLS VPN eBGP Multipath Load Sharing with Carrier Supporting Carrier on the CSC-CE Devices

Perform this task to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-customer edge (CSC-CE) devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **maximum-paths** *number-paths*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **redistribute** *protocol*
7. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
8. **neighbor** {*ip-address* | *peer-group-name*} **activate**
9. **neighbor** *ip-address* **send-label**
10. **exit-address-family**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**maximum-paths** *number-paths* **Example:** `Device(config-router)# maximum-paths 2`	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. • On the CSC-CE routers, this command is issued in router configuration mode. • The *number-paths* argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table.
Step 5	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] **Example:**	Specifies the IPv4 address family type and enters address family configuration mode.

	Command or Action	Purpose
	`Device(config-router)# address-family ipv4`	• The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**redistribute** *protocol* **Example:** `Device(config-router-af)# redistribute static`	Redistributes routes from one routing domain into another routing domain. • The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **mobile**, **ospf**, **rip**, and **static** [**ip**]. • The **static** [**ip**] keyword redistributes IP static routes. **Note** The optional **ip** keyword is used when you redistribute static routes into Intermediate System- to-Intermediate System (IS-IS). • • The **connected** keyword refers to routes that are established automatically when IP is enabled on an interface. • For routing protocols such as Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS), these routes are redistributed as external to the autonomous system.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.2 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 activate`	Enables the exchange of information with a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighbor.

	Command or Action	Purpose
		• The *peer-group-name* argument specifies the name of a BGP peer group.
Step 9	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 send-label`	Enables a BGP device to send Multiprotocol Label Switching (MPLS) labels with BGP routes to a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighboring device.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**end** **Example:** `Device(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuration Examples for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Inter-AS

The following example shows how to configure external Border Gateway Protocol (eBGP) multipath for Multiprotocol Label Switching (MPLS) virtual private network (VPN) interautonomous systems with Autonomous System Boundary Routers (ASBRs) exchanging IPv4 routes and MPLS labels:

```
Device# configure terminal
Device(config)# router bgp 100
Device(config-router)# neighbor 10.0.0.1 remote-as 200
Device(config-router)# address-family ipv4
Device(config-router-af)# maximum-paths 2
Device(config-router-af)# neighbor 10.0.0.1 activate
Device(config-router-af)# neighbor 10.0.0.1 send-label
Device(config-router-af)# exit-address-family
Device(config-router-af)# end
```

Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-PE Devices

The following example shows how to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-provider edge (CSC-PE) devices that distribute BGP routes with Multiprotocol Label Switching (MPLS) labels:

```
Device# configure terminal
Device(config)# router bgp 100
Device(config-router)# address-family ipv4 vrf vpn1
Device(config-router-af)# maximum-paths 2
Device(config-router-af)# neighbor 10.0.0.1 remote-as 200
Device(config-router-af)# neighbor 10.0.0.1 activate
Device(config-router-af)# neighbor 10.0.0.1 as-override
Device(config-router-af)# neighbor 10.0.0.1 send-label
Device(config-router-af)# exit-address-family
Device(config-router)# end
```

Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-CE Devices

The following example shows how to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-customer edge (CSC-CE) devices:

```
Device# configure terminal
Device(config)# router bgp 200
Device(config-router)# maximum-paths 2
Device(config-router)# address-family ipv4
Device(config-router-af)# redistribute static
Device(config-router-af)# neighbor 10.0.0.2 remote-as 100
Device(config-router-af)# neighbor 10.0.0.2 activate
Device(config-router-af)# neighbor 10.0.0.2 send-label
Device(config-router-af)# exit-address-family
Device(config-router)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring MPLS VPN CSC with BGP	"MPLS VPN Carrier Supporting Carrier with BGP" module in the *MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide*

Related Topic	Document Title
Configuring BGP	"Configuring BGP" module in the *IP Routing: BGP Configuration Guide*
Configuring BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN" module in the *IP Routing: BGP Configuration Guide*

RFCs

RFC	Title
RFC 1164	Application of the Border Gateway Protocol in the Internet
RFC 1171	A Border Gateway Protocol 4
RFC 1700	Assigned Numbers
RFC 1966	BGP Route Reflection: An Alternative to Full Mesh IBGP
RFC 2283	Multiprotocol Extensions for BGP-4
RFC 2373	IP Version 6 Addressing Architecture
RFC 2547	BGP/MPLS VPNs
RFC 2842	Capabilities Advertisement with BGP-4
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3107	Carrying Label Information in BGP-4

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

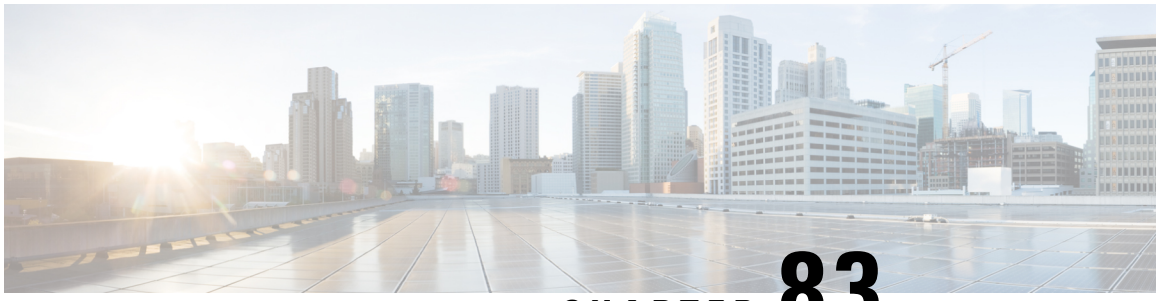
Feature Information for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 157: Feature Information for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Feature Name	Releases	Feature Information
MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs	12.0(27)S 12.2(30)S 12.2(33)SRA 12.2(33)SXH Cisco IOS XE Release 2.2	The MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs feature installs multiple paths in the IP routing table when the eBGP paths are learned from a neighboring Autonomous System (AS), instead of picking one best path. In Cisco IOS Release 12.0(27)S, this feature was introduced. In Cisco IOS Release 12.2(30)S, 12.2(33)SRA, and 12.2(33)SXH, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. No commands were introduced or modified.



CHAPTER 83

MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

The MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session feature provides a method to advertise explicit null in a Border Gateway Protocol (BGP) label session for a carrier supporting carrier (CSC) customer edge (CE) device.

Prerequisites for MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

- You must configure your network for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN).
- You must configure the Border Gateway Protocol (BGP) to distribute labels between the carrier supporting carrier (CSC) customer edge (CE) device (CSC-CE) and CSC-provider edge (PE) devices.

Restrictions for MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

- Configure an explicit null label only in a carrier supporting carrier (CSC) customer edge (CE) device (CSC-CE) topology.

- Configure an explicit null label only on a per-neighbor basis.

Information About MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

Feature Design of MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

On a carrier supporting carrier (CSC) customer edge (CE) device (CSC-CE) with Border Gateway Protocol (BGP) IPv4 label distribution, BGP advertises an implicit null label for directly connected routes. This causes the previous hop (penultimate) device to do penultimate hop popping (PHP).

The MPLS VPN Explicit Null Label Support BGP IPv4 Label Session feature makes the penultimate device swap the incoming label for (or impose) the explicit null label. This action forces the egress device to process the explicit null label by popping it and inspecting the packet that remains.

Benefits of MPLS VPN Explicit Null Label Support BGP IPv4 Label Session

The explicit null label helps to preserve quality of service (QoS) bits from one Service Level Agreement (SLA) to another until the packets reach their carrier supporting carrier (CSC) customer edge (CE) destination.

How to Configure MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

Configuring CSC with BGP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast**]
5. **neighbor** *ip-address* **send-label explicit-null**
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode and configures the device to run a Border Gateway Protocol (BGP) process.
Step 4	**address-family ipv4** [**unicast**] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for the IPv4 address family from which you can configure routing sessions that use standard IPv4 address prefixes.
Step 5	**neighbor** *ip-address* **send-label explicit-null** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 send-label explicit-null`	Advertises the capability of a device to send Multiprotocol Label Switching (MPLS) labels with BGP routes. • The explicit-null keyword allows a carrier supporting carrier (CSC) customer edge (CE) device to send labels with a value of 0 to its neighbor.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.99.70 activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local device.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Returns to privileged EXEC mode.

Verifying the Explicit Null Configuration

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** [*ip-address* [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** [*regexp*] | **received prefix-filter** | **received-routes** | **routes**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp neighbors** [*ip-address* [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** [*regexp*] | **received prefix-filter** | **received-routes** | **routes**]] **Example:** `Device# show ip bgp neighbors`	Displays information about the TCP and Border Gateway Protocol (BGP) connections to neighbors including explicit null. • The optional *ip-address* argument displays the IP address of the neighbor whose routes you have learned If you omit this argument, all neighbors are displayed. • The optional **advertised-routes** keyword displays all the routes the device has advertised to the neighbor. • The optional **dampened-routes** keyword displays the dampened routes to the neighbor at the IP address specified. • The optional **flap-statistics** keyword displays the flap statistics of the routes learned from the specified neighbor (external BGP [eBGP] peers only). • The optional **path** *regexp* keyword and argument displays autonomous system paths learned from the specified neighbor. An optional regular expression can be used to filter the output. • The optional **received prefix-filter** keyword displays the configured prefix list filter for the specified IP address. • The optional **received-routes** keyword displays all received routes (both accepted and rejected) from the specified neighbor. • The optional **routes** keyword displays all routes that are received and accepted. This is a subset of the output from the **received-routes** keyword.

What to do next

•

Configuration Examples for MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

Example: Configuring CSC-CE with BGP

In the following example, the carrier supporting carrier (CSC) is configured with the Border Gateway Protocol (BGP) to distribute labels and to advertise explicit null for all its connected routes:

```
neighbor 10.0.0.0 send-label explicit-null
router bgp 100
 bgp log-neighbor-changes
 neighbor 10.0.0.0 remote-as 200
 !
address-family ipv4
 neighbor 10.0.0.0 activate
 neighbor 10.0.0.0 send-label explicit-null
 no auto-summary
 no synchronization
 exit-address-family
```

Example: Verifying the Explicit Null Configuration

In this example, the **show ip bgp neighbors** command displays information about connected Border Gateway Protocol (BGP) neighbors, including IP addresses, version numbers, neighbor capabilities, message statistics, and address family statistics that show if explicit null is configured:

```
Device# show ip bgp neighbors

BGP neighbor is 10.0.0.2,  remote AS 300, external link
  BGP version 4, remote router ID 10.0.0.20
  BGP state = Established, up for 00:45:16
  Last read 00:00:16, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    ipv4 MPLS Label capability: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            47         47
    Route Refresh:          0          0
    Total:                 49         50
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 9, neighbor version 9/0
  Output queue sizes : 0 self, 0 replicated
  Index 1, Offset 0, Mask 0x2
  Member of update-group 1
  My AS number is allowed for 3 number of times
```

```
AF-dependant capabilities:
  Outbound Route Filter (ORF) type (128) Prefix-list:
Sending Prefix & Label(advertise explicit-null set)       !Explicit null is configured
                                Sent       Rcvd
Prefix activity:                ----       ----
  Prefixes Current:                3          3 (Consumes 144 bytes)
  Prefixes Total:                  3          6
  Implicit Withdraw:               0          3
  Explicit Withdraw:               0          0
  .........
  .........
```

Additional References for MPLS VPN Explicit Null Label with BGP IPv4 Label Session

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
BGP configuration tasks	*IP Routing: BGP Configuration Guide*
BGP commands	Cisco IOS IP Routing: BGP Command Reference

RFCs

RFC	Title
RFC 1163	*A Border Gateway Protocol*
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 2283	*Multiprotocol Extensions for BGP-4*
RFC 2547	*BGP/MPLS VPNs*
RFC 3107	*Carrying Label Information in BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 158: Feature Information for MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session

Feature Name	Releases	Feature Information
MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session	12.0(27)S 12.0(27)S1 12.2(27)SBA 12.2(33)SRA 12.2(33)SXH Cisco IOS XE Release 2.3	The MPLS VPN Explicit Null Label Support with BGP IPv4 Label Session feature provides a method to advertise explicit null in a BGP label session for a carrier supporting carrier (CSC) customer edge (CE) device. In 12.0(27)S, this feature was introduced. In 12.0(27)S1, support was added for the Cisco 12000 series Internet routers. In 12.2(27)SBA, support was added for the Cisco 10000 series router. This feature was integrated into Cisco IOS Release 12.2(33)SRA. This feature was integrated into Cisco IOS Release 12.2(33)SXH. In Cisco IOS XE Release 2.3, support was added for the Cisco ASR 1000 Series Routers. The following commands were introduced or modified: **debug ip bgp**, **neighbor send-label explicit-null**, **show ip bgp neighbors**, **show ip bgp vpnv4**, **show mpls forwarding-table**

Glossary

BGP

Border Gateway Protocol. The exterior Border Gateway Protocol used to exchange routing information between devices in separate autonomous systems. BGP uses TCP. Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

CE device

customer edge device. A device on the border between a VPN provider and a VPN customer that belongs to the customer.

eBGP

external Border Gateway Protocol. A BGP session between devices in different autonomous systems. When a pair of devices in different autonomous systems are more than one IP hop away from each other, an external BGP session between those two devices is called multihop external BGP.

label

A short, fixed-length data identifier that tells switching nodes how to forward data (packets or cells).

label distribution

The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

LDP

Label Distribution Protocol. The protocol that supports MPLS hop-by-hop forwarding by distributing bindings between labels and network prefixes. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LSP

label switched path. A configured connection between two devices, in which MPLS is used to carry packets. A path created by the concatenation of one or more label switched hops, allowing a packet to be forwarded by swapping labels from an MPLS node to another MPLS node.

MPLS

Multiprotocol Label Switching. A method for directing packets primarily through Layer 2 switching rather than Layer 3 routing. In MPLS, packets are assigned short, fixed-length labels at the ingress to an MPLS cloud by using the concept of forwarding equivalence classes. Within the MPLS domain, the labels are used to make forwarding decisions mostly without recourse to the original packet headers; formerly known as tag switching.

NLRI

Network Layer Reachability Information. BGP sends routing update messages containing NLRI, which describes the route. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes. The route attributes include a BGP next hop gateway address, community values, and other information.

PE device

provider edge device. A device on the border between a VPN provider and a VPN customer that belongs to the provider.

QoS

quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

router

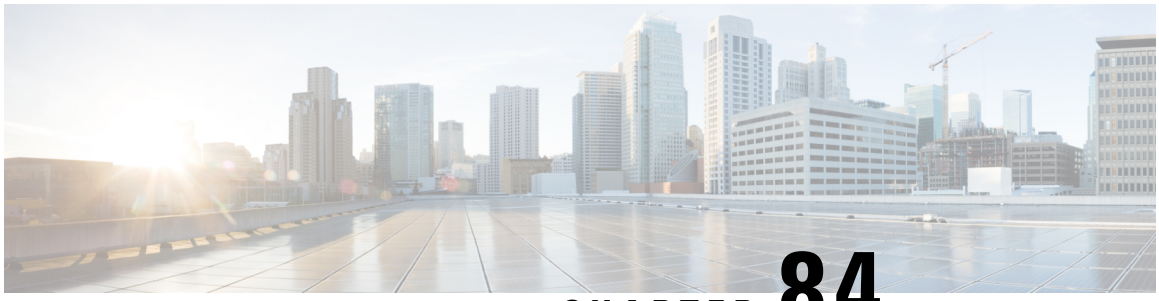
A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

VPN

Virtual Private Network. A secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.

PART VII

MPLS Traffic Engineering



CHAPTER 84

MPLS Traffic Engineering - DiffServ Aware (DS-TE)

The Multiprotocol Label Switching Traffic Engineering (MPLS TE) - DiffServ-Aware Traffic Engineering (DS-TE) feature enables service providers to perform separate admission control and separate route computation for discrete subsets of traffic (for example, voice and data traffic).

When DS-TE is combined with other Cisco software features such as QoS, the service provider can:

- Develop QoS services for end customers based on *signaled* rather than *provisioned* QoS
- Build the higher-revenue generating "strict-commitment" QoS services, without over-provisioning
- Offer virtual IP leased-line, Layer 2 service emulation, and point-to-point guaranteed bandwidth services including voice-trunking
- Enjoy the scalability properties offered by MPLS.

Information About MPLS Traffic Engineering - DiffServ Aware (DS-TE)

MPLS TE and Constraint-Based Routing (CBR)

MPLS TE allows constraint-based routing (CBR) of IP traffic. One of the constraints satisfied by CBR is the availability of required bandwidth over a selected path. DiffServ-aware TE extends MPLS traffic engineering to enable you to perform constraint-based routing of "guaranteed" traffic, which satisfies a more restrictive bandwidth constraint than that satisfied by CBR for regular traffic. The more restrictive bandwidth is termed a *sub-pool*, while the regular TE tunnel bandwidth is called the *global pool*. (The sub-pool is a portion of the

global pool. In the new IETF-Standard, the global pool is called BC0 and the sub-pool is called BC1. These are two of an eventually available eight Class Types). This ability to satisfy a more restrictive bandwidth constraint translates into an ability to achieve higher Quality of Service (QoS) performance in terms of delay, jitter, or loss for the guaranteed bandwidth services end-to-end across the network.

DS-TE has been augmented to conform to IETF standards that were developed after the initial creation of Cisco DS-TE. Now both the traditional and the IETF versions of DS-TE can be run on your network; the new releases are backwards compatible.

For example, DS-TE can be used to ensure that traffic is routed over the network so that, on every link, there is never more than 40 per cent (or any assigned percentage) of the link capacity of guaranteed traffic (for example, voice), while there can be up to 100 per cent of the link capacity of regular traffic. Assuming that QoS mechanisms are also used on every link to queue guaranteed traffic separately from regular traffic, it then becomes possible to enforce separate “overbooking” ratios for guaranteed and regular traffic. In fact, for the guaranteed traffic it becomes possible to enforce no overbooking at all--or even an underbooking--so that very high QoS can be achieved end-to-end for that traffic, even while for the regular traffic a significant overbooking continues to be enforced.

Also, through the ability to enforce a maximum percentage of guaranteed traffic on any link, the network administrator can directly control the end-to-end QoS performance parameters without having to rely on over-engineering or on expected shortest path routing behavior. This is essential for transport of applications that have very high QoS requirements such as real-time voice, virtual IP leased line, and bandwidth trading, where over-engineering cannot be assumed everywhere in the network.

The new IETF-Standard functionality of DS-TE expands the means for allocating constrained bandwidth into two distinct models, called the “Russian Dolls Model” and the “Maximum Allocation Model”. They differ from each other as follows:

Table 159: Bandwidth Constraint Model Capabilities

MODEL	Achieves Bandwidth Efficiency	Ensures Isolation across Class Types		Protects against QoS Degradation...	
		When Preemption is Not Used	When Preemption is Used	...of the Premium Class Type	...of all other Class Types
Maximum Allocation	Yes	Yes	Yes	Yes	**No**
Russian Dolls	Yes	**No**	Yes	Yes	Yes

Therefore in practice, a Network Administrator might prefer to use:

- the Maximum Allocation Model when s/he needs to ensure isolation across all Class Types without having to use pre-emption, and s/he can afford to risk some QoS degradation of Class Types other than the Premium Class.
- the Russian Dolls Model when s/he needs to prevent QoS degradation of all Class Types and can impose pre-emption.

DS-TE involves extending OSPF (Open Shortest Path First routing protocol), so that the available sub-pool or class-type bandwidth at each preemption level is advertised in addition to the available global pool bandwidth at each preemption level. And DS-TE modifies constraint-based routing to take this more complex advertised information into account during path computation.

With the addition of IETF-Standard functionality (beginning with Cisco IOS Release 12.2(33)SRB), networks may accomplish DS-TE in three different combinations or "modes", so that they may transition to the IETF-Standard formats in a manner that will not degrade their ongoing traffic service. These three situations or modes are summarized as follows:

1. The original, or "Traditional" (pre-IETF-Standard) mode. This describes networks that already operate the form of DS-TE that was introduced by Cisco a few years ago. Such networks can continue to operate is this traditional mode, even when they use the new Release 12.2(33)SRB and subsequent releases.
2. The "Migration" or combination mode. Networks already running traditional DS-TE that would like to upgrade to the IETF-Standard should first configure their routers into the Migration mode. This will allow them to continue to operate DS-TE without tunnels being torn down. In Migration mode, routers will continue to generate IGP and tunnel signalling as in the Traditional form, but now these routers will add TE-class mapping and will accept advertisement in both the Traditional and the new IETF-Standard formats.
3. The "Liberal IETF" mode. Networks already running in the Migration mode can then move into IETF formats by reconfiguring their routers into this flexible (hence "Liberal") combination: their routers will henceforth generate IGP advertisement and tunnel signalling according to the new IETF Standard, but they will remain capable of accepting advertisement in the Traditional format, as well as in the new IETF format.

The table below summarizes these distinctions among the three modes.

Table 160: Summary of DS-TE Mode behaviors

	Uses TE-class mapping	**Generates**		**Processes**	
MODE		IGP Advertisement	RSVP-TE Signaling	IGP Advertisement	RSVP-TE Signaling
Traditional	No	traditional	traditional	traditional1	traditional
Migration	Yes	traditional	traditional	traditional & IETF	traditional & IETF
Liberal IETF	Yes	IETF	traditional & IETF	traditional & IETF	traditional & IETF

1Note that it is not possible for the Traditional mode to be liberal in what it accepts in terms of IGP, since it does not use TE-Class mapping and therefore cannot interpret the "Unreserved Bandwidth" in the IETF-compliant way when the Subpool Sub-TLV is absent.

From Traditional to IETF-Standard Commands

DS-TE commands originally were developed from the then-existing command set that had been used to configure MPLS traffic engineering. The only difference introduced at that time to create DS-TE was the expansion of two commands:

- **ip rsvp bandwidth** was expanded to configure the size of the sub-pool on every link.
- **tunnel mpls traffic-eng bandwidth** was expanded to enable a TE tunnel to reserve bandwidth from the sub-pool.

The ip rsvp bandwidth command

The early MPLS command had been

```
ip rsvp bandwidth x y
```

where x = the size of the only possible pool, and y = the size of a single traffic flow (ignored by traffic engineering).

Then, to create the original implementation of DS-TE, the command was made into

```
ip rsvp bandwidth x y sub-pool z
```

where x = the size of the global pool, and z = the size of the sub-pool.

With the addition of the IETF-Standard version of DS-TE, the command has been further extended to become:

```
ip rsvp bandwidth x y [ [rdm x {subpool z | bc1 z}] | [mam bc0 x bc1 z]]
```

where x = the size of the global pool (now called **bc0**), and z = the size of the sub-pool (now called also **bc1**).

Two bandwidth constraint models also have become available, “Russian Dolls” (indicated by the keyword **rdm**) and “Maximum Allocation” (**mam**). The former model allows greater sharing of bandwidth across all Class Types (bandwidth pools), while the latter protects especially the premium Class Type. (The IETF Standard makes possible the future implementation of as many as seven sub-pools within one LSP, instead of just one sub-pool per LSP).

The tunnel mpls traffic-eng bandwidth command

The pre-DS-TE traffic engineering command was

```
tunnel mpls traffic-eng bandwidth b
```

where b = the amount of bandwidth this tunnel requires.

So for the original DS-TE, you specified from which pool (global or sub) the tunnel's bandwidth would come. You could enter

```
tunnel mpls traffic-eng bandwidth sub-pool b
```

to indicate that the tunnel should use bandwidth from the sub-pool. Alternatively, you could enter

```
tunnel mpls traffic-eng bandwidth b
```

to indicate that the tunnel should use bandwidth from the global pool (which was the default).

With the addition of the IETF-Standard version of DS-TE, the command has been extended to become:

```
tunnel mpls traffic-eng bandwidth [sub-pool|class-type 1] b
```

where both **sub-pool** and **class-type 1** indicate the same, smaller bandwidth pool (now called class-type 1). The two keywords can be used interchangeably.

The mpls traffic-eng ds-te commands

The IETF Standard introduces two new commands, one to indicate the Bandwidth Constraints model

```
mpls traffic-eng ds-te bc-model [rdm | mam]
```

and one to select the DS-TE mode:

```
mpls traffic-eng ds-te mode [migration|ietf]
```

(The concepts of bc-model and DS-TE mode were explained in the section above).

The first command allows you to select between the Russian Dolls Model (**rdm**) and the Maximum Allocation Model (**mam**) of bandwidth constraints.

The second command allows you to transition a network from traditional DS-TE tunnels to the IETF Standard without disrupting any of the tunnels' operation. To accomplish this, you first put the routers into Migration mode (using the **migration** keyword) and subsequently into the Liberal-IETF mode (using the **ietf** keyword).

Transitioning a Network to the IETF Standard

Networks already operating DS-TE tunnels by means of the traditional, pre-IETF-Standard software can switch to the IETF-Standard without interrupting their DS-TE service by following this sequence:

1. Install Cisco IOS Release 12.2(33)SRB (or a subsequent release) on each router in the network, gradually, one router at a time, using Cisco's In Service Software Upgrade (ISSU) procedure which protects ongoing network traffic from interruption. (After that installation, DS-TE tunnels in the network will continue to operate by using the pre-IETF-Standard formats.)
2. Enter the global configuration command **mpls traffic-eng ds-te mode migration** on each router in the network, one router at a time. This will enable the routers to receive IETF-format IGP advertisement and RSVP-TE signaling, while the routers will continue to generate and receive the pre-Standard formats for those two functions.
3. After all the routers in the network have begun to operate in Migration mode, enter the global configuration command **mpls traffic-eng ds-te mode ietf** on each router, one at a time. This will cause the router to refresh its TE tunnels with IETF-compliant Path signaling, without disrupting the tunnels' operation. This mode also causes the router to generate IGP advertisement in the IETF-Standard format.

Guaranteed Bandwidth Service Configuration

Once two bandwidth pools are configured traffic can be managed in the following ways:

- Use one pool, the sub-pool, for tunnels that carry traffic requiring strict bandwidth guarantees or delay guarantees
- Use the other pool, the global pool, for tunnels that carry traffic requiring only Differentiated Service.

Having a separate pool for traffic requiring strict guarantees allows you to limit the amount of such traffic admitted on any given link. Often, it is possible to achieve strict QoS guarantees only if the amount of guaranteed traffic is limited to a portion of the total link bandwidth.

Having a separate pool for other traffic (best-effort or diffserv traffic) allows you to have a separate limit for the amount of such traffic admitted on any given link. This is useful because it allows you to fill up links with best-effort/diffserv traffic, thereby achieving a greater utilization of those links.

Providing Strict QoS Guarantees Using DS-TE Sub-pool Tunnels

A tunnel using sub-pool bandwidth can satisfy the stricter requirements if you do all of the following:

- Select a queue--or in diffserv terminology, select a PHB (per-hop behavior)--to be used exclusively by the strict guarantee traffic. This shall be called the “GB queue.”

  If delay/jitter guarantees are sought, the diffserv Expedited Forwarding queue (EF PHB) is used. (On the Cisco 7500 [VIP], it is the "priority" queue.) You must configure the bandwidth of the queue to be at least equal to the bandwidth of the sub-pool.

  If only bandwidth guarantees are sought, the diffserv Assured Forwarding PHB (AF PHB) is used. (On the Cisco 7500 [VIP], you use one of the existing Class-Based Weighted Fair Queuing [CBWFQ] queues.)

- Ensure that the guaranteed traffic sent through the sub-pool tunnel is placed in the GB queue *at the outbound interface of every tunnel hop* , and that no other traffic is placed in this queue. This is done by marking the traffic that enters the tunnel with a unique value in the *mpls exp bits* field, and steering only traffic with that marking into the GB queue.

- Ensure that this GB queue is never oversubscribed; that is, see that no more traffic is sent into the sub-pool tunnel than the GB queue can handle.

  This done by rate-limiting the guaranteed traffic before it enters the sub-pool tunnel. The aggregate rate of all traffic entering the sub-pool tunnel should be less than or equal to the bandwidth capacity of the sub-pool tunnel. Excess traffic can be dropped (in the case of delay/jitter guarantees) or can be marked differently for preferential discard (in the case of bandwidth guarantees).

- Ensure that the amount of traffic entering the GB queue is limited to an appropriate percentage of the total bandwidth of the corresponding outbound link. The exact percentage to use depends on several factors that can contribute to accumulated delay in your network: your QoS performance objective, the total number of tunnel hops, the amount of link fan-in along the tunnel path, burstiness of the input traffic, and so on.

  This is done by setting the sub-pool bandwidth of each outbound link to the appropriate percentage of the total link bandwidth (that is, by adjusting the z parameter of the **ip rsvp bandwidth** command).

Providing Differentiated Service Using DS-TE Global Pool Tunnels

You can configure a tunnel using global pool bandwidth to carry best-effort as well as several other classes of traffic. Traffic from each class can receive differentiated service if you do all of the following:

1. Select a separate queue (a distinct diffserv PHB) for each traffic class. For example, if there are three classes (gold, silver, and bronze) there must be three queues (diffserv AF2, AF3, and AF4).
2. Mark each class of traffic using a unique value in the MPLS experimental bits field (for example gold = 4, silver = 5, bronze = 6).
3. Ensure that packets marked as Gold are placed in the gold queue, Silver in the silver queue, and so on. The tunnel bandwidth is set based on the expected aggregate traffic across all classes of service.

To control the amount of diffserv tunnel traffic you intend to support on a given link, adjust the size of the global pool on that link.

Providing Strict Guarantees and Differentiated Service in the Same Network

Because DS-TE allows simultaneous constraint-based routing of sub-pool and global pool tunnels, strict guarantees and diffserv can be supported simultaneously in a given network.

Prerequisites for MPLS Traffic Engineering - DiffServ Aware (DS-TE)

Your network must support the following Cisco software features in order to support guaranteed bandwidth services based on DiffServ-aware Traffic Engineering:

- MPLS
- IP Cisco Express Forwarding (CEF)
- OSPF or ISIS
- RSVP-TE
- QoS

How to Configure MPLS Traffic Engineering - DiffServ Aware (DS-TE)

Configuring DS-TE Tunnels

To establish a sub-pool (BC1) traffic engineering tunnel, you must enter configurations at three levels:

- the device level (router or switch router)
- the physical interface
- the tunnel interface

On the first two levels, you activate traffic engineering; on the third level--the tunnel interface--you establish the sub-pool tunnel. Therefore, it is only at the tunnel headend device that you need to configure all three levels. At the tunnel midpoints and tail, it is sufficient to configure the first two levels.

In the tables below, each command is explained in brief. For a more complete explanation of any command, type it into the Command Lookup Tool at http://www.cisco.com/cgi-bin/Support/Cmdlookup/home.pl . (If prompted to log in there, use your Cisco.com account username and password).

Level 1 Configuring the Device

At this level, you tell the device (router or switch router) to use accelerated packet-forwarding (known as Cisco Express Forwarding or CEF), MultiProtocol Label Switching (MPLS), traffic-engineering tunneling, a bandwidth constraints model, and either the OSPF or IS-IS routing algorithm (Open Shortest Path First or Intermediate System to Intermediate System). This level is called the global configuration mode, because the configuration is applied globally, to the entire device, rather than to a specific interface or routing instance.

You enter the following commands:

SUMMARY STEPS

1. Router(config)# **ip cef distributed**
2. Router(config)# **mpls traffic-eng tunnels**
3. Router(config)# **mpls traffic-eng ds-te bc-model** [**rdm** | **mam**]
4. Choose one of the following:
   - Router(config)# **router ospf**
   - Router(config)# **router isis**
5. Router (config-router)# **net** *network-entity-title*
6. Router (config-router)# **metric-style wide**
7. Router (config-router)# **is-type level** *n*
8. Router (config-router)# **mpls traffic-eng level***l*
9. Router (config-router)# **passive-interface loopback0**
10. Router(config-router)# **mpls traffic-eng router-id loopback0**
11. Router(config-router)# **mpls traffic-eng area** *num*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **ip cef distributed**	Enables CEF--which accelerates the flow of packets through the device.
Step 2	Router(config)# **mpls traffic-eng tunnels**	Enables MPLS, and specifically its traffic engineering tunnel capability.
Step 3	Router(config)# **mpls traffic-eng ds-te bc-model** [**rdm** | **mam**]	Specifies the bandwidth constraints model (see the Feature Overview section).
Step 4	Choose one of the following: • Router(config)# **router ospf** • Router(config)# **router isis**	Invokes the OSPF routing process for IP and puts the device into router configuration mode. Proceed now to Steps 10 and 11. Alternatively, you may invoke the IS-IS routing process with this command, and continue with Step 5.
Step 5	Router (config-router)# **net** *network-entity-title*	Specifies the IS-IS network entity title (NET) for the routing process.
Step 6	Router (config-router)# **metric-style wide**	Enables the router to generate and accept IS-IS new-style TLVs (type, length, and value objects).
Step 7	Router (config-router)# **is-type level** *n*	Configures the router to learn about destinations inside its own area or “IS-IS level”.
Step 8	Router (config-router)# **mpls traffic-eng level***l*	Specifies the IS-IS level (which must be same level as in the preceding step) to which the router will flood MPLS traffic- engineering link information.
Step 9	Router (config-router)# **passive-interface loopback0**	Instructs IS-IS to advertise the IP address of the loopback interface without actually running IS-IS on that interface.

	Command or Action	Purpose
		Continue with Step 10 but don't do Step 11--because Step 11 refers to OSPF.
Step 10	Router(config-router)# **mpls traffic-eng router-id loopback0**	Specifies that the traffic engineering router identifier is the IP address associated with the *loopback0* interface.
Step 11	Router(config-router)# **mpls traffic-eng area** *num*	Turns on MPLS traffic engineering for a particular OSPF area.

Level 2 Configuring the Physical Interface

Having configured the device, you now must configure the interface on that device through which the tunnel will run. To do that, you first put the router into interface-configuration mode.

You then enable Resource Reservation Protocol (RSVP). This protocol is used to signal (set up) a traffic engineering tunnel, and to tell devices along the tunnel path to reserve a specific amount of bandwidth for the traffic that will flow through that tunnel. It is with this command that you establish the maximum size of the sub-pool (BC1).

Finally, you enable the MPLS traffic engineering tunnel feature on this physical interface--and if you will be relying on the IS-IS routing protocol, you enable that as well .

To accomplish these tasks, you enter the following commands:

SUMMARY STEPS

1. Router(config)# **interface** *interface-id*
2. Router(config-if)# **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*][[**rdm** *kbps*{[**subpool** *kbps*]|[**bc1** *subpool*]}]|[**mam max-reservable-bw** *kbps* **bc0** *kbps* **bc1** *kbps*]]
3. Router(config-if)# **mpls traffic-eng tunnels**
4. Router(config-if)# **ip router isis**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **interface** *interface-id*	Moves configuration to the interface level, directing subsequent configuration commands to the specific interface identified by the *interface-id*.
Step 2	Router(config-if)# **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*][[**rdm** *kbps*{[**subpool** *kbps*]|[**bc1** *subpool*]}]|[**mam max-reservable-bw** *kbps* **bc0** *kbps* **bc1** *kbps*]] **Example:**	Enables RSVP on this interface, indicates the Bandwidth Constraints Model to be used (explained in the Feature Overview section), and limits the amount of bandwidth RSVP can reserve on this interface. The sum of bandwidth used by all tunnels on this interface cannot exceed *interface-kbps.*
Step 3	Router(config-if)# **mpls traffic-eng tunnels**	Enables the MPLS traffic engineering tunnel feature on this interface.
Step 4	Router(config-if)# **ip router isis**	Enables the IS-IS routing protocol on this interface. Do not enter this command if you are configuring for OSPF.

Level 3 Configuring the Tunnel Interface

Now you create a set of attributes for the tunnel itself; those attributes are configured on the "tunnel interface" (not to be confused with the physical interface just configured above).

You enter the following commands:

SUMMARY STEPS

1. Router(config)# **interface tunnel1**
2. Router(config-if)# **tunnel destination** *A.B.C.D*
3. Router(config-if)# **tunnel mode mpls traffic-eng**
4. Router(config-if)# **tunnel mpls traffic-eng bandwidth** {**sub-pool** | **class-type1**} *bandwidth*
5. Router(config-if)# **tunnel mpls traffic-eng priority**
6. Router(config-if)# **tunnel mpls traffic-eng path-option**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **interface tunnel1**	Creates a tunnel interface (named in this example **tunnel1**) and enters interface configuration mode.
Step 2	Router(config-if)# **tunnel destination** *A.B.C.D*	Specifies the IP address of the tunnel tail device.
Step 3	Router(config-if)# **tunnel mode mpls traffic-eng**	Sets the tunnel's encapsulation mode to MPLS traffic engineering.
Step 4	Router(config-if)# **tunnel mpls traffic-eng bandwidth** {**sub-pool** | **class-type1**} *bandwidth*	Configures the tunnel's bandwidth, and assigns it either to the sub-pool (when you use that keyword or the IETF-Standard keyword **class-type1**) or to the global pool (when you leave out both keywords).
Step 5	Router(config-if)# **tunnel mpls traffic-eng priority**	Sets the priority to be used when the system determines which existing tunnels are eligible to be preempted.
Step 6	Router(config-if)# **tunnel mpls traffic-eng path-option**	Configures the paths (hops) a tunnel should use. The user can enter an explicit path (can specify the IP addresses of the hops) or can specify a dynamic path (the router figures out the best set of hops).

Verifying the Configuration

To view the complete configuration you have entered, use the EXEC command **show running-config** and check its output display for correctness.

To check just one tunnel's configuration, enter **show interfaces tunnel** followed by the tunnel interface number. And to see that tunnel's RSVP bandwidth and flow, enter **show ip rsvp interface** followed by the name or number of the physical interface.

Here is an example of the information displayed by these latter two commands. (To see an explanation of each field used in the following displays, enter **show interfaces tunnel** or **show ip rsvp interface** into the

Command Lookup Tool at http://www.cisco.com/cgi-bin/Support/Cmdlookup/home.pl . If prompted to log in there, use your Cisco.com account username and password.)

```
Router# show interfaces tunnel 4
Tunnel4 is up, line protocol is down
  Hardware is Routing Tunnel
  MTU 1500 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set, keepalive set (10 sec)
  Tunnel source 0.0.0.0, destination 0.0.0.0
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets, 0 restarts
Router# show ip rsvp interface pos4/0
interface    allocated  i/f max  flow max sub max
PO4/0        300K       466500K  466500K  0M
```

To view all tunnels at once on the router you have configured, enter **show mpls traffic-eng tunnels brief** . The information displayed when tunnels are functioning properly looks like this:

```
Router# show mpls traffic-eng tunnels brief
Signalling Summary:
LSP Tunnels Process:           running
RSVP Process:                  running
Forwarding:                    enabled
Periodic reoptimization:       every 3600 seconds, next in 3029 seconds
TUNNEL NAME  DESTINATION      UP IF     DOWN IF   STATE/PROT
GSR1_t0  192.168.1.13     -          SR3/0     up/up
GSR1_t1  192.168.1.13     -          SR3/0     up/up
GSR1_t2  192.168.1.13     -          PO4/0     up/up
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

When one or more tunnels is not functioning properly, the display could instead look like this. (In the following example, tunnels t0 and t1 are down, as indicated in the far right column).

```
Router# show mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:           running
    RSVP Process:                  running
    Forwarding:                    enabled
    Periodic reoptimization:       every 3600 seconds, next in 2279 seconds
TUNNEL NAME  DESTINATION      UP IF     DOWN IF   STATE/PROT
GSR1_t0  192.168.1.13     -          SR3/0     up/down
GSR1_t1  192.168.1.13     -          SR3/0     up/down
GSR1_t2  192.168.1.13     -          PO4/0     up/up
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

To find out why a tunnel is down, insert its name into this same command, after adding the keyword **name** and omitting the keyword **brief**. For example:

```
Router# show mpls traffic-eng tunnels name GSR1_t0

Name:GSR1_t0                           (Tunnel0) Destination:192.168.1.13
```

```
Status:
  Admin:up         Oper:down  Path: not valid       Signalling:connected
```

If, as in this example, the Path is displayed as not valid, use the **show mpls traffic-eng topology** command to make sure the router has received the needed updates.

Additionally, you can use any of the following **show** commands to inspect particular aspects of the network, router, or interface concerned:

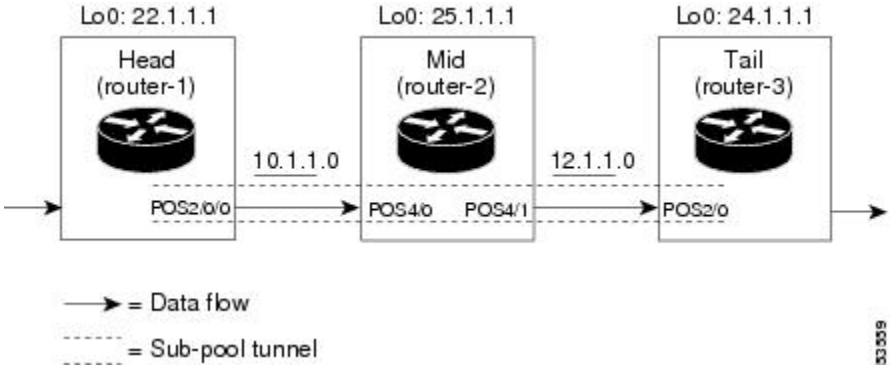
To see information about...		Use this command
this level	and this item...	
Network	Advertised bandwidth allocation information	**show mpls traffic-eng link-management advertisements**
	Preemptions along the tunnel path	**debug mpls traffic-eng link-management preemption**
	Available TE link band- width on all head routers	**show mpls traffic-eng topology**
Router	Status of all tunnels cur- rently signalled by this router	**show mpls traffic-eng link-management admission-control**
	Tunnels configured on midpoint routers	**show mpls traffic-eng link-management summary**
Physical interface	Detailed information on current bandwidth pools	**show mpls traffic-eng link-management bandwidth-allocation** [*interface-name*]
	TE RSVP bookkeeping	**show mpls traffic-eng link-management interfaces**
	Entire configuration of one interface	**show run interface**

MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples

Note The following examples illustrate DS-TE in the traditional, pre-IETF-Standard mode. You may update these examples simply by inserting the new Device Level command **mpls traffic-eng ds-te bc-model** , and by applying the updated syntax within the two modified commands as each is shown respectively at the Physical Interface Level (**ip rsvp bandwidth**), and at the Tunnel Interface Level (**tunnel mpls traffic-eng bandwidth**).

First this section presents the DS-TE configurations needed to create the sub-pool tunnel. Then it presents the more comprehensive design for building end-to-end guaranteed bandwidth service, which involves configuring Quality of Service as well.

As shown in the figure below, the tunnel configuration involves at least three devices--tunnel head, midpoint, and tail. On each of those devices one or two network interfaces must be configured, for traffic ingress and egress.



Tunnel Head: Example

At the device level:

```
router-1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
router-1(config)# ip cef distributed
router-1(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-1(config)# router isis`	`router ospf 100`
`router-1(config-router)# net 49.0000.1000.0000.0010.00`	`redistribute connected`
`router-1(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-1(config-router)# is-type level-1`	`network 22.1.1.1 0.0.0.0 area 0`
`router-1(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`
`router-1(config-router)# passive-interface Loopback0`	

[now one resumes the common command set]:

```
router-1(config-router)# mpls traffic-eng router-id Loopback0
router-1(config-router)# exit
router-1(config)# interface Loopback0
```

At the virtual interface level:

```
router-1(config-if)# ip address 22.1.1.1 255.255.255.255
router-1(config-if)# no ip directed-broadcast
router-1(config-if)# exit
```

At the device level:

```
router-1(config)# interface POS2/0/0
```

At the physical interface level (egress):

```
router-1(config-if)# ip address 10.1.1.1 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
[and if using IS-IS instead of OSPF]:
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

At the device level:

```
router-1(config)# interface Tunnel1
```

At the tunnel interface level:

```
router-1(config-if)# bandwidth 110000
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 24.1.1.1
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
router-1(config-if)# exit
router-1(config)#
```

Midpoint Devices: Example

At the device level:

```
router-2# configure terminal
router-2(config)# ip cef distributed
router-2(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-2(config)# router isis`	`router ospf 100`
`router-2(config-router)# net 49.0000.1000.0000.0012.00`	`redistribute connected`
`router-2(config-router)# metric-style wide`	`network 11.1.1.0 0.0.0.255 area 0`

router-2(config-router)# **is-type level-1**	**network 12.1.1.0 0.0.0.255 area 0**
router-2(config-router)# **mpls traffic-eng level-1**	**network 25.1.1.1 0.0.0.0 area 0**
router-2(config-router)# **passive-interface Loopback0**	**mpls traffic-eng area 0**

[now one resumes the common command set]:

```
router-2(config-router)# mpls traffic-eng router-id Loopback0
router-2(config-router)# exit
router-2(config)# interface Loopback0
```

At the virtual interface level:

```
router-2(config-if)# ip address 25.1.1.1 255.255.255.255
router-2(config-if)# no ip directed-broadcast
router-2(config-if)# exit
```

```
At the device level:
router-1(config)# interface POS4/0
router-1(config-if)# ip address 11.1.1.2 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
```

[If using IS-IS instead of OSPF]:

```
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
At the device level:
router-1(config)# interface POS4/1
router-1(config-if)# ip address 12.1.1.2 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
```

[If using IS-IS instead of OSPF]:

```
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

Note that there is no configuring of tunnel interfaces at the mid-point devices, only network interfaces and the device globally.

Tail-End Device: Example

At the device level:

```
router-3# configure terminal
router-3(config)# ip cef distributed
router-3(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

router-3(config)# **router isis**	**router ospf 100**
router-3(config-router)# **net 49.0000.1000.0000.0013.00**	**redistribute connected**
router-3(config-router)# **metric-style wide**	**network 12.1.1.0 0.0.0.255 area 0**
router-3(config-router)# **is-type level-1**	**network 24.1.1.1 0.0.0.0 area 0**
router-3(config-router)# **mpls traffic-eng level-1**	**mpls traffic-eng area 0**
router-3(config-router)# **passive-interface Loopback0**	

[now one resumes the common command set]:

```
router-3(config-router)# mpls traffic-eng router-id Loopback0
router-3(config-router)# exit
router-3(config)# interface Loopback0
```

At the virtual interface level:

```
router-3(config-if)# ip address 24.1.1.1 255.255.255.255
router-3(config-if)# no ip directed-broadcast
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the device level:

```
router-1(config)# interface POS4/0
router-1(config-if)# ip address 12.1.1.3 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
```

[If using IS-IS instead of OSPF]:

```
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

Guaranteed Bandwidth Service: Examples

Given the many topologies in which Guaranteed Bandwidth Services can be applied, there is space here only to present two examples. They illustrate opposite ends of the spectrum of possibilities.

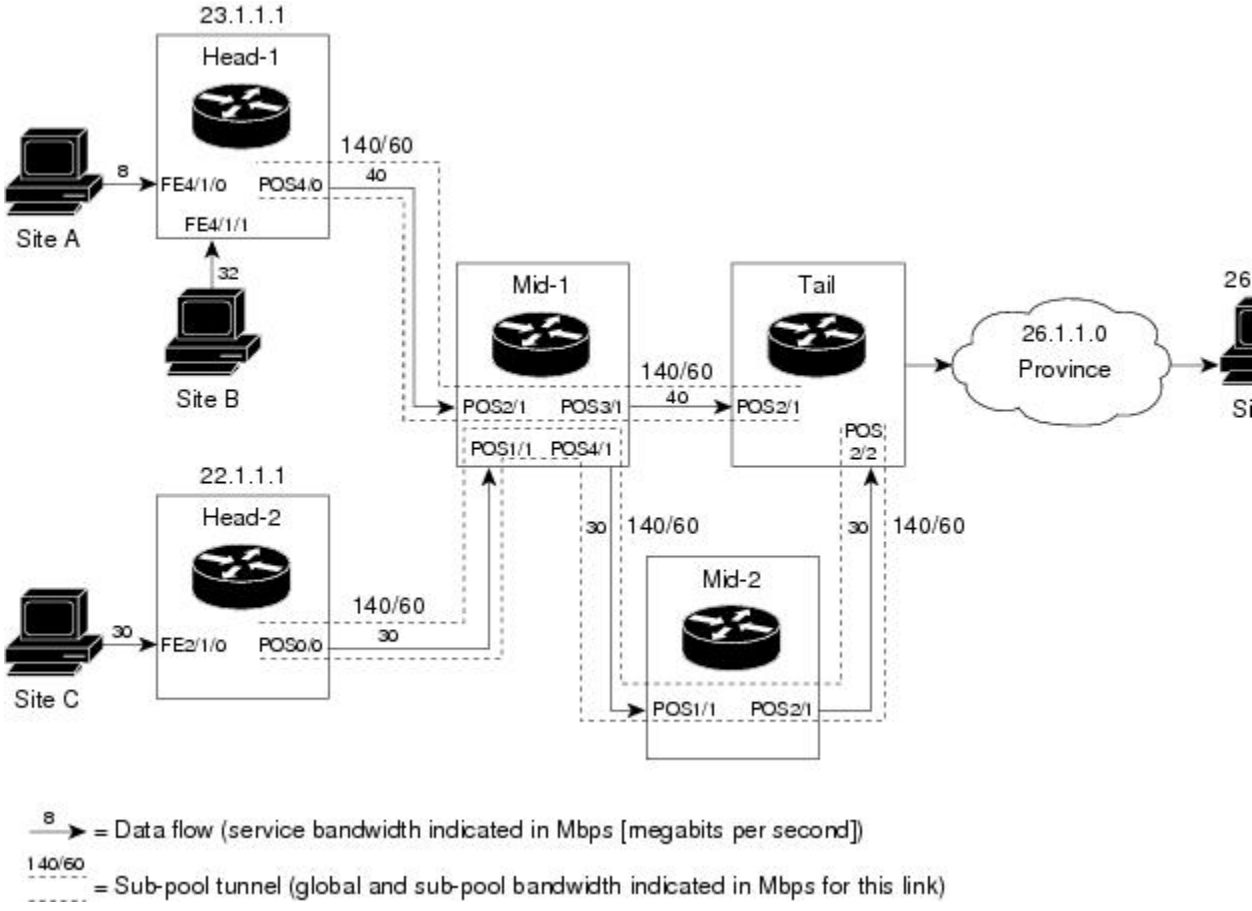
In the first example, the guaranteed bandwidth tunnel can be easily specified by its destination. So the forwarding criteria refer to a single destination prefix.

In the second example, there can be many final destinations for the guaranteed bandwidth traffic, including a dynamically changing number of destination prefixes. So the forwarding criteria are specified by Border Gateway Protocol (BGP) policies.

Single Destination Prefix: Example

The figure below illustrates atopology for guaranteed bandwidth services whose destination is specified by a single prefix, either Site D (like a voice gateway, here bearing prefix 26.1.1.1) or a subnet (like the location of a web farm, here called “Province” and bearing prefix 26.1.1.0). Three services are offered:

- From Site A (defined as all traffic arriving at interface FE4/1/0): to host 26.1.1.1, 8 Mbps of guaranteed bandwidth with low loss, low delay and low jitter
- From Site B (defined as all traffic arriving at interface FE4/1/1): towards subnet 26.1.1.0, 32 Mbps of guaranteed bandwidth with low loss
- From Site C (defined as all traffic arriving at interface FE2/1/0): 30 Mbps of guaranteed bandwidth with low loss



These three services run through two sub-pool tunnels:

- From the Head-1 router, 23.1.1.1, to the router-4 tail

- From the Head-2 router, 22.1.1.1, to the router-4 tail

Both tunnels use the same tail router, though they have different heads. (In the figure above one midpoint router is shared by both tunnels. In the real world there could of course be many more midpoints.)

All POS interfaces in this example are OC3, whose capacity is 155 Mbps.

Configuring Tunnel Head-1 Example

First we recapitulate commands that establish two bandwidth pools and a sub-pool tunnel (as presented earlier in this Configuration Examples section). Then we present the QoS commands that guarantee end-to-end service on the subpool tunnel. (With the 7500 router, Modular QoS CLI is used.)

At the device level:

```
router-1(config)# ip cef distributed
router-1(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-1(config)# router isis`	`router ospf 100`
`router-1(config-router)# net 49.0000.1000.0000.0010.00`	`redistribute connected`
`router-1(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-1(config-router)# is-type level-1`	`network 23.1.1.1 0.0.0.0 area 0`
`router-1(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`
`router-1(config-router)# passive-interface Loopback0`	

[now one resumes the common command set]:

```
router-1(config-router)# mpls traffic-eng router-id Loopback0
router-1(config-router)# exit
```

Create a virtual interface:

```
router-1(config)# interface Loopback0
router-1(config-if)# ip address 23.1.1.1 255.255.255.255
router-1(config-if)# no ip directed-broadcast
router-1(config-if)# exit
```

At the outgoing physical interface:

```
router-1(config)# interface pos4/0
router-1(config-if)# ip address 10.1.1.1 255.0.0.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-1(config-if)# ip router isis
[and in all cases}:
router-1(config-if)# exit
```

At the tunnel interface:

```
router-1(config)# interface Tunnel1
router-1(config-if)# bandwidth 110000
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 27.1.1.1
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 40000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
```

To ensure that packets destined to host 26.1.1.1 and subnet 26.1.1.0 are sent into the sub-pool tunnel, we create a static route. At the device level:

```
router-1(config)# ip route 26.1.1.0 255.255.255.0 Tunnel1
router-1(config)# exit
```

And in order to make sure that the Interior Gateway Protocol (IGP) will not send any other traffic down this tunnel, we disable autoroute announce:

```
router-1(config)# no tunnel mpls traffic-eng autoroute announce
```

At the inbound physical interface (FE4/1/0):

1. In global configuration mode, create a class of traffic matching ACL 100, called "sla-1-class":

```
class-map match-all sla-1-class
match access-group 100
```

1. Create an ACL 100 to refer to all packets destined to 26.1.1.1:

```
access-list 100 permit ip any host 26.1.1.1
```

1. Create a policy named “sla-1-input-policy”, and according to that policy:
   a. Packets in the class called “sla-1-class” are rate-limited to:

- a rate of 8 million bits per second

- a normal burst of 1 million bytes

- a maximum burst of 2 million bytes

1. Packets which conform to this rate are marked with MPLS experimental bit 5 and are forwarded.
2. Packets which exceed this rate are dropped.
3. All other packets are marked with experimental bit 0 and are forwarded.

```
policy-map sla-1-input-policy
class sla-1-class
police 8000000 1000000 2000000 conform-action set-mpls-exp-transmit 5 \ exceed-action drop
class class-default
set-mpls-exp-transmit 0
```

1. The policy is applied to packets entering interface FE4/1/0.

```
interface FastEthernet4/1/0
service-policy input sla-1-input-policy
```

At the inbound physical interface (FE4/1/1):

1. In global configuration mode, create a class of traffic matching ACL 120, called "sla-2-class":

```
class-map match-all sla-2-class
match access-group 120
```

1. Create an ACL, 120, to refer to all packets destined to subnet 26.1.1.0:

```
access-list 120 permit ip any 26.1.1.0 0.0.0.255
```

1. Create a policy named “sla-2-input-policy”, and according to that policy:
   a. Packets in the class called “sla-2-class” are rate-limited to:

- a rate of 32 million bits per second

- a normal burst of 1 million bytes

- a maximum burst of 2 million bytes

1. Packets which conform to this rate are marked with MPLS experimental bit 5 and are forwarded.
2. Packets which exceed this rate are dropped.
3. All other packets are marked with experimental bit 0 and are forwarded.

```
policy-map sla-2-input-policy
class sla-2-class
police 32000000 1000000 2000000 conform-action set-mpls-exp-transmit 5 \ exceed-action drop
class class-default
set-mpls-exp-transmit 0
```

1. The policy is applied to packets entering interface FE4/1/1.

```
interface FastEthernet4/1/1
service-policy input sla-2-input-policy
```

The outbound interface (POS4/0) is configured as follows:

1. In global configuration mode, create a class of traffic matching experimental bit 5, called "exp-5-traffic".

```
class-map match-all exp-5-traffic
match mpls experimental 5
```

1. Create a policy named “output-interface-policy”. According to that policy, packets in the class “exp-5-traffic” are put in the priority queue (which is rate-limited to 62 kbits/sec).

```
policy-map output-interface-policy
class exp-5-traffic
priority 32
```

1. The policy is applied to packets exiting interface POS4/0.

```
   interface POS4/0
service-policy output output-interface-policy
```

The result of the above configuration lines is that packets entering the Head-1 router via interface FE4/1/0 destined to host 26.1.1.1, or entering the router via interface FE4/1/1 destined to subnet 26.1.1.0, will have their MPLS experimental bit set to 5. We assume that no other packets entering the router (on any interface) are using this value. (If this cannot be assumed, an additional configuration must be added to mark all such packets to another experimental value.) Packets marked with experimental bit 5, when exiting the router via interface POS4/0, will be placed into the priority queue.

Note Packets entering the router via FE4/1/0 or FE4/1/1 and exiting POS4/0 enter as IP packets and exit as MPLS packets.

Configuring Tunnel Head-2 Example

First we recapitulate commands that establish two bandwidth pools and a sub-pool tunnel (as presented earlier in this Configuration Examples section). Then we present the QoS commands that guarantee end-to-end service on the sub-pool tunnel.

At the device level:

```
router-2(config)# ip cef distributed
router-2(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-2(config)# router isis`	`router ospf 100`
`router-2(config-router)# net 49.0000.1000.0000.0011.00`	`redistribute connected`
`router-2(config-router)# metric-style wide`	`network 11.1.1.0 0.0.0.255 area 0`
`router-2(config-router)# is-type level-1`	`network 22.1.1.1 0.0.0.0 area 0`
`router-2(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`

router-2(config-router)# **passive-interface Loopback0**	

[now one resumes the common command set]:

```
router-2(config-router)# mpls traffic-eng router-id Loopback0
router-2(config-router)# exit
```

Create a virtual interface:

```
router-2(config)# interface Loopback0
router-2(config-if)# ip address 22.1.1.1 255.255.255.255
router-2(config-if)# no ip directed broadcast
router-2(config-if)# exit
```

At the outgoing physical interface:

```
router-2(config)# interface pos0/0
router-2(config-if)# ip address 11.1.1.1 255.0.0.0
router-2(config-if)# mpls traffic-eng tunnels
router-2(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-2(config-if)# ip router isis
[and in all cases]:
router-2(config-if)# exit
```

At the tunnel interface:

```
router-2(config)# interface Tunnel2
router-2(config-if)# ip unnumbered Loopback0
router-2(config-if)# tunnel destination 27.1.1.1
router-2(config-if)# tunnel mode mpls traffic-eng
router-2(config-if)# tunnel mpls traffic-eng priority 0 0
router-2(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000
router-2(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
router-2(config-if)# exit
```

And to ensure that packets destined to subnet 26.1.1.0 are sent into the sub-pool tunnel, we create a static route, at the device level:

```
router-2(config)# ip route 26.1.1.0 255.255.255.0 Tunnel2
router-2(config)# exit
```

Finally, in order to make sure that the Interior Gateway Protocol (IGP) will not send any other traffic down this tunnel, we disable autoroute announce:

```
router-2(config)# no tunnel mpls traffic-eng autoroute announce
 At the inbound physical interface (FE2/1/0):
```

1. In global configuration mode, create a class of traffic matching ACL 130, called "sla-3-class":

```
class-map match-all sla-3-class
match access-group 130
```

1. Create an ACL, 130, to refer to all packets destined to subnet 26.1.1.0:

```
access-list 130 permit ip any 26.1.1.0 0.0.0.255
```

1. Create a policy named “sla-3-input-policy”, and according to that policy:
   a. Packets in the class called “sla-3-class” are rate-limited to:

- a rate of 30 million bits per second
- a normal burst of 1 million bytes
- a maximum burst of 2 million bytes

1. Packets which conform to this rate are marked with MPLS experimental bit 5 and are forwarded.
2. Packets which exceed this rate are dropped.
3. All other packets are marked with experimental bit 0 and are forwarded.

```
policy-map sla-3-input-policy
class sla-3-class
police 30000000 1000000 2000000 conform-action set-mpls-exp-transmit 5 \ exceed-action drop
class class-default
set-mpls-exp-transmit 0
```

1. The policy is applied to packets entering interface FE2/1/0.

```
interface FastEthernet2/1/0
service-policy input sla-3-input-policy
```

The outbound interface POS0/0 is configured as follows:

1. In global configuration mode, create a class of traffic matching experimental bit 5, called "exp-5-traffic".

```
class-map match-all exp-5-traffic
match mpls experimental 5
```

1. Create a policy named “output-interface-policy”. According to that policy, packets in the class “exp-5-traffic” are put in the priority queue (which is rate-limited to 32 kbits/sec).

```
policy-map output-interface-policy
class exp-5-traffic
priority 32
```

1. The policy is applied to packets exiting interface POS0/0:

```
interface POS0/0
service-policy output output-interface-policy
```

As a result of all the above configuration lines, packets entering the Head-2 router via interface FE2/1/0 and destined for subnet 26.1.1.0 have their IP precedence field set to 5. It is assumed that no other packets entering this router (on any interface) are using this precedence. (If this cannot be assumed, an additional configuration must be added to mark all such packets with another precedence value.) When exiting this router via interface POS0/0, packets marked with precedence 5 are placed in the priority queue.

Note Packets entering the router via FE2/1/0 and exiting through POS0/0 enter as IP packets and exit as MPLS packets.

Tunnel Midpoint Configuration Mid-1 Example

All four interfaces on the midpoint router are configured identically to the outbound interface of the head router (except, of course, for the IDs of the individual interfaces):

Configuring the Pools and Tunnels

At the device level:

```
router-3(config)# ip cef distributed
router-3(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-3(config)# router isis`	`router ospf 100`
`router-3(config-router)# net 49.0000.2400.0000.0011.00`	`redistribute connected`
`router-3(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-3(config-router)# is-type level-1`	`network 11.1.1.0 0.0.0.255 area 0`
`router-3(config-router)# mpls traffic-eng level-1`	`network 24.1.1.1 0.0.0.0 area 0`
`router-3(config-router)# passive-interface Loopback0`	`network 12.1.1.0 0.0.0.255 area 0`
`router-3(config-router)#`	`network 13.1.1.0 0.0.0.255 area 0`
`router-3(config-router)#`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-3(config-router)# mpls traffic-eng router-id Loopback0
router-3(config-router)# exit
```

Create a virtual interface:

```
router-3(config)# interface Loopback0
router-3(config-if)# ip address 24.1.1.1 255.255.255.255
router-3(config-if)# exit
```

At the physical interface level (ingress):

```
router-3(config)# interface pos2/1
router-3(config-if)# ip address 10.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
router-3(config)# interface pos1/1
router-3(config-if)# ip address 11.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the physical interface level (egress):

```
router-3(config)# interface pos3/1
router-3(config-if)# ip address 12.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
router-3(config)# interface pos4/1
router-3(config-if)# ip address 13.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

Tunnel Midpoint Configuration Mid-2 Example

Both interfaces on the midpoint router are configured identically to the outbound interface of the head router (except, of course, for the IDs of the individual interfaces):

At the device level:

```
router-5(config)# ip cef distributed
router-5(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-5(config)# router isis`	`router ospf 100`

router-5(config-router)# **net 49.2500.1000.0000.0012.00**	**redistribute connected**
router-5(config-router)# **metric-style wide**	**network 13.1.1.0 0.0.0.255 area 0**
router-5(config-router)# **is-type level-1**	**network 14.1.1.0 0.0.0.255 area 0**
router-5(config-router)# **mpls traffic-eng level-1**	**network 25.1.1.1 0.0.0.0 area 0**
router-5(config-router)# **passive-interface Loopback0**	**mpls traffic-eng area 0**

[now one resumes the common command set]:

```
router-5(config-router)# mpls traffic-eng router-id Loopback0
router-5(config-router)# exit
```

Create a virtual interface:

```
router-5(config)# interface Loopback0
router-5(config-if)# ip address 25.1.1.1 255.255.255.255
router-5(config-if)# exit
```

At the physical interface level (ingress):

```
router-5(config)# interface pos1/1
router-5(config-if)# ip address 13.1.1.2 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

At the physical interface level (egress):

```
router-5(config)# interface pos2/1
router-5(config-if)# ip address 14.1.1.1 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

Tunnel Tail Configuration Example

The inbound interfaces on the tail router are configured identically to the inbound interfaces of the midpoint routers (except, of course, for the ID of each particular interface):

At the device level:

```
router-4(config)# ip cef distributed
router-4(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-4(config)# router isis`	`router ospf 100`
`router-4(config-router)# net 49.0000.2700.0000.0000.00`	`redistribute connected`
`router-4(config-router)# metric-style wide`	`network 12.1.1.0 0.0.0.255 area 0`
`router-4(config-router)# is-type level-1`	`network 14.1.1.0 0.0.0.255 area 0`
`router-4(config-router)# mpls traffic-eng level-1`	`network 27.1.1.1 0.0.0.0 area 0`
`router-4(config-router)# passive-interface Loopback0`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-4(config-router)# mpls traffic-eng router-id Loopback0
router-4(config-router)# exit
```

Create a virtual interface:

```
router-4(config)# interface Loopback0
router-4(config-if)# ip address 27.1.1.1 255.255.255.255
router-4(config-if)# exit
```

At the physical interface (ingress):

```
router-4(config)# interface pos2/1
router-4(config-if)# ip address 12.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
router-4(config)# interface pos2/2
router-4(config-if)# ip address 14.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
```

Because the tunnel ends on the tail (does not include any outbound interfaces of the tail router), no outbound QoS configuration is used.

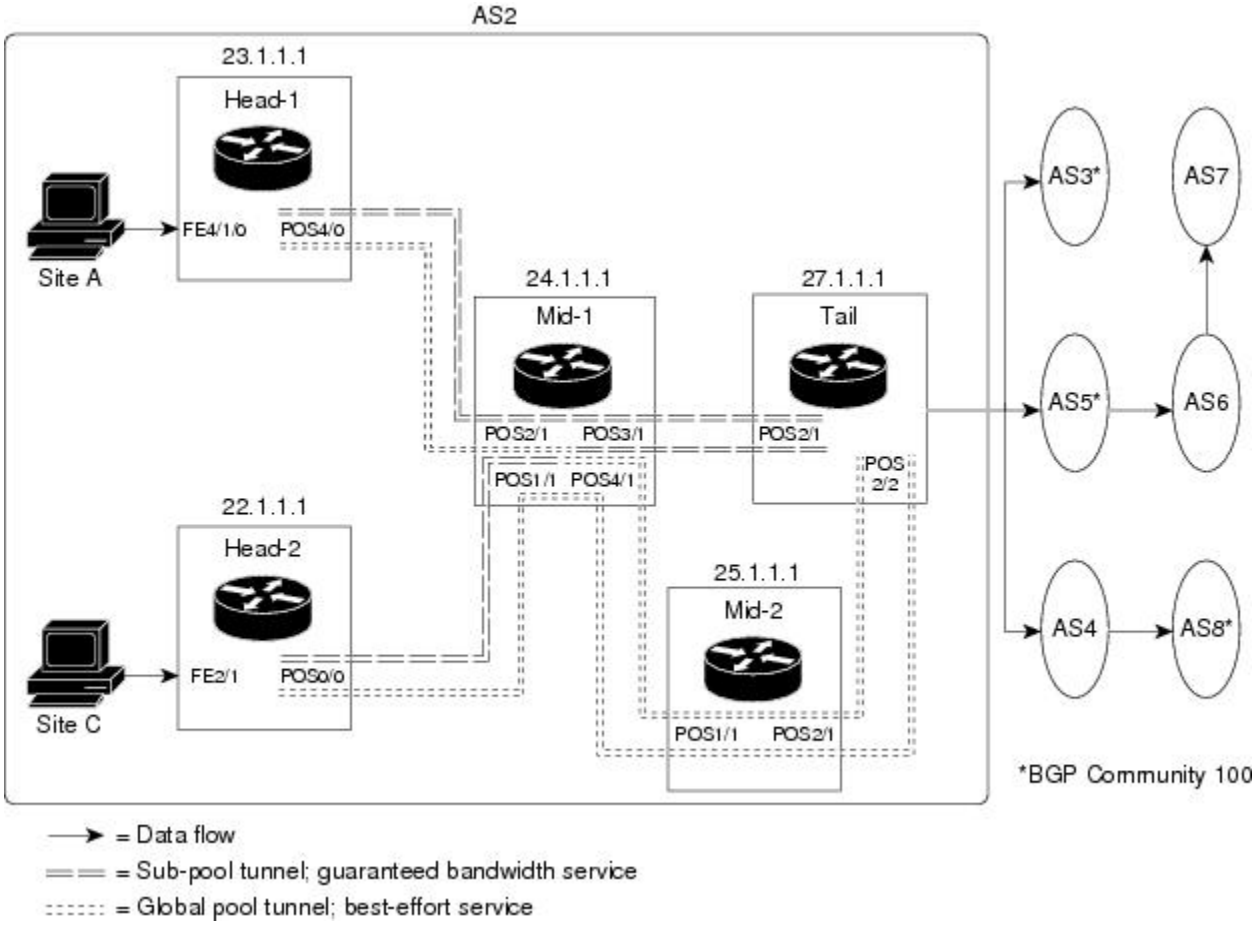
Many Destination Prefixes: Example

The figure below illustrates a topology for guaranteed bandwidth services whose destinations are a set of prefixes. Those prefixes usually share some common properties such as belonging to the same Autonomous System (AS) or transiting through the same AS. Although the individual prefixes may change dynamically because of route flaps in the downstream autonomous systems, the properties the prefixes share will not change. Policies addressing the destination prefix set are enforced through Border Gateway Protocol (BGP), which is described in the following documents:

- "Configuring QoS Policy Propagation via Border Gateway Protocol" in the *Cisco IOS Quality of Service Solutions Configuration Guide*
- "Configuring BGP" in the *Cisco IOS IP and IP Routing Configuration Guide*
- "BGP Commands" in the *Cisco IOS IP and IP Routing Command Reference*
- "BGP-Policy Command" in the *Cisco IOS Quality of Service Solutions Command Reference*

In this example, three guaranteed bandwidth services are offered, each coming through a 7500 or a 12000 edge device:

- Traffic coming from Site A (defined as all traffic arriving at interface FE4/1/0) and from Site C (defined as all traffic arriving at interface FE2/1) destined to AS5
- Traffic coming from Sites A and C that transits AS5 but is not destined to AS5. (In the figure, the transiting traffic will go to AS6 and AS7)
- Traffic coming from Sites A and C destined to prefixes advertised with a particular BGP community attribute (100:1). In this example, Autonomous Systems #3, #5, and #8 are the BGP community assigned the attribute 100:1.



The applicability of guaranteed bandwidth service is not limited to the three types of multiple destination scenarios described above. There is not room in this document to present all possible scenarios. These three were chosen as representative of the wide range of possible deployments.

The guaranteed bandwidth services run through two sub-pool tunnels:

- From the Head-1 router, 23.1.1.1, to the tail
- From the Head-2 router, 22.1.1.1, to that same tail

In addition, a global pool tunnel has been configured from each head end, to carry best-effort traffic to the same destinations. All four tunnels use the same tail router, even though they have different heads and differ in their passage through the midpoints. (Of course in the real world there would be many more midpoints than just the two shown here.)

All POS interfaces in this example are OC3, whose capacity is 155 Mbps.

Configuring a multi-destination guaranteed bandwidth service involves:

1. Building a sub-pool MPLS-TE tunnel
2. Configuring DiffServ QoS
3. Configuring QoS Policy Propagation via BGP (QPPB)
4. Mapping traffic onto the tunnels

All of these tasks are included in the following example.

Configuration of Tunnel Head-1 Example

First we recapitulate commands that establish a sub-pool tunnel (commands presented earlier in MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples, on page 1758) and now we also configure a global pool tunnel. Additionally, we present QoS and BGP commands that guarantee end-to-end service on the sub-pool tunnel. (With the 7500(VIP) router, Modular QoS CLI is used).

At the device level:

```
router-1(config)# ip cef distributed
router-1(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-1(config)# router isis`	`router ospf 100`
`router-1(config-router)# net 49.0000.1000.0000.0010.00`	`redistribute connected`
`router-1(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-1(config-router)# is-type level-1`	`network 23.1.1.1 0.0.0.0 area 0`
`router-1(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-1(config-router)# mpls traffic-eng router-id Loopback0
router-1(config-router)# exit
```

Create a virtual interface:

```
router-1(config)# interface Loopback0
router-1(config-if)# ip address 23.1.1.1 255.255.255.255
router-1(config-if)# exit
```

At the outgoing physical interface:

```
router-1(config)# interface pos4/0
router-1(config-if)# ip address 10.1.1.1 255.0.0.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

At one tunnel interface, create a sub-pool tunnel:

```
router-1(config)# interface Tunnel1
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 27.1.1.1
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 40000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 explicit name gbs-path1
router-1(config-if)# exit
```

and at a second tunnel interface, create a global pool tunnel:

```
router-1(config)# interface Tunnel2
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 27.1.1.1
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth 80000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 explicit name \ best-effort-path1
router-1(config-if)# exit
```

In this example explicit paths are used instead of dynamic, to ensure that best-effort traffic and guaranteed bandwidth traffic will travel along different paths.

At the device level:

```
router-1(config)# ip explicit-path name gbs-path1
router-1(config-ip-expl-path)# next-address 24.1.1.1
router-1(config-ip-expl-path)# next-address 27.1.1.1
router-1(config-ip-expl-path)# exit
router-1(config)# ip explicit-path name best-effort-path1
router-1(config-ip-expl-path)# next-address 24.1.1.1
router-1(config-ip-expl-path)# next-address 25.1.1.1
router-1(config-ip-expl-path)# next-address 27.1.1.1
router-1(config-ip-expl-path)# exit
```

Note that autoroute is not used, as that could cause the Interior Gateway Protocol (IGP) to send other traffic down these tunnels.

At the inbound physical interface (in the figure above this is FE4/1/0), packets received are rate-limited to:

1. a rate of 30 Mbps
2. a normal burst of 1 MB
3. a maximum burst of 2 MB

Packets that are mapped to qos-group 6 and that conform to the rate-limit are marked with experimental value 5 and the BGP destination community string, and are forwarded; packets that do not conform (exceed action) are dropped:

```
router-1(config)# interface FastEthernet4/1/0
router-1(config-if)# rate-limit input qos-group 6 30000000 1000000 2000000 \
 conform-action set-mpls-exp-transmit 5 exceed-action drop
router-1(config-if)# bgp-policy destination ip-qos-map
router-1(config-if)# exit
```

At the device level create a class of traffic called “exp5-class” that has MPLS experimental bit set to 5:

```
router-1(config)# class-map match-all exp5-class
```

```
router-1(config-cmap)# match mpls experimental 5
router-1(config-cmap)# exit
```

Create a policy that creates a priority queue for “exp5-class”:

```
router-1(config)# policy-map core-out-policy
router-1(config-pmap)# class exp5-class
router-1(config-pmap-c)# priority 100000
router-1(config-pmap-c)# exit
router-1(config-pmap)# class class-default
router-1(config-pmap-c)# bandwidth 55000
router-1(config-pmap-c)# exit
router-1(config-pmap)# exit
```

The policy is applied to packets exiting the outbound interface POS4/0.

```
router-1(config)# interface POS4/0
router-1(config-if)# service-policy output core-out-policy
```

Create a table map under BGP to map (tie) the prefixes to a qos-group. At the device level:

```
router-1(config)# ip bgp-community new-format
router-1(config)# router bgp 2
router-1(config-router)# no synchronization
router-1(config-router)# table-map set-qos-group
router-1(config-router)# bgp log-neighbor-changes
router-1(config-router)# neighbor 27.1.1.1 remote-as 2
router-1(config-router)# neighbor 27.1.1.1 update-source Loopback0
router-1(config-router)# no auto-summary
router-1(config-router)# exit
```

Create a distinct route map for this service. This includes setting the next-hop of packets matching 29.1.1.1 so they will be mapped onto Tunnel #1 (the guaranteed bandwidth service tunnel). At the device level:

```
router-1(config)# route-map set-qos-group permit 10
router-1(config-route-map)# match as-path 100
router-1(config-route-map)# set ip qos-group 6
router-1(config-route-map)# set ip next-hop 29.1.1.1
router-1(config-route-map)# exit
router-1(config)# ip as-path access-list 100 permit ^5$
```

Create a distinct route map for this service. (Its traffic will go to AS6 and AS7).

At the device level:

```
router-1(config)# route-map set-qos-group permit 10
router-1(config-route-map)# match as-path 101
router-1(config-route-map)# set ip qos-group 6
router-1(config-route-map)# set ip next-hop 29.1.1.1
router-1(config-route-map)# exit
router-1(config)# ip as-path access-list 101 permit _5_
```

Create a distinct route map for all traffic destined to prefixes that have community value 100:1. This traffic will go to AS3, AS5, and AS8.

At the device level:

```
router-1(config)# route-map set-qos-group permit 10
router-1(config-route-map)# match community 20
router-1(config-route-map)# set ip qos-group 6
```

```
router-1(config-route-map)# set ip next-hop 29.1.1.1
router-1(config-route-map)# exit
router-1(config)# ip community-list 20 permit 100:1
```

Map all guaranteed bandwidth traffic onto Tunnel #1:

```
router-1(config)# ip route 29.1.1.1 255.255.255.255 Tunnel1
```

Map all best-effort traffic onto Tunnel #2:

```
router-1(config)# ip route 30.1.1.1 255.255.255.255 Tunnel2
```

Configuration of Tunnel Head-2 Example

As with the Head-1 device and interfaces, the following Head-2 configuration first presents commands that establish a sub-pool tunnel (commands presented earlier in MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples, on page 1758) and then also configures a global pool tunnel. After that it presents QoS and BGP commands that guarantee end-to-end service on the sub-pool tunnel. (Because this is a 7500 (VIP) router, Modular QoS CLI is used).

At the device level:

```
router-2(config)# ip cef distributed
router-2(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-2(config)# router isis`	`router ospf 100`
`router-2(config-router)# net 49.0000.1000.0000.0011.00`	`redistribute connected`
`router-2(config-router)# metric-style wide`	`network 11.1.1.0 0.0.0.255 area 0`
`router-2(config-router)# is-type level-1`	`network 22.1.1.1 0.0.0.0 area 0`
`router-2(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-2(config-router)# mpls traffic-eng router-id Loopback0
router-2(config-router)# exit
```

Create a virtual interface:

```
router-2(config)# interface Loopback0
router-2(config-if)# ip address 22.1.1.1 255.255.255.255
router-2(config-if)# exit
```

At the outgoing physical interface:

```
router-2(config)# interface pos0/0
router-2(config-if)# ip address 11.1.1.1 255.0.0.0
router-2(config-if)# mpls traffic-eng tunnels
router-2(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-2(config-if)# ip router isis
[and in all cases]:
router-2(config-if)# exit
```

At one tunnel interface, create a sub-pool tunnel:

```
router-2(config)# interface Tunnel3
router-2(config-if)# ip unnumbered Loopback0
router-2(config-if)# tunnel destination 27.1.1.1
router-2(config-if)# tunnel mode mpls traffic-eng
router-2(config-if)# tunnel mpls traffic-eng priority 0 0
router-2(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000
router-2(config-if)# tunnel mpls traffic-eng path-option 1 explicit name gbs-path2
router-2(config-if)# exit
```

and at a second tunnel interface, create a global pool tunnel:

```
router-2(config)# interface Tunnel4
router-2(config-if)# ip unnumbered Loopback0
router-2(config-if)# tunnel destination 27.1.1.1
router-2(config-if)# tunnel mode mpls traffic-eng
router-2(config-if)# tunnel mpls traffic-eng priority 0 0
router-2(config-if)# tunnel mpls traffic-eng bandwidth 70000
router-2(config-if)# tunnel mpls traffic-eng path-option 1 explicit name \ best-effort-path2
router-2(config-if)# exit
```

In this example explicit paths are used instead of dynamic, to ensure that best-effort traffic and guaranteed bandwidth traffic will travel along different paths.

At the device level:

```
router-2(config)# ip explicit-path name gbs-path2
router-2(config-ip-expl-path)# next-address 24.1.1.1
router-2(config-ip-expl-path)# next-address 27.1.1.1
router-2(config-ip-expl-path)# exit
router-2(config)# ip explicit-path name best-effort-path2
router-2(config-ip-expl-path)# next-address 24.1.1.1
router-2(config-ip-expl-path)# next-address 25.1.1.1
router-2(config-ip-expl-path)# next-address 27.1.1.1
router-2(config-ip-expl-path)# exit
```

Note that autoroute is not used, as that could cause the Interior Gateway Protocol (IGP) to send other traffic down these tunnels.

At the inbound physical interface (in the figure above this is FE2/1), packets received are rate-limited to:

1. a rate of 30 Mbps
2. a normal burst of 1 MB
3. a maximum burst of 2 MB

Packets that are mapped to qos-group 6 and that conform to the rate-limit are marked with experimental value 5 and the BGP destination community string, and are forwarded; packets that do not conform (exceed action) are dropped:

```
router-2(config)# interface FastEthernet2/1
router-2(config-if)# rate-limit input qos-group 6 30000000 1000000 2000000 \ conform-action
 set-mpls-exp-transmit 5 exceed-action drop
router-2(config-if)# bgp-policy destination ip-qos-map
router-1(config-if)# exit
```

At the device level create a class of traffic called “exp5-class” that has MPLS experimental bit set to 5:

```
router-2(config)# class-map match-all exp5-class
router-2(config-cmap)# match mpls experimental 5
router-2(config-cmap)# exit
```

Create a policy that creates a priority queue for “exp5-class”:

```
router-2(config)# policy-map core-out-policy
router-2(config-pmap)# class exp5-class
router-2(config-pmap-c)# priority 100000
router-2(config-pmap-c)# exit
router-2(config-pmap)# class class-default
router-2(config-pmap-c)# bandwidth 55000
router-2(config-pmap-c)# exit
router-2(config-pmap)# exit
```

The policy is applied to packets exiting interface POS0/0:

```
interface POS0/0
service-policy output core-out-policy
```

As a result of all the above configuration lines, packets entering the Head-2 router via interface FE2/1 and destined for AS5, BGP community 100:1, or transiting AS5 will have their experimental field set to 5. It is assumed that no other packets entering this router (on any interface) are using this exp bit value. (If this cannot be assumed, an additional configuration must be added to mark all such packets with another experimental value.) When exiting this router via interface POS0/0, packets marked with experimental value 5 are placed into the priority queue.

Note Packets entering the router via FE2/1 and exiting through POS0/0 enter as IP packets and exit as MPLS packets.

Create a table map under BGP to map (tie) the prefixes to a qos-group. At the device level:

```
router-2(config)# ip bgp-community new-format
router-2(config)# router bgp 2
router-2(config-router)# no synchronization
router-2(config-router)# table-map set-qos-group
router-2(config-router)# bgp log-neighbor-changes
router-2(config-router)# neighbor 27.1.1.1 remote-as 2
router-2(config-router)# neighbor 27.1.1.1 update-source Loopback0
router-2(config-router)# no auto-summary
router-2(config-router)# exit
```

Create a distinct route map for this service. This includes setting the next-hop of packets matching 29.1.1.1 so they will be mapped onto Tunnel #3 (the guaranteed bandwidth service tunnel). At the device level:

```
router-2(config)# route-map set-qos-group permit 10
router-2(config-route-map)# match as-path 100
router-2(config-route-map)# set ip qos-group 6
router-2(config-route-map)# set ip next-hop 29.1.1.1
router-2(config-route-map)# exit
router-2(config)# ip as-path access-list 100 permit ^5$
```

Create a distinct route map for this service. (Its traffic will go to AS6 and AS7).

At the device level:

```
router-2(config)# route-map set-qos-group permit 10
router-2(config-route-map)# match as-path 101
router-2(config-route-map)# set ip qos-group 6
router-2(config-route-map)# set ip next-hop 29.1.1.1
router-2(config-route-map)# exit
router-2(config)# ip as-path access-list 101 permit _5_
```

Create a distinct route map for all traffic destined to prefixes that have community value 100:1. This traffic will go to AS3, AS5, and AS8.

At the device level:

```
router-2(config)# route-map set-qos-group permit 10
router-2(config-route-map)# match community 20
router-2(config-route-map)# set ip qos-group 6
router-2(config-route-map)# set ip next-hop 29.1.1.1
router-2(config-route-map)# exit
router-2(config)# ip community-list 20 permit 100:1
```

Map all guaranteed bandwidth traffic onto Tunnel #3:

```
router-2(config)# ip route 29.1.1.1 255.255.255.255 Tunnel3
```

Map all best-effort traffic onto Tunnel #4:

```
router-2(config)# ip route 30.1.1.1 255.255.255.255 Tunnel4
```

Tunnel Midpoint Configuration Mid-1 Example

All four interfaces on the midpoint router are configured very much like the outbound interface of the head router. The strategy is to have all mid-point routers in this Autonomous System ready to carry future as well as presently configured sub-pool and global pool tunnels.

At the device level:

```
router-3(config)# ip cef distributed
router-3(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-3(config)# router isis`	`router ospf 100`

router-3(config-router)# **net 49.0000.2400.0000.0011.00**	**redistribute connected**
router-3(config-router)# **metric-style wide**	**network 10.1.1.0 0.0.0.255 area 0**
router-3(config-router)# **is-type level-1**	**network 11.1.1.0 0.0.0.255 area 0**
router-3(config-router)# **mpls traffic-eng level-1**	**network 24.1.1.1 0.0.0.0 area 0**
router-3(config-router)#	**network 12.1.1.0 0.0.0.255 area 0**
router-3(config-router)#	**network 13.1.1.0 0.0.0.255 area 0**
router-3(config-router)#	**mpls traffic-eng area 0**

[now one resumes the common command set]:

```
router-3(config-router)# mpls traffic-eng router-id Loopback0
router-3(config-router)# exit
```

Create a virtual interface:

```
router-3(config)# interface Loopback0
router-3(config-if)# ip address 24.1.1.1 255.255.255.255
router-3(config-if)# exit
```

At the physical interface level (ingress):

```
router-3(config)# interface pos2/1
router-3(config-if)# ip address 10.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit

router-3(config)# interface pos1/1
router-3(config-if)# ip address 11.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the physical interface level (egress), through which two sub-pool tunnels currently exit:

```
router-3(config)# interface pos3/1
router-3(config-if)# ip address 12.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the physical interface level (egress), through which two global pool tunnels currently exit:

```
router-3(config)# interface pos4/1
router-3(config-if)# ip address 13.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

Tunnel Midpoint Configuration Mid-2 Example

Both interfaces on this midpoint router are configured like the outbound interfaces of the Mid-1 router.

At the device level:

```
router-5(config)# ip cef distributed
router-5(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-5(config)# router isis`	`router ospf 100`
`router-5(config-router)# net 49.2500.1000.0000.0012.00`	`redistribute connected`
`router-5(config-router)# metric-style wide`	`network 13.1.1.0 0.0.0.255 area 0`
`router-5(config-router)# is-type level-1`	`network 14.1.1.0 0.0.0.255 area 0`
`router-5(config-router)# mpls traffic-eng level-1`	`network 25.1.1.1 0.0.0.0 area 0`
`router-5(config-router)#`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-5(config-router)# mpls traffic-eng router-id Loopback0
router-5(config-router)# exit
```

Create a virtual interface:

```
router-5(config)# interface Loopback0
router-5(config-if)# ip address 25.1.1.1 255.255.255.255
router-5(config-if)# exit
```

At the physical interface level (ingress):

```
router-5(config)# interface pos1/1
router-5(config-if)# ip address 13.1.1.2 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

At the physical interface level (egress):

```
router-5(config)# interface pos2/1
router-5(config-if)# ip address 14.1.1.1 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

Tunnel Tail Configuration Example

The inbound interfaces on the tail router are configured much like the outbound interfaces of the midpoint routers:

At the device level:

```
router-4(config)# ip cef distributed
router-4(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right. In the case of OSPF, one must advertise two new loopback interfaces--29.1.1.1 and 30.1.1.1 in our example--which are defined in the QoS Policy Propagation section, further along on this page]:

`router-4(config)# router isis`	`router ospf 100`
`router-4(config-router)# net 49.0000.2700.0000.0000.00`	`redistribute connected`
`router-4(config-router)# metric-style wide`	`network 12.1.1.0 0.0.0.255 area 0`
`router-4(config-router)# is-type level-1`	`network 14.1.1.0 0.0.0.255 area 0`

router-4(config-router)# **mpls traffic-eng level-1**	**network 27.1.1.1 0.0.0.0 area 0**
router-4(config-router)#	**network 29.1.1.1 0.0.0.0 area 0**
router-4(config-router)#	**network 30.1.1.1 0.0.0.0 area 0**
router-4(config-router)#	**mpls traffic-eng area 0**

[now one resumes the common command set, taking care to include the two additional loopback interfaces]:

```
router-4(config-router)# mpls traffic-eng router-id Loopback0
router-4(config-router)# mpls traffic-eng router-id Loopback1
router-4(config-router)# mpls traffic-eng router-id Loopback2
router-4(config-router)# exit
```

Create a virtual interface:

```
router-4(config)# interface Loopback0
router-4(config-if)# ip address 27.1.1.1 255.255.255.255
router-4(config-if)# exit
```

At the physical interface (ingress):

```
router-4(config)# interface pos2/1
router-4(config-if)# ip address 12.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
router-4(config)# interface pos2/2
router-4(config-if)# ip address 14.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
```

On the tail device, one must configure a separate virtual loopback IP address for each class-of-service terminating here. The headend routers need these addresses to map traffic into the proper tunnels. In the current example, four tunnels terminate on the same tail device but they represent only two service classes, so only two additional loopback addresses are needed:

Create two virtual interfaces:

```
router-4(config)# interface Loopback1
router-4(config-if)# ip address 29.1.1.1 255.255.255.255
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
router-4(config)# interface Loopback2
router-4(config-if)# ip address 30.1.1.1 255.255.255.255
```

```
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
```

At the device level, configure BGP to send the community to each tunnel head:

```
router-4(config)# ip bgp-community new-format
router-4(config)# router bgp 2
router-4(config-router)# neighbor 23.1.1.1 send-community
router-4(config-router)# neighbor 22.1.1.1 send-community
router-4(config-router)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IS-IS	Cisco IOS IP Routing: ISIS Command Reference
MPLS commands	Cisco IOS MPLS Command Reference
OSPF	Cisco IOS IP Routing: OSPF Command Reference
QoS	Cisco IOS Quality of Service Solutions Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 3270	*Requirements for Support of Diff-Serv-aware MPLS Traffic Engineering* Multi-Protocol Label Switching (MPLS) Support of Differentiated Services F. Le Faucheur, L. Wu, B. Davie, P. Vaananen, R. Krishnan, P. Cheval, & J. Heinanen
RFC 4124	Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering ed. by F. Le Faucheur
RFC 4127	Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering ed. by F. Le Faucheur
RFC 4125	Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering by F. Le Faucheur & W. Lai

Standard/RFC	Title
IETF Diff-Serv-aware MPLS Traffic Engineering	The new concept of "Class-Type" defined in the IETF Standard corresponds to the prior concept of “bandwidth pool” that was implemented in the original version of DS-TE. Likewise, the two bandwidth pools implemented in the original version of DS-TE (global pool and sub-pool) correspond to two of the IETF Standard’s new Class-Types (Class-Type 0 and Class-Type 1, respectively).

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Glossary

This section defines acronyms and words that may not be readily understood.

AS --Autonomous System. A collection of networks under a common administration, sharing a common routing strategy and identified by a unique 16-bit number (assigned by the Internet Assigned Numbers Authority).

BGP --Border Gateway Protocol. The predominant interdomain routing protocol. It is defined by RFC 1163. Version 4 uses route aggregation mechanisms to reduce the size of routing tables.

CBR --Constraint Based Routing. The computation of traffic paths that simultaneously satisfy label-switched path attributes and current network resource limitations.

CEF --Cisco Express Forwarding. A means for accelerating the forwarding of packets within a router, by storing route lookup information in several data structures instead of in a route cache.

CLI --Command Line Interface. Cisco’s interface for configuring and managing its routers.

DS-TE --Diff Serv-aware Traffic Engineering. The capability to configure two bandwidth pools on each link, a *global pool* and a *sub-pool*. MPLS traffic engineering tunnels using the sub-pool bandwidth can be configured with Quality of Service mechanisms to deliver guaranteed bandwidth services end-to-end across the network. Simultaneously, tunnels using the global pool can convey DiffServ traffic.

flooding --A traffic passing technique used by switches and bridges in which traffic received on an interface is sent out through all of the interfaces of that device except the interface on which the information was originally received.

GB queue --Guaranteed Bandwidth queue. A per-hop behavior (PHB) used exclusively by the strict guarantee traffic. If delay/jitter guarantees are sought, the diffserv Expedited Forwarding queue (EF PHB) is used. If only bandwidth guarantees are sought, the diffserv Assured Forwarding PHB (AF PHB) is used.

Global Pool --The total bandwidth allocated to an MPLS traffic engineering link.

IGP --Interior Gateway Protocol. An internet protocol used to exchange routing information within an autonomous system. Examples of common internet IGPs include IGRP, OSPF, and RIP.

label-switched path (LSP) tunnel --A configured connection between two routers, using label switching to carry the packets.

IS-IS --Intermediate System-to-Intermediate System. A link-state hierarchical routing protocol, based on DECnet Phase V routing, whereby nodes exchange routing information based on a single metric, to determine network topology.

LCAC --Link-level (per-hop) call admission control.

LSP --Label-switched path (see above). Also Link-state packet--A broadcast packet used by link-state protocols that contains information about neighbors and path costs. LSPs are used by the receiving routers to maintain their routing tables. Also called link-state advertisement (LSA).

MPLS --Multi-Protocol Label Switching (formerly known as Tag Switching). A method for directing packets primarily through Layer 2 switching rather than Layer 3 routing, by assigning the packets short fixed-length labels at the ingress to an MPLS cloud, using the concept of forwarding equivalence classes. Within the MPLS domain, the labels are used to make forwarding decisions mostly without recourse to the original packet headers.

MPLS TE --MPLS Traffic Engineering (formerly known as “RRR” or Resource Reservation Routing). The use of label switching to improve traffic performance along with an efficient use of network resources.

OSPF --Open Shortest Path First. A link-state, hierarchical IGP routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

RSVP --Resource reSerVation Protocol. An IETF protocol used for signaling requests (to set aside internet services) by a customer before that customer is permitted to transmit data over that portion of the network.

Sub-pool --The more restrictive bandwidth in an MPLS traffic engineering link. The sub-pool is a portion of the link’s overall global pool bandwidth.

TE --Traffic engineering. The application of scientific principles and technology to measure, model, and control internet traffic in order to simultaneously optimize traffic performance and network resource utilization.

Feature Information for MPLS Traffic Engineering - DiffServ Aware (DS-TE)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

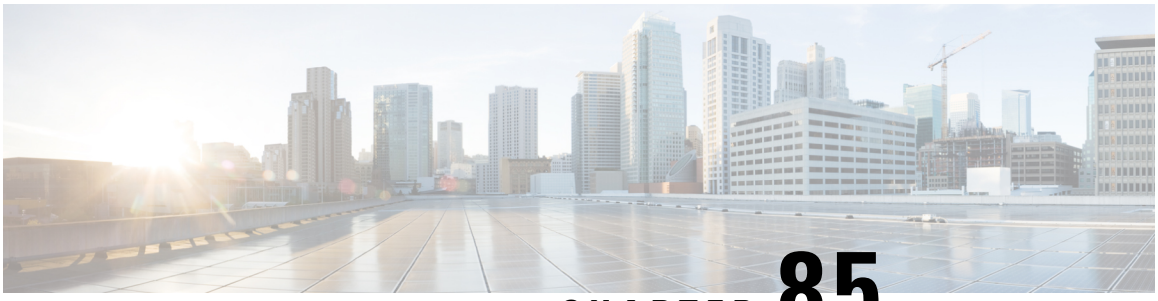
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 161: Feature Information for MPLS Traffic Engineering - DiffServe Aware (DS-TE)

Feature Name	Releases	Feature Information
MPLS Traffic Engineering - DiffServ Aware (DS-TE)	12.0(11) ST 12.0(14) ST 12.0(14) ST-1 12.0(22)S 12.2(14)S 12.2(18)S 12.2(18)SXD 12.2(28)SB 12.2(33)SRB Cisco IOS XE Release 3.5S	

Feature Name	Releases	Feature Information
		The Multiprotocol Label Switching Traffic Engineering (MPLS TE) - DiffServ-Aware Traffic Engineering (DS-TE) feature enables service providers to perform separate admission control and separate route computation for discrete subsets of traffic (for example, voice and data traffic). When DS-TE is combined with other Cisco software features such as QoS, the service provider can: • Develop QoS services for end customers based on *signaled* rather than *provisioned* QoS • Build the higher-revenue generating "strict-commitment" QoS services, without over-provisioning • Offer virtual IP leased-line, Layer 2 service emulation, and point-to-point guaranteed bandwidth services including voice-trunking • Enjoy the scalability properties offered by MPLS. DS-TE feature introduced in Cisco IOS Release 12.0(11) ST. In Cisco IOS Release 12.0(14) ST-1, support was added for guaranteed bandwidth service directed to many destination prefixes (for example, guaranteed bandwidth service destined to an autonomous system or to a BGP community). In Cisco IOS Release 12.0(14) ST-1, support was added for the IS-IS Interior Gateway Protocol. In Cisco IOS Release 12.0(14) ST, support was added for the Cisco Series 7500(VIP) platform. Feature was implemented in Cisco

Feature Name	Releases	Feature Information
		IOS Release 12.0(22)S. Feature was integrated into Cisco IOS Release 12.2(14)S. Feature was implemented in Cisco IOS Release 12.2(18)S. Feature was implemented in Cisco IOS Release 12.2(18)SXD. Feature was implemented in Cisco IOS Release 12.2(28)SB. In Cisco IOS Release 12.2(33)SRB, this feature was augmented to include the new IETF-Standard functionality of DS-TE, as described in RFCs 3270, 4124, 4125, and 4127. Feature was implemented in Cisco IOS XE Release 3.5S.



CHAPTER 85

MPLS DiffServ Tunneling Modes

MPLS DiffServ Tunneling Modes allows service providers to manage the quality of service (QoS) that a router will provide to a Multiprotocol Label Switching (MPLS) packet in an MPLS network. MPLS DiffServ Tunneling Modes conforms to the IETF draft standard for Uniform, Short Pipe, and Pipe modes. It also conforms to Cisco-defined extensions for scalable command line interface (CLI) management of those modes at customer edge, provider edge, and core routers.

The following features are supported on MPLS DiffServ Tunneling Modes:

- MPLS per-hop behavior (PHB) layer management.
- There is improved scalability of the MPLS layer management by control on managed customer edge (CE) routers.
- MPLS can “tunnel” a packet’s QoS (that is, the QoS is transparent from edge to edge).
- The MPLS experimental (MPLS EXP) field can be marked differently and independently of the PHB marked in the IP Precedence or differentiated services code point (DSCP) field.
- There are three MPLS QoS tunneling modes for the operation and interaction between the DiffServ marking in the IP header and the DiffServ marking in the MPLS header: Pipe mode with an explicit NULL LSP, Short Pipe mode, and Uniform mode. Pipe mode with an explicit NULL LSP and Short Pipe mode allow an MPLS network to transparently tunnel the DiffServ marking of packets.

MPLS DiffServ Tunneling Modes has the following benefits:

- Tunneling modes provide added QoS functionality by the creative manipulation of the MPLS EXP field during label imposition, forwarding, and label disposition.
- Tunneling modes provide a common set of PHBs to different service provider customers.
- Pipe mode provides transparency and customized edge service.
- Pipe mode with an explicit NULL LSP improves the scalability of management by performing per-customer packet metering and marking closer to the service provider’s customer networks.
- Pipe mode with an explicit NULL LSP provides QoS transparency by ensuring that customer’s packets will not be re-marked in the service provider’s network.
- In Pipe mode with an explicit NULL LSP, the explicit NULL LSP applies the service provider’s PHBs on the ingress CE-to-PE link.
- In Pipe mode with an explicit NULL LSP, the service provider’s PHBs are applied on the egress PE-to-CE link.

- Short Pipe mode provides transparency, standard edge service, and scalability.
- Short Pipe mode provides PHB management on the PE router. The customer's set of PHBs is applied on both the egress PE-to-CE link and on the ingress CE-to-PE link.
- Customers are likely to use Uniform mode if they have no markings or few markings. The customer lets the Internet service provider (ISP) mark the packets and retain their markings.
- In Uniform mode, all changes to QoS markings are reflected at each level (that is, IGP, BGP, and IP).
- In Uniform mode, if a QoS marking is changed in the MPLS network, it is changed in the IP packet too.

Prerequisites for MPLS DiffServ Tunneling Modes

- Set up the network to run MPLS.
- Enable IP Cisco Express Forwarding (CEF).
- Define the Service Level Agreement (SLA).
- Know each customer's per-hop behavior.
  - What do customers expect you to provide?
  - Are customers going to mark the traffic?
- Identify whether the customer's traffic will be voice or data.
- Determine the topology and interfaces that need to be configured.
- Understand how IP and MPLS packets are forwarded.

Restrictions for MPLS DiffServ Tunneling Modes

- A single label-switched path (LSP) can support up to eight classes of traffic (that is, eight PHBs) because the MPLS EXP field is a 3-bit field.
- MPLS DiffServ Tunneling Modes does not support L-LSPs. Only E-LSPs are supported.

Information About MPLS DiffServ Tunneling Modes

QoS and Its Use in MPLS Tunneling

This section includes the following subsections:

What is QoS

Critical applications must be guaranteed the network resources they need, despite a varying network traffic load. QoS is a set of techniques that manage the following:

- Network bandwidth--Noncritical traffic is prevented from using bandwidth that critical applications need. The main cause of congestion is lack of bandwidth.
- Network delay (also called latency)--The time required to move a packet from the source to the destination over a path.
- Jitter--The interpacket delay variance; that is, the difference between interpacket arrival and departure. Jitter can cause data loss.
- Packet loss--The dropping of packets.

Service providers offering MPLS VPN and traffic engineering (TE) services can provide varying levels of QoS for different types of network traffic. For example, Voice-over-IP (VoIP) traffic receives service with an assured minimum of delay, whereas e-commerce traffic might receive a minimum bandwidth guarantee (but not a delay guarantee).

For more information about QoS, see the *Cisco IOS Quality of Service Solutions Configuration Guide* , Release 12.2 and the *Cisco IOS Quality of Service Solutions Command Reference* , Release 12.2.

Services Supported by MPLS QoS

MPLS QoS supports the following services:

- Class-based weighted fair queuing (CBWFQ)--Provides queuing based on defined classes, with no strict priority queue available for real-time traffic. Weighted fair queuing allows you to define traffic classes based on match criteria. Once a class has been defined, you can assign characteristics to the class. For example, you can designate the minimum bandwidth delivered to the class during congestion.
- Low latency queuing (LLQ)--Provides strict priority queuing, which allows delay-sensitive data such as voice to be processed and sent first, before packets in other queues are processed. This provides preferential treatment to delay-sensitive data over other traffic.
- Weighed fair queuing (WFQ)--An automated scheduling system that uses a queuing algorithm to ensure fair bandwidth allocation to all network traffic. Weighted fair queuing is based on a relative bandwidth applied to each of the queues.
- Weighted random early detection (WRED)--RED is a congestion avoidance mechanism that controls the average queue size by indicating to the end hosts when they should temporarily stop sending packets. A small percentage of packets is dropped when congestion is detected and before the queue in question overflows completely.

The weighted aspect of WRED ensures that high-precedence traffic has lower loss rates than other traffic during congestion. WRED can be configured to discard packets that have certain markings. When a packet comes into a router, it is assigned an internal variable that is called a discard class. If desired, you can set the discard class at the input interface. At the output interface, the router can be configured to use the discard class for WRED instead of the MPLS EXP field.

Service Level Agreements Used in MPLS Tunneling

The service provider has an SLA with each customer. Each customer can have a different SLA. For example, the SLA for customer C1 may allow 256 kilobits of bandwidth for TCP packets (such as FTP packets or Telnet packets) and 1 megabyte of voice traffic per second. If the customer transmits 1 megabyte of voice traffic per second, the service provider delivers it to the other side of the customer's network. If the customer transmits more, the excess traffic is considered out-of-rate traffic and may or may not be discarded.

If the service provider experiences congestion, the service provider decides how to handle those packets. For example, the service provider may drop packets or give them less bandwidth. The PHB may be to drop a packet or to give it 20 percent of the link bandwidth.

The PHB that the service provider provides for a packet may be different from the PHB that the customer wants traffic to have in their network. The customer may be providing QoS at the output interface of each router in their network. However, the customer may be providing a different amount of bandwidth on those links than the service provider will provide. For example, a customer may give 50 percent of the link bandwidth to voice. The service provider may want to give only 10 percent of the link bandwidth to voice.

Providing QoS to an IP Packet

In an IP packet, the QoS that a router must provide has traditionally been designated in the IP Precedence field, which is the first three bits of the type of service (ToS) byte in the header of an IP packet. The IP Precedence and the differentiated services code point (DSCP) in an IP packet define the class. They may also designate the discard profile within a class. The DSCP is specified in the IETF standard for DiffServ. It is a new IETF standard for QoS.

Although some people still use the IP Precedence field, others use the DSCP to indicate the PHB that will be provided to an IP packet.

After label imposition, a configurable mapping function marks an equivalent PHB into the 3-bit MPLS EXP field value based on the IP Precedence or the IP DSCP marking.

Providing QoS to an MPLS Packet

In an MPLS packet, the PHB is marked in the MPLS EXP field within the MPLS label entry.

The EXP bits are similar in function to the IP Precedence and the DSCP in the IP network. The EXP bits generally carry all the information encoded in the IP Precedence or the DSCP.

The edge LSR that imposes the MPLS header sets the MPLS EXP field to a value.

DiffServ as a Standardization of QoS

DiffServ is a QoS architecture for IP networks. Packets within a DiffServ-enabled network may be classified into classes such as premium, gold, silver, or bronze based on QoS requirements. For example, VoIP packets may be grouped into the premium class, and e-commerce HTTP packets may be grouped into the gold class.

Each class has a marking associated with it. This makes packet classification extremely scalable and assures appropriate bandwidth and delay guarantees in the network. When packets enter the network, they are marked based on classification policies at the network boundary routers. The boundary routers also apply traffic

conditioning functions to control the amount of traffic entering the network. Traffic conditioning includes the following:

- Shaping--Smoothing the rate at which packets are sent into the network
- Policing--Dropping packets that exceed a subscribed-to-rate, or re-marking packets exceeding the rate so that the probability of dropping them increases when there is congestion

Each router within the network then applies different queuing and dropping policies on each packet based on the marking that the packet carries.

For more information about DiffServ, see the *Cisco IOS Switching Services Configuration Guide* , Release 2.2.

Tunneling Modes for MPLS DiffServ

Tunneling is the ability of QoS to be transparent from one edge of a network to the other edge of the network. A tunnel starts where there is label imposition. A tunnel ends where there is label disposition; that is, where the label is popped off of the stack and the packet goes out as an MPLS packet with a different PHB layer underneath or as an IP packet with the IP PHB layer.

There are three ways to forward packets through a network:

- Pipe mode with an explicit NULL LSP
- Short Pipe mode
- Uniform mode

Pipe mode and Short Pipe mode provide QoS transparency. With QoS transparency, the customer's IP marking in the IP packet is preserved.

Note The only difference between Pipe mode and Short Pipe mode is which PHB is used on the service provider's egress edge router. In Pipe mode with an explicit NULL LSP, QoS is done on the PE-to-CE link based on the service provider's PHB markings. The egress LSR still uses the marking that was used by intermediate LSRs.

All three tunneling modes affect the behavior of edge and penultimate label switching routers (LSRs) where labels are pushed (put onto packets) and popped (removed from packets). They do not affect label swapping at intermediate routers. A service provider can choose different types of tunneling modes for each customer.

Following is a brief description of each tunneling mode:

- Pipe mode with an explicit NULL LSP--QoS is done on the output interface of the PE router based on the received MPLS EXP field, even though one or more label entries have been popped. The IP Precedence field, EXP bits, and the DSCP field are not altered when they travel from the ingress to the egress of the MPLS network.

Any changes to the packet marking within the MPLS network are not permanent and do not get propagated when the packet leaves the MPLS network. The egress LSR still uses the marking that was used by intermediate provider core (P) routers. However, the egress provider edge (PE) router has to remove labels imposed on the original packet. To preserve the marking carried in the labels, the edge PE router keeps an internal copy of the marking before removing the labels. This internal copy is used to classify the packet on the outbound interface (facing the CE) after the labels are removed.

For a detailed description, see the Pipe Mode with an Explicit NULL LSP, on page 1799.

For the configuration procedure, see the Configuring Pipe Mode with an Explicit NULL LSP, on page 1807.

For an example, see the Pipe Mode with an Explicit NULL LSP Configuration Example, on page 1830.

- Short Pipe mode--In Short Pipe mode, the egress PE router uses the original packet marking instead of the marking used by the intermediate P routers.

For a detailed description, see the Short Pipe Mode, on page 1802.

For the configuration procedure, see the Configuring Short Pipe Mode, on page 1816.

For an example, see the Short Pipe Mode Configuration Example, on page 1832.

- Uniform mode--In Uniform mode, the marking in the IP packet may be manipulated to reflect the service provider's QoS marking in the core.

For a detailed description, see the Uniform Mode, on page 1805.

For the configuration procedure, see the Configuring Uniform Mode, on page 1821.

For an example, see the Uniform Mode Configuration Example, on page 1833.

MPLS PHB Layer Management

Through the network of routers, the MPLS EXP field can be marked differently and independently of the PHB marked in the IP Precedence or the DSCP field. A service provider can choose from existing classification criteria, including or excluding the IP PHB marking, to classify packets into a different PHB which is then marked only in the MPLS EXP field during label imposition.

Layer management is the ability to apply an additional layer of PHB marking to a packet. The PHB is the behavior of a packet at a router (that is, the unique discard and scheduling behavior that is applied to a packet). Layer management can occur at a service provider-managed CE router or at the service provider edge (PE) router.

If a packet arrives in a network as an IP packet, it may already have a PHB layer that is represented by a marking in the ToS byte. The marking can be IP Precedence bits or the DSCP.

If a packet arrives as an MPLS packet, it already has the following two PHB layers:

- IP layer
- MPLS layer, where the marking is in the MPLS EXP field of the topmost label entry

At a given hop, one PHB layer can be added to a packet. If only one label is being pushed onto the packet, the marking for the PHB layer being added is contained in only one label.

If two or more labels are being pushed onto a packet, the PHB layer being added is marked with the same MPLS EXP field in all of the label entries being pushed on at that time.

Tunneling Modes Operation

Note Cisco IOS allows a flexible configuration. You can configure the PHB definition of the MPLS EXP field differently from the PHB definition of the IP Precedence and DSCP.

A service provider may or may not care about the PHB marking of their customer's packet. For example, in customer C1's network, an IP Precedence value of 5 may mean voice. In customer C2's network, an IP Precedence value of 3 may mean voice. The service provider does not want to have two different IP Precedence values for voice. If the service provider has a large number of customers, there could be "many" values for voice. There are only eight possible EXP values.

To deal with different IP Precedence values representing the same PHB (in our example, for voice), the service provider does the following:

1. Arbitrarily chooses a common MPLS EXP field value to represent a PHB. For example, 2 can represent voice.
2. Looks at the packets of each customer. The service provider may look at the IP Precedence field value or at the UDP port number for voice, which is constant in every network.
3. For all customers, sets each voice packet to the MPLS EXP field value 2 on all the service provider's customer ports. Consequently, each router in the service provider's network only has to look for the MPLS EXP field value 2 for voice.

Another solution would be to set the DSCP value to 2, but that would alter the customer's PHB. MPLS DiffServ tunneling modes achieve the same results without altering the DSCP value.

This section illustrates and describes the following:

Pipe Mode with an Explicit NULL LSP

This section describes the following:

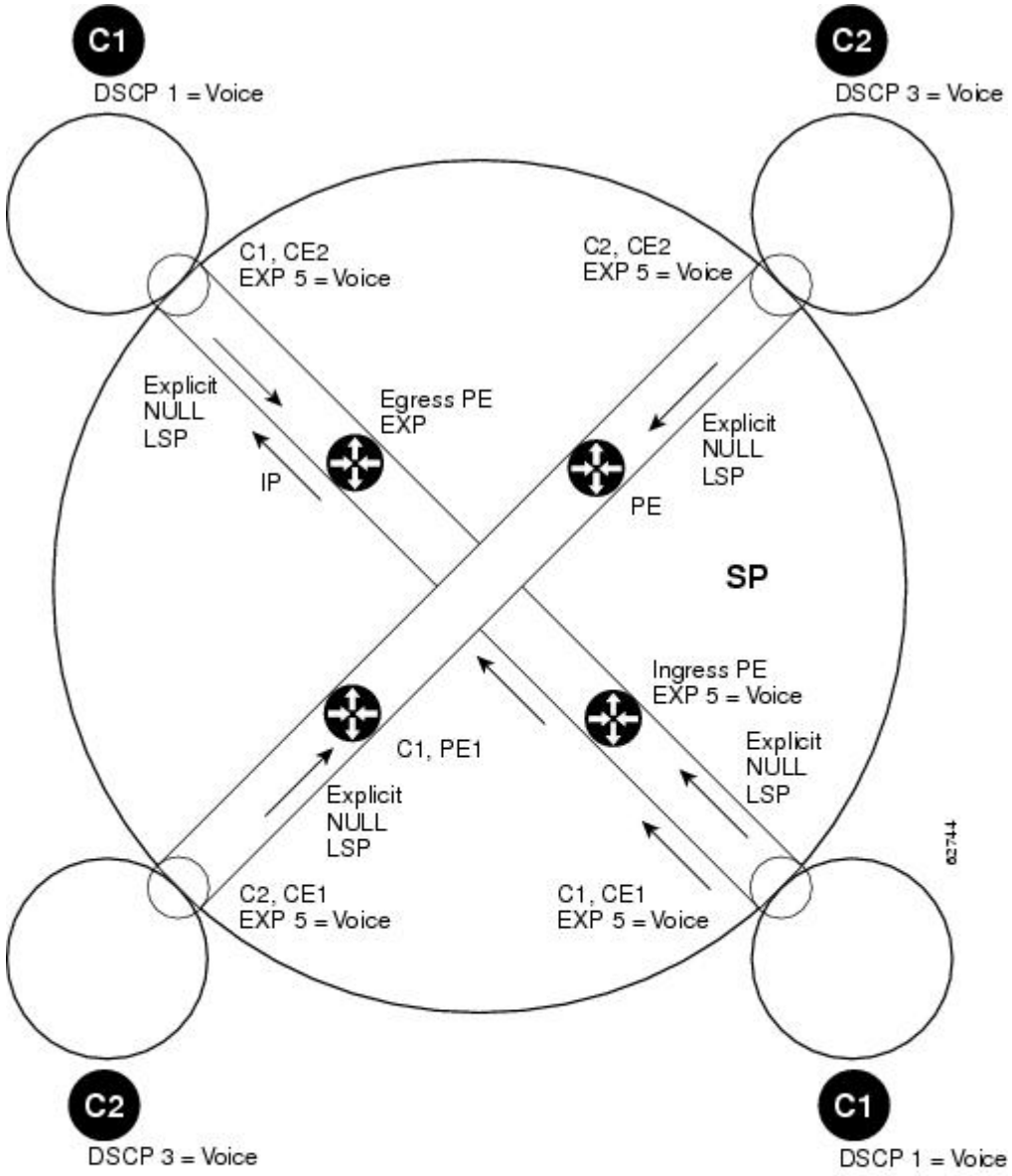
Pipe Mode with an Explicit NULL LSP Overview

Pipe mode with an explicit NULL LSP has the following characteristics:

- The QoS tunnel goes from the ingress CE router through the PE router to the egress CE router.
- There is an explicit NULL LSP from the CE router to the PE router. The label entry contains an MPLS EXP field, but does not carry a label value for forwarding purposes. It contains a zero (a null label value) for all packets going to the ingress PE router.
- The egress PE router removes the label entry and forwards packets as IP, but QoS is done on the output interface based on the MPLS EXP field received by the egress PE router.
- The service provider does not overwrite the IP Precedence value in the service provider's network.

The figure below shows an overview of Pipe mode with an explicit NULL LSP.

Figure 141: Pipe Mode with an Explicit NULL LSP Overview



Symbol	Meaning
C1	Customer 1's DiffServ domain
C2	Customer 2's DiffServ domain
CE1	Customer edge router 1
PE1	Service provider edge router (ingress LSR)
P1	Service provider router within the core of the provider's network
P2	Service provider router within the core of the provider's network
PE2	Service provider's edge router (egress LSR)
CE2	Customer edge router 2
SP	Service provider DiffServ domain

Note PE1 and PE2 are at the boundaries between the MPLS network and the IP network.

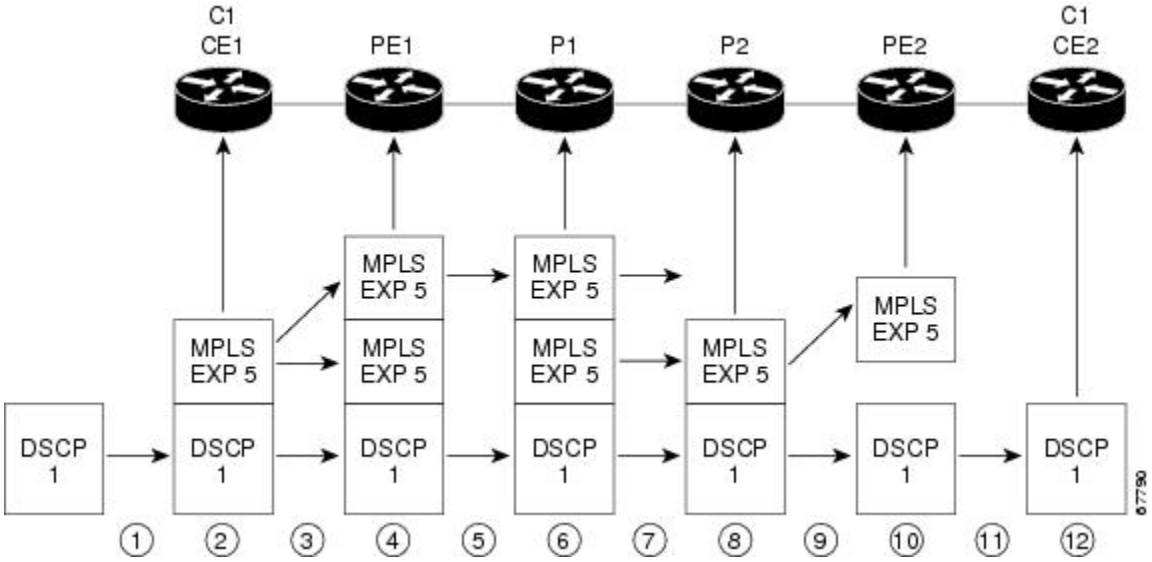
The figure above illustrates the following:

1. An IP packet arrives at C1, CE1 with a DSCP value of 1.
2. C2, CE1 sets the MPLS EXP field value to 5 during label imposition of the null label.
3. The packet goes through the service provider's network with the MPLS EXP field value set to 5.
4. Each router in the service provider's network looks at the MPLS EXP field and does QoS based on that value.
5. When the packet gets to the egress PE router going back into C1's network, it does QoS based on the packet's MPLS EXP field even though the packet is transmitted as an IP packet.

Pipe Mode with an Explicit NULL LSP Operating Procedure

The figure below illustrates the operation of Pipe mode with an explicit NULL LSP for Customer 1, when MPLS VPN is enabled. Since VPN is enabled, there are two MPLS label entries. Otherwise, there would be only one entry. The functionality would be similar for Customer 2, but the DSCP value would be 3.

Figure 142: Pipe Mode with an Explicit NULL LSP Operation with MPLS VPN Enabled



Pipe mode with an explicit NULL LSP functions as follows. The circled numbers at the bottom of the illustration correspond to the step numbers.

1. IP packets arrive at the router CE1, the managed CE router, with a DSCP value of 1.
2. An explicit NULL label entry is imposed onto the packet that contains an EXP value of 5.
3. The packet is transmitted to PE1 on the explicit NULL LSP.

4. The PE1 router saves the value of the MPLS EXP field and removes the explicit NULL entry. The PE1 router then imposes new labels onto the IP packet. Each label entry is set to the saved MPLS EXP field 5.
5. The packet is transmitted to P1.
6. At P1, the received EXP value is copied into the swapped label entry.
7. The packet is transmitted to P2.
8. At P2, the topmost label is popped, exposing a label entry that also has an EXP value of 5.
9. The packet is transmitted to PE2.
10. PE2 stores the value of the MPLS EXP field in the qos-group and discard-class variables, and removes the label entry from the packet.
11. While transmitting the packet to CE2, PE2 does QoS on its egress interface based on the saved value of the MPLS EXP field (qos-group and discard-class).
12. The IP packet arrives at the CE2 router.

Short Pipe Mode

This section describes the following:

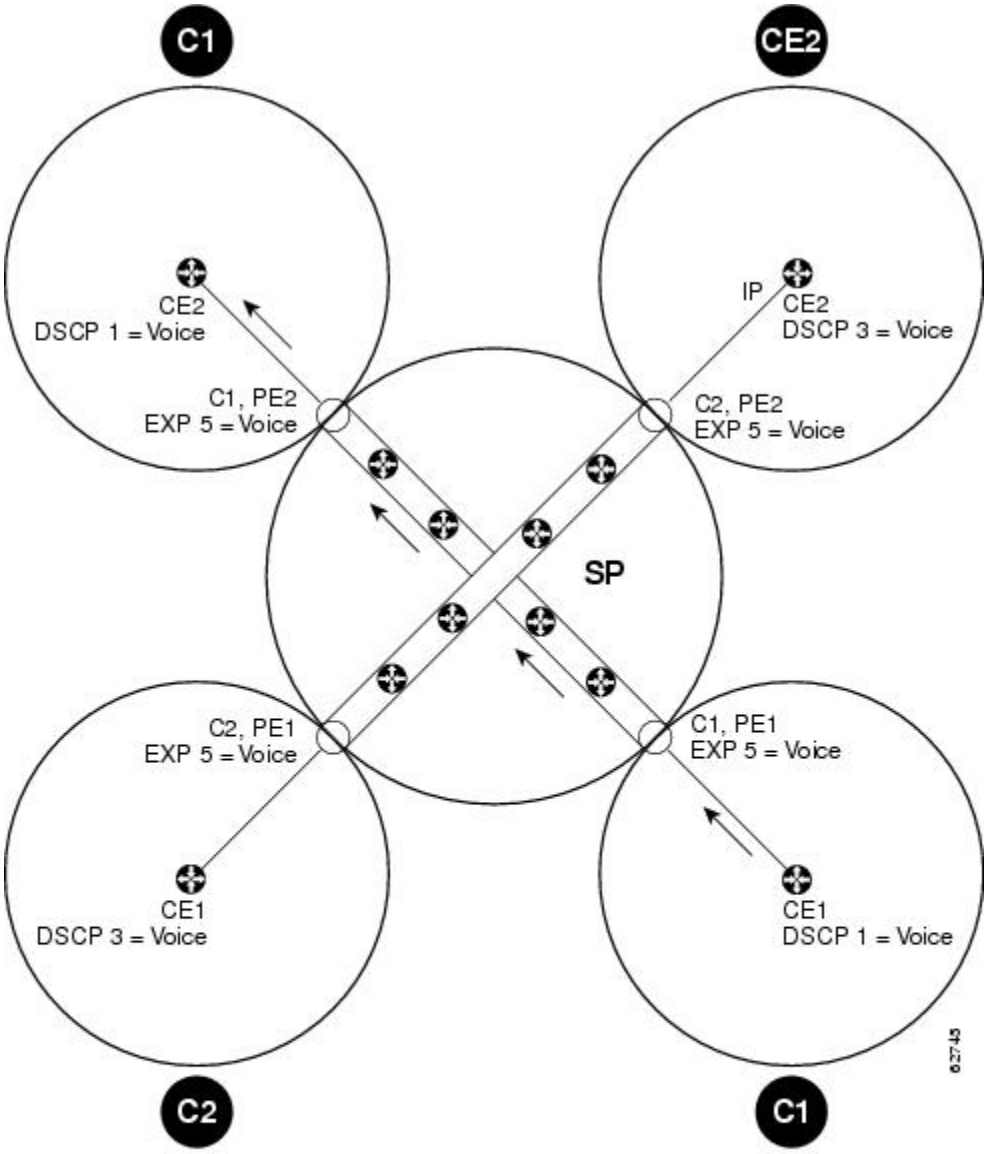
Short Pipe Mode Overview

Short Pipe mode has the following characteristics:

- The QoS tunnel goes from the ingress PE router to the egress PE router.
- The egress PE router transmits packets as IP and QoS is done on the output interface based on the IP DSCP or IP Precedence value.
- The service provider does not overwrite the DSCP or IP Precedence value in the service provider's network.

The figure below shows an overview of Short Pipe mode.

Figure 143: Short Pipe Mode Overview



Symbol	Meaning
C1	Customer 1's DiffServ domain
C2	Customer 2's DiffServ domain
CE1	Customer edge router 1
PE1	Service provider edge router (ingress LSR)
P1	Service provider router within the core of the provider's network
P2	Service provider router within the core of the provider's network
PE2	Service provider's edge router (egress LSR)
CE2	Customer edge router 2
SP	Service provider DiffServ domain

Note PE1 and PE2 are at the boundaries between the MPLS network and the IP network.

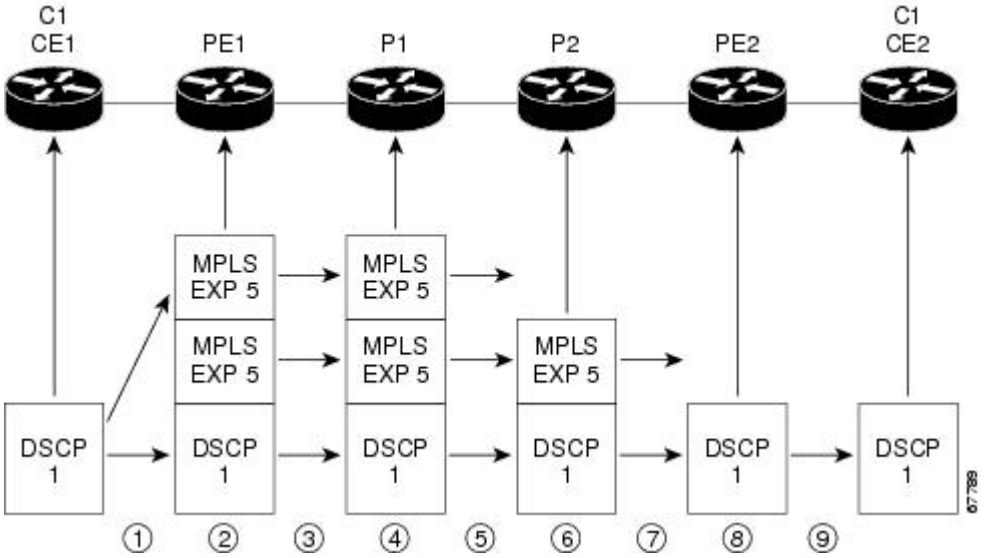
The figure above shows the following:

1. An IP packet arrives at C1, CE1 with a DSCP value of 1.
2. C1, CE1 transmits the IP packet to C1, PE1.
3. C1, PE1 sets the MPLS EXP field value to 5 during label imposition of the VPN label entries.
4. The packet goes through the service provider's network with the MPLS EXP field value set to 5.
5. Each router in the service provider's network looks at the MPLS EXP field and does QoS based on that value.
6. When the packet gets to the egress PE router going back into C1's network, it does QoS based on the IP DSCP field.

Short Pipe Mode Operating Procedure

The figure below illustrates Short Pipe mode.

Figure 144: Short Pipe Mode Operation



Short Pipe mode functions as follows. The circled numbers at the bottom of the illustration correspond to the step numbers.

1. C1, CE1 transmits an IP packet to PE1 with an IP DSCP value of 1.
2. PE1 sets the MPLS EXP field to 5 in the imposed label entries.
3. PE1 transmits the packet to P1.
4. P1 sets the MPLS EXP field value to 5 in the swapped label entry.
5. P1 transmits the packet to P2.
6. P2 pops the IGP label entry.
7. P2 transmits the packet to PE2.

8. PE2 pops the BGP label.
9. PE2 transmits the packet to C1, CE2, but does QoS based on the IP DSCP value.

Uniform Mode

This section describes the following:

Uniform Mode Overview

In a label, the MPLS EXP field is not the same as the label value.

The topmost label entry contains the following:

- Label value, which contains labels and other information, to forward the packet.
- MPLS EXP field, which only pertains to the QoS of the packet, not the route. The EXP field value is not advertised. Its value comes from the way that the packet is received.

In Uniform mode, packets are treated uniformly in the IP and MPLS networks; that is, the IP Precedence value and the MPLS EXP bits always are identical. Whenever a router changes or recolors the PHB of a packet, that change must be propagated to all encapsulation markings. The propagation is performed by a router only when a PHB is added or exposed due to label imposition or disposition on any router in the packet's path. The color must be reflected everywhere, at all levels. For example, if a packet's QoS marking is changed in the MPLS network, the IP QoS marking reflects that change.

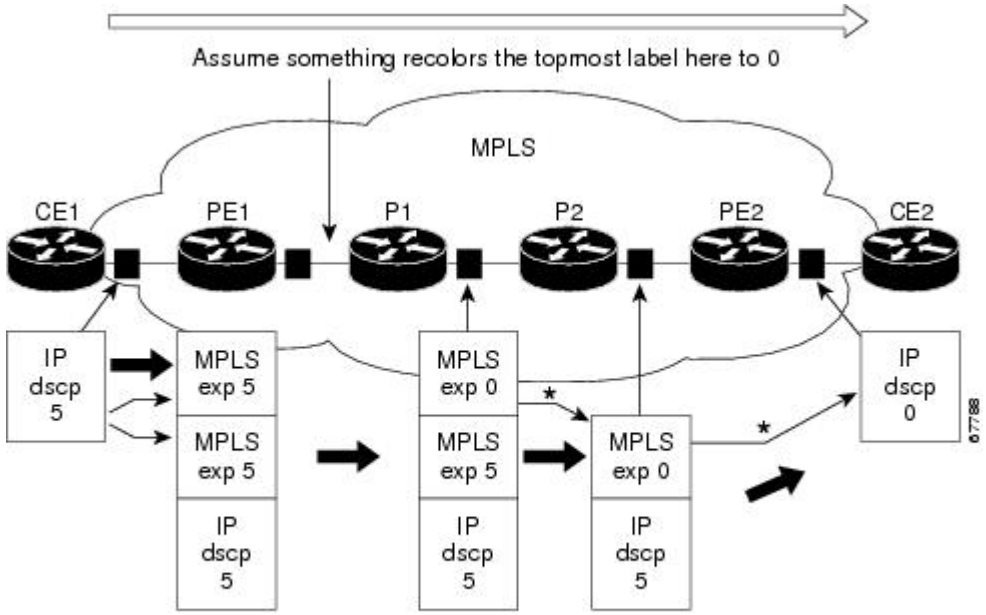
Uniform mode functions as follows:

- In both the MPLS-to-MPLS path and the MPLS-to-IP path, the PHBs of the topmost popped label are copied into the new top label or into the IP DSCP if no label remains.
- There can be a maximum of eight PHBs.
- If the PHBs are enclosed using more than the three Precedence bits, you must map DSCP to MPLS at the entry to the MPLS cloud.
- When packets leave the MPLS cloud, you must remap from the MPLS EXP value to the DSCP field in the IP header.

Uniform Mode Operating Procedure

The figure below illustrates the operation of Uniform mode.

Figure 145: Uniform Mode Operation



*In both the MPLS-to-MPLS and the MPLS-to-IP cases, the PHBs of the topmost popped label is copied into the new top label or the IP DSCP if no label remains

The procedure varies according to whether there are IP Precedence bit markings or DSCP markings.

The following actions occur if there are IP Precedence bit markings:

1. IP packets arrive in the MPLS network at PE1, the service provider edge router.
2. A label is copied onto the packet.
3. If the MPLS EXP field value is recolored (for example, if the packet becomes out-of-rate because too many packets are being transmitted), that value is copied to the IGP label. The value of the BGP label is not changed.
4. At the penultimate hop, the IGP label is removed. That value is copied into the next lower level label.
5. When all MPLS labels have been removed from the packet which is sent out as an IP packet, the IP Precedence or DSCP value is set to the last changed EXP value in the core.

Following is an example when there are IP precedence bit markings:

1. At CE1 (customer equipment 1), the IP packet has an IP Precedence value of 5.
2. When the packet arrives in the MPLS network at PE1 (the service provider edge router), the IP Precedence value of 5 is copied to the imposed label entries of the packet.
3. The MPLS EXP field in the IGP label header might be changed within the MPLS core (for example, at P1).

Note Since the IP Precedence bits are 5, the BGP label and the IGP label also contain 5 because in Uniform mode the labels always are identical. The packet is treated uniformly in the IP and MPLS networks.

1. At P2, when the IGP label is removed, the MPLS EXP field in this label entry is copied into the underlaying BGP label.
2. At PE2, when the BGP label is popped, the EXP field in this label header is copied into the IP Precedence field of the underlying IP header.

How to Configure MPLS DiffServ Tunneling Modes

Note You can configure only one of the tunneling modes.

Determining Which Tunneling Mode is Appropriate

- If there are managed customer edge (CE) routers, we recommend that you use Pipe mode with an explicit NULL LSP so that there is service provider PHB on the PE-to-CE link.
- If there is no managed CE router, we recommend that you use Short Pipe mode.
- If there are no markings or few markings, customers are likely to use Uniform mode.

Setting the MPLS EXP field

There are two ways to set the MPLS EXP field:

- Use the set mpls experimental topmost command to set the topmost label entry's value directly in the packet on the input and/or output interfaces.
- Use the set mpls experimental imposition command on the input interface to set the pushed label entry's value during label imposition.

By default, the label edge router copies the IP Precedence of the IP packet to the MPLS EXP field in all pushed label entries.

You can optionally map the IP Precedence or DSCP field to the MPLS EXP field in the MPLS header by using the **set mpls experimental imposition** command.

Configuring Pipe Mode with an Explicit NULL LSP

This section describes how to configure the following:

For examples, see the Pipe Mode with an Explicit NULL LSP Configuration Example, on page 1830.

Note The steps that follow show one way, but not the only way, to configure Pipe Mode with an Explicit NULL LSP.

Ingress CE Router--Customer Facing Interface

This procedure configures a policy map to set the MPLS EXP field in imposed label entries.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action***action*]
6. **interface** *type slot /port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map IP-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria for control plane traffic and other traffic that will be transmitted as IP.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-MPLS-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class IP-AF11`	Associates the traffic class with the service policy.
Step 5	**police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action***action*] **Example:** `Router(config-p-map-c)# police 8000 conform-action set-mpls-experimental-imposition-transmit 4 exceed-action set-mpls-experimental-imposition-transmit 2`	Configures the Traffic Policing feature, including the following: • Action to take on packets that conform to the rate limit specified in the SLA (service level agreement) • Action to take on packets that exceed the rate limit specified in the SLA At the action field, enter **set-mpls-experimental-imposition** *value*, where *value*is the value to which the MPLS EXP field will be set.

	Command or Action	Purpose
Step 6	**interface** *type slot /port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-MPLS-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Ingress CE Router--PE Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map match-any** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **match ip dscp** *dscp-values*
4. **policy-map** *name*
5. **class** *class-name*
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot/port*
9. **service-policy output** *name*
10. **mpls ip encapsulate explicit-null**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map match-any** *class-name* **Example:** `Router(config)# class-map match-any MPLS-AF1`	Specifies that packets must meet one of the match criteria to be considered a member of the traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Matches up to eight MPLS EXP field values. Puts matching packets into the same class.

	Command or Action	Purpose
Step 3	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria for control plane traffic and other traffic that will be transmitted as IP.
Step 4	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 5	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.
Step 10	**mpls ip encapsulate explicit-null** **Example:** `Router(config-if)# mpls ip encapsulate explicit-null`	Encapsulates with an explicit NULL label header all packets forwarded from the interface or subinterface.

Ingress PE Router--P Facing Interface

In this procedure, the default label swap behavior copies the received MPLS EXP field value to the output MPLS EXP field.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS values to use as match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.

	Command or Action	Purpose
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

P Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set mpls experimental topmost** *value*
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot/port*
9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.

	Command or Action	Purpose
Step 5	**set mpls experimental topmost** *value* **Example:** `Router(config-p-map-c)# set mpls experimental topmost 3`	Sets the MPLS EXP field value in the topmost MPLS label header at the input and/or output interfaces. This command is optional.
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Egress PE Router--P Facing Interface

In this procedure, the qos-group and discard-class convey a packet's PHB to the output interface. The qos-group and discard-class will be used for QoS classification and then will be discarded. The output IP packet's ToS field will not be overwritten.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set qos-group** *qos-group-value*
6. **set discard-class** *value*
7. **interface** *type slot/port*
8. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 4`	Specifies the packet characteristics that will be matched to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF11`	Associates the traffic class with the service policy.
Step 5	**set qos-group** *qos-group-value* **Example:** `Router(config-p-map-c)# set qos-group 1`	Sets a group ID that can be used later to classify packets. Valid values are from 0 to 99.
Step 6	**set discard-class** *value* **Example:** `Router(config-p-map-c)# set discard-class 1`	Marks a packet with a discard-class value. Specifies the type of traffic that will be dropped when there is congestion. Valid values are from 0 to 7.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Egress PE Router--Customer Facing Interface

This procedure classifies a packet according to the QoS group ID and determines a packet's discard treatment according to the discard-class attribute.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match qos-group** *qos-group-value*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect discard-class-based**
7. **interface** *type slot/port*
8. mpls ip
9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map Local-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match qos-group** *qos-group-value* **Example:** `Router(config-c-map)# match qos-group 1`	Identifies a specified QoS group value as a match criteria.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class Local-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect discard-class-based** **Example:** `Router(config-p-map-c)# random-detect discard-class-based`	Bases WRED on the discard class value of a packet.
Step 7	**interface** *type slot/port* **Example:**	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be

	Command or Action	Purpose
	`Router(config)# interface ethernet 3/0`	configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	mpls ip **Example:** `Router(config-if)# mpls ip`	Enables MPLS forwarding of IP version 4 (IPv4) packets along normally routed paths for a particular interface. **Note** You must issue the **mpls ip**command on this interface to receive packets with an explicit-NULL label from the CE router. The **mpls ip** command is not configured on the CE router's interface connected to this interface and therefore no LDP nor other label distribution protocol sessions will be established on this link.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Configuring Short Pipe Mode

This section describes how to configure the following:

For examples, see the Short Pipe Mode Configuration Example, on page 1832.

Note The steps that follow show one way, but not the only way, to configure Short Pipe mode.

Ingress PE Router--Customer Facing Interface

This procedure configures a policy map to set the MPLS EXP field in imposed label entries.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
6. **interface** *type slot/port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map IP-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-MPLS-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class IP-AF11`	Associates the traffic class with the service policy.
Step 5	**police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*] **Example:** `Router(config-p-map-c)# police 8000 conform-action set-mpls-experimental-imposition-transmit 4 exceed-action set-mpls-experimental-imposition-transmit 2`	Configures the Traffic Policing feature, including the following: • Action to take on packets that conform to the rate limit specified in the SLA. • Action to take on packets that exceed the rate limit specified in the SLA. At the action field, enter **set-mpls-experimental-imposition** *value*, where *value*is the value to which the MPLS EXP field will be set.
Step 6	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-MPLS-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Ingress PE Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.

	Command or Action	Purpose
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

P Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.

	Command or Action	Purpose
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Egress PE Router--Customer Facing Interface

This procedure classifies a packet based on its IP DSCP value and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect dscp-based**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:**	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.

	Command or Action	Purpose
	`Router(config)# class-map IP-AF1`	
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4 0`	Uses the DSCP values as the match criteria.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect dscp-based** **Example:** `Router(config-p-map-c)# random-detect dscp-based`	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Configuring Uniform Mode

This section describes how to configure the following:

For examples, see the Uniform Mode Configuration Example, on page 1833.

Note The steps that follow show one way, but not the only way, to configure Uniform mode.

Ingress PE Router--Customer Facing Interface

This procedure configures a policy map to set the MPLS EXP field in imposed label entries.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
6. **interface** *type slot/port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map IP-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-MPLS-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class IP-AF11`	Associates the traffic class with the service policy.
Step 5	**police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*] **Example:** `Router(config-p-map-c)# police 8000 conform-action set-mpls-experimental-imposition-transmit 3 exceed-action set-mpls-experimental-imposition-transmit 2`	Configures the Traffic Policing feature, including the following: • Action to take on packets that conform to the rate limit specified in the SLA. • Action to take on packets that exceed the rate limit specified in the SLA. At the action field, enter **set-mpls-experimental-imposition** *value*,where *value* is the value to which the MPLS EXP field will be set.

	Command or Action	Purpose
Step 6	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-MPLS-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Ingress PE Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 3`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.

	Command or Action	Purpose
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

P Router--Upstream P Facing Interface

This procedure classifies a packet based on the MPLS EXP field and sets the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set qos-group mpls experimental topmost**
6. **interface** *type slot/port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map mpls-in`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.

	Command or Action	Purpose
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 4 5`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map policy2`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class mpls-in`	Associates the traffic class with the service policy.
Step 5	**set qos-group mpls experimental topmost** **Example:** `Router(config-p-map-c)# set qos-group mpls experimental topmost`	Copies the MPLS EXP topmost field value into the QoS group ID. For more information, refer to *Enhanced Packet Marking* , Release 12.2(13)T.
Step 6	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input policy2`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

P Router--Downstream P Facing Interface

This procedure matches packets based on their QoS ID and sets the MPLS EXP field in the topmost label header to the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match qos-group** *qos-group-value*
3. **policy-map** *name*
4. **class** *class-name*
5. **set mpls experimental topmost qos-group**
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot/port*

9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map qos-group-out`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match qos-group** *qos-group-value* **Example:** `Router(config-c-map)# match qos-group 4`	Identifies a specified QoS group value as a match criterion.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map policy3`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class qos-group-out`	Associates the traffic class with the service policy.
Step 5	**set mpls experimental topmost qos-group** **Example:** `Router(config-p-map-c)# set mpls experimental topmost qos-group`	Copies the QoS group ID into the MPLS EXP field of the topmost label header.
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/1`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card numbers, and the backplane slot number.

	Command or Action	Purpose
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output policy3`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Egress PE Router--P Facing Interface

This procedure classifies a packet based on the MPLS EXP field and sets the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set qos-group mpls experimental topmost**
6. **interface** *type slot /port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map mpls-in`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 4 5`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map foo`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class mpls-in`	Associates the traffic class with the service policy.
Step 5	**set qos-group mpls experimental topmost** **Example:**	Copies the MPLS EXP topmost field value into the QoS group ID.

	Command or Action	Purpose
	`Router(config-p-map)# set qos-group mpls experimental topmost`	
Step 6	**interface** *type slot /port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card numbers, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input foo`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Egress PE Router--Customer Facing Interface

This procedure matches packets based on their QoS ID and sets the IP Precedence field to the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match qos-group** *qos-group-value*
3. **policy-map** *name*
4. **class** *class-name*
5. **set precedence qos-group**
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot /port*
9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map qos-out`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match qos-group** *qos-group-value* **Example:** `Router(config-c-map)# match qos-group 4`	Identifies a specified QoS group value as a match criterion.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map foo-out`	Configures the QoS policy for packets that match the class or classes.

	Command or Action	Purpose
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class qos-out`	Associates the traffic class with the service policy.
Step 5	**set precedence qos-group** **Example:** `Router(config-p-map-c)# set precedence qos-group`	Sets the Precedence value in the packet header.
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot /port* **Example:** `Router(config)# interface ethernet 3/1`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card numbers, and the backplane slot number.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output foo-out`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Verifying MPLS DiffServ Tunneling Mode Support

- On PE routers, the **show policy-map interface** command displays the discard-class-based WRED in the output.
- In Short Pipe mode, the **show policy-map interface** command looks for the **set mpls experimental imposition** command.

Troubleshooting Tips

- The IP QoS marking should not change in the service provider's network.
- QoS statistics should indicate that packets were scheduled in the correct classes.

Configuration Examples for MPLS DiffServ Tunneling Modes

Note You can configure only one tunneling mode.

- The examples that follow show one way, but not the only way, to configure the tunneling modes.

Pipe Mode with an Explicit NULL LSP Configuration Example

Ingress CE Router--Customer Facing Interface

In this example, packets are matched to class-map IP-AF11. The DSCP value 4 is used as the match criterion to determine whether a packet belongs to that class. Packets that are conforming have their MPLS EXP field set to 4. Packets that are out-of-rate have their MPLS EXP field set to 2.

```
class-map IP-AF11
 match ip dscp 4
policy-map set-MPLS-PHB
 class IP-AF11
  police 8000 conform-action set-mpls-experimental-imposition-transmit 4 exceed-action
  set-mpls-experimental-imposition-transmit 2
interface ethernet 3/0
 service-policy input set-MPLS-PHB
```

Ingress CE Router--PE Facing Interface

In this example, MPLS EXP 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map match-any MPLS-AF1
 match mpls experimental topmost 2 4
 match ip dscp 4
policy-map output-qos
class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
 mpls ip encapsulate explicit-null
```

Ingress PE Router--P Facing Interface

In this example, the default label swap behavior copies the received MPLS EXP field value to the output MPLS EXP field. Packets that have an MPLS EXP value of 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
   bandwidth percent 40
```

```
   random-detect
interface ethernet 3/0
 service-policy output output-qos
```

P Router--P Facing Interface

In this example, packets that have an MPLS EXP value of 2 or 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

Egress PE Router--P Facing Interface

In this example, qos-group 1 and discard-class 1 must be set to indicate the packet's PHB. The qos-group and discard-class are used for QoS classification at the output interface.

```
class-map MPLS-AF11
 match mpls experimental topmost 4
class-map MPLS-AF12
 match mpls experimental topmost 2
policy-map set-PHB
 class MPLS-AF11
  set qos-group 1
  set discard-class 1
class MPLS-AF12
  set qos-group 1
  set discard-class 2
interface ethernet 3/0
 service-policy input set-PHB
```

Egress PE Router--Customer Facing Interface

In this example, packets that have a qos-group value of 1 are matched to class-map Local-AF1. Packets that match that class have WRED based on their discard class value applied.

Note You must issue the **mpls ip** command on this interface to receive packets with an explicit-NULL label from the CE router. The **mpls ip** command is not configured on the CE router's interface connected to this interface and therefore no LDP nor other label distribution protocol sessions will be established on this link.

```
class-map Local-AF1
 match qos-group 1
policy-map output-qos
 class Local-AF1
  bandwidth percent 40
  random-detect discard-class-based
interface ethernet 3/0
 mpls ip
 service-policy output output-qos
```

Short Pipe Mode Configuration Example

Note Short Pipe mode is not configured on CE routers.

Ingress PE Router--Customer Facing Interface

In this example, IP packets are matched to class-map IP-AF11. Packets that are conforming have their MPLS EXP field set to 4. Packets that are out-of-rate have their MPLS EXP field set to 2.

```
class-map IP-AF11
 match ip dscp 4
policy-map set-MPLS-PHB
 class IP-AF11
  police 8000 conform-action set-mpls-experimental-imposition-transmit 4 exceed-action
  set-mpls-experimental-imposition-transmit 2
interface ethernet 3/0
 service-policy input set-MPLS-PHB
```

Ingress PE Router--P Facing Interface

In this example, MPLS EXP 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

P Router--P Facing Interface

In this example, MPLS EXP 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

Egress PE Router--Customer Facing Interface

In this example, the egress PE router transmits IP packets. Packets are matched to class-map IP-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map IP-AF1
 match ip dscp 4 0
```

```
policy-map output-qos
 class AF1
  bandwidth percent 40
  random-detect dscp-based
interface ethernet 3/0
 service-policy output output-qos
```

Uniform Mode Configuration Example

Ingress PE Router--Customer Facing Interface

In this example, IP packets are matched to class-map IP-AF11. Packets that are conforming have their MPLS EXP field set to 3. Packets that are out-of-rate have their MPLS EXP field set to 2.

```
class-map IP-AF11
 match ip dscp 4
policy-map set-MPLS-PHB
 class IP-AF11
  police 8000 conform-action set-mpls-experimental-imposition-transmit 3 exceed-action
  set-mpls-experimental-imposition-transmit 2
interface ethernet 3/0
 service-policy input set-MPLS-PHB
```

Ingress PE Router--P Facing Interface

In this example, MPLS EXP 2 and 3 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 3
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

P Router--Upstream P Facing Interface

At the penultimate P router's input interface where the IGP label is popped, the EXP field value in the IGP label is copied to the QoS group ID. Suppose the MPLS EXP field value in the IGP label was recolored in the core to 4 or 5. In this example, MPLS EXP values 4 and 5 are matched to class-map mpls-in. For packets that match that class, the MPLS EXP value in the IGP label is copied to the QoS group ID.

```
class-map mpls-in
 match mpls experimental topmost 4 5
policy-map policy2
 class mpls-in
  set qos-group mpls experimental topmost
interface ethernet 3/0
 service-policy input policy2
```

P Router--Downstream P Facing Interface

In this example, QoS group IDs 4 and 5 are matched to class-map qos-group-out. For packets that match that class, the MPLS EXP field in the topmost outgoing label is set to the QoS group ID.

```
class-map qos-group-out
 match qos-group 4
 match qos-group 5
policy-map policy3
 class qos-group-out
  set mpls experimental topmost qos-group
  bandwidth percent 40
  random-detect
interface ethernet 3/1
 service-policy output policy3
```

Egress PE Router--P Facing Interface

In this example, packets with MPLS EXP values 4 or 5 are matched to class-map mpls-in. The EXP field value from the label header is copied to the QoS group ID.

```
class-map mpls-in
 match mpls experimental topmost 4 5
policy-map foo
 class mpls-in
  set qos-group mpls experimental topmost
interface ethernet 3/0
 service-policy input foo
```

Egress PE Router--Customer Facing Interface

In this example, the egress PE router transmits IP packets. QoS group IDs 4 and 5 are matched into class-map qos-out and the IP Precedence field of those packets is set to the QoS group ID.

```
class-map qos-out
 match qos-group 4
 match qos-group 5
policy-map foo-out
 class qos-out
  set precedence qos-group
  bandwidth percent 40
  random-detect
interface ethernet 3/1
 service-policy output foo-out
```

Additional References

Related Documents

Related Topic	Document Title
MPLS Traffic Engineering	MPL S Configuration Guide
QoS	Quality of Service Configuration Guide

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS DiffServ Tunneling Modes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 162: Feature Information for MPLS DiffServ Tunneling Modes

Feature Name	Releases	Feature Information
MPLS DiffServ Tunneling Modes	12.2(13)T 12.2(18)S 12.2(27)SBA 12.2(27)SBB 12.2(28)SB 12.3(2)T Cisco IOS Release IOS XE 2.1 15.4(1)S	The MPLS DiffServ Tunneling Modes feature allows service providers to manage the QoS that a router will provide to an MPLS packet in an MPLS network. In 12.2(13)T, this feature was introduced. In 12.2(18)S, this feature was integrated. In 12.2(27)SBA, this feature was integrated. In 12.2(27)SBB, this feature was integrated. In 12.2(28)SB, this feature was integrated. In 12.3(2)T, this feature was integrated. In Cisco IOS Release IOS XE 2.1, this feature was integrated. In Cisco IOS Release 15.4(1) S, support was added for the Cisco 901S platform.

Glossary

CE router --customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

class --Classifies traffic, such as voice. You define a traffic class with the **class-map** command.

class-map --Defines what you want to match in a packet. For example, a class-map may specify voice packets.

core --The MPLS network. At the edges, there are edge routers.

customer network --A network that is under the control of an end customer. A customer network can use private addresses as defined in RFC 1918. Customer networks are logically isolated from each other and from the service provider's network.

DiffServ --Application-level QoS and traffic management in an architecture that incorporates mechanisms to control bandwidth, delay, jitter, and packet loss. Application traffic can be categorized into multiple classes (aggregates), with QoS parameters defined for each class. A typical arrangement would be to categorize traffic into premium, gold, silver, bronze, and best-effort classes.

DSCP --differentiated services code point, or DiffServ code point. A marker in the header of each IP packet that prompts network routers to apply differentiated grades of service to various packet streams. The value in the IP header indicates which PHB is to be applied to the packet.

discard-class --Local variable used to indicate the discard profile.

E-LSP --An LSP in which the QoS of a packet is determined solely by the MPLS EXP field in the MPLS header. E-LSPs are not supported by ATM-LSRs.

edge router --A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

egress router --Router at the edge of the network where packets are leaving.

encapsulation --The wrapping of data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit.

explicit null label --A label that just has an EXP value. A value of zero (0) represents the explicit NULL label. This label can only be at the bottom of the label stack. It indicates that the label stack must be popped, and the forwarding of the packet must then be based on the IPv4 header. Sometimes there may be requirements to have a label in the stack when no label is required. If you want to retain the MPLS EXP field to the next hop, you use an explicit null.

ingress router --Router at the edge of the network where packets are being received by the network.

IP Precedence field --The first three bits in the header of IP packets. These bits allow you to specify the QoS for an IP packet.

L-LSP --An LSP where a particular mechanism of implementing QoS using DiffServ is used. An LSP in which routers infer the QoS treatment for MPLS packets from the packet label and the EXP bits (or the CLP bit for cell-mode MPLS). The label is used to encode the class to which a packet belongs and the MPLS EXP field (or the CLP bit for cell-mode MPLS) is used to encode the drop precedence of the packet.

LSR --A router that is part of the MPLS network. An LSR forwards a packet based on the value of a label encapsulated in the packet.

label --A short, fixed-length label that tells switching nodes how to forward data (packets). MPLS associates a label with each route. A label associates a network address with the output interface onto which the packet should be transmitted. In the MPLS network, the next-hop IGP (Interior Gateway Protocol) router always advertises to the preceding IGP router (the upstream router) what label should be placed on the packets. The next-hop BGP (Border Gateway Protocol) router always advertises to the preceding BGP router what label should be placed on the packets.

label disposition --The act of removing the last MPLS label from a packet.

label entry --A label entry contains a label value (which includes labels and other information for forwarding the packet) and an MPLS EXP field (which pertains to the QoS of the packet). When there are two label entries, the top label entry is the IGP (Interior Gateway Protocol) label. The bottom label entry is the BGP (Border Gateway Protocol) label.

label imposition --The act of putting MPLS labels onto a packet for transmission on a label switched path (LSP).

layer management --Ability to apply an additional layer of PHB marking to a packet.

MPLS --Multiprotocol Label Switching. Emerging industry standard upon which label switching is based.

MPLS EXP field --In an MPLS entry, the per-hop behavior (PHB) is marked in the MPLS EXP field within the MPLS label entry.

P router --provider core router.

PE router --provider edge router. A router, at the edge of a service provider's network, that interfaces to CE routers.

penultimate hop popping --Removing a label at the penultimate router. A label is removed and copied to the label that is one lower.

penultimate router --The second-to-last router; that is, the router that is immediately before the egress router.

PHB --per-hop behavior. A unique discard and scheduling behavior that is applied to a packet. The DiffServ treatment (scheduling/dropping) applied by a router to all the packets that are to experience the same DiffServ service.

policing --Limiting the input or output transmission rate of a class of traffic based on user-defined criteria. Policing marks packets by setting the IP precedence value, the qos-group, or the DSCP value.

policy map --Action that is taken if a packet matches what was specified in the class-map. For example, if voice packets were identified and the class-map and voice packets are received, the specified policy map action is taken.

pop --The act of removing a label entry from a packet.

provider network --A backbone network that is under the control of a service provider, and provides transport between customer sites.

push --To put a label entry onto a packet.

QoS --quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

QoS transparency --Method of forwarding packets through a network where the customer's IP marking in the IP packet is preserved.

qos-group --Local variable that indicates the PHB scheduling class (PSC).

rate limiting --See *policing* .

recolor --To change the PHB marking on a packet.

swap --To replace a label entry on a packet.

ToS --type of service. Byte in the IPv4 header.

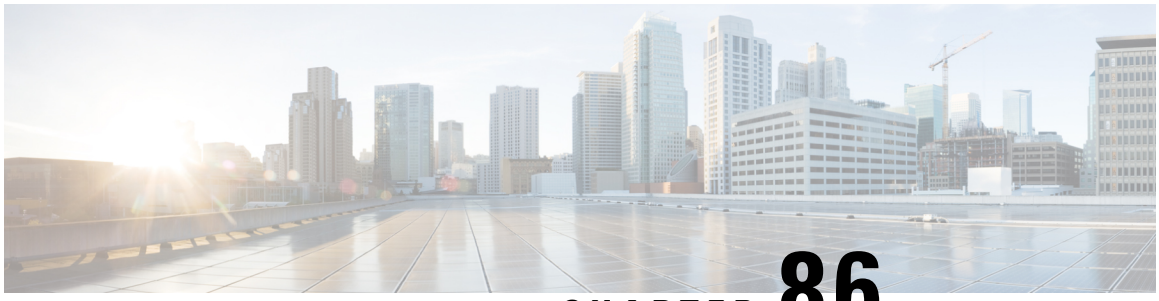
traffic policy --A traffic policy consists of a traffic class and one or more QoS features. You create a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).

transparency --Preservation of the customer's IP marking in the IP packet.

tunneling --The ability of QoS to be transparent from one edge of a network to the other edge of the network.

VPN --Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

WRED --weighted random early detection. A queuing method that ensures that high-precedence traffic has lower loss rates than other traffic during times of congestion.



CHAPTER 86

MPLS Traffic Engineering and Enhancements

Multiprotocol Label Switching (MPLS) traffic engineering software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what previously could be achieved only by overlaying a Layer 3 network on a Layer 2 network.

Prerequisites for MPLS Traffic Engineering and Enhancements

Your network must support the following Cisco IOS XE features before you enable MPLS traffic engineering:

- Multiprotocol Label Switching
- IP Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)

Restrictions for MPLS Traffic Engineering and Enhancements

- MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.
- MPLS traffic engineering does not support ATM MPLS-controlled subinterfaces.
- The MPLS traffic engineering feature does not support routing and signaling of LSPs over unnumbered IP links. Therefore, do not configure the feature over those links.

Information About MPLS Traffic Engineering and Enhancements

Introduction to MPLS Traffic Engineering and Enhancements

Multiprotocol Label Switching (MPLS) traffic engineering software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what now can be achieved only by overlaying a Layer 3 network on a Layer 2 network.

Traffic engineering is essential for service provider and Internet service provider (ISP) backbones. Such backbones must support a high use of transmission capacity, and the networks must be very resilient so that they can withstand link or node failures.

MPLS traffic engineering provides an integrated approach to traffic engineering. With MPLS, traffic engineering capabilities are integrated into Layer 3, which optimizes the routing of IP traffic, given the constraints imposed by backbone capacity and topology.

MPLS traffic engineering supports the following functionality:

- Enhances standard Interior Gateway Protocols (IGPs), such as IS-IS or OSPF, to automatically map packets onto the appropriate traffic flows.
- Transports traffic flows across a network using MPLS forwarding.
- Determines the routes for traffic flows across a network based on the resources the traffic flow requires and the resources available in the network.
- Employs “constraint-based routing,” in which the path for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow. In MPLS traffic engineering, the traffic flow has bandwidth requirements, media requirements, a priority that is compared to the priority of other flows, and so forth.
- Recovers from link or node failures by adapting to the new constraints presented by the changed topology.
- Transports packets using MPLS forwarding crossing a multihop label switched path (LSP).
- Uses the routing and signaling capability of LSPs across a backbone topology that
  - Understands the backbone topology and available resources
  - Accounts for link bandwidth and for the size of the traffic flow when determining routes for LSPs across the backbone
  - Has a dynamic adaptation mechanism that enables the backbone to be resilient to failures, even if several primary paths are precalculated off-line
  - Includes enhancements to the IGP (IS-IS or OSPF) shortest path first (SPF) calculations to automatically calculate what traffic should be sent over what LSPs.

Benefits of MPLS Traffic Engineering

WAN connections are an expensive item in an ISP budget. Traffic engineering enables ISPs to route network traffic to offer the best service to their users in terms of throughput and delay. By making the service provider more efficient, traffic engineering reduces the cost of the network.

Currently, some ISPs base their services on an overlay model. In th e overlay model, transmission facilities are managed by Layer 2 switching. The routers see only a fully meshed virtual topology, making most destinations appear one hop away. If you use the explicit Layer 2 transit layer, you can precisely control how traffic uses available bandwidth. However, the overlay model has numerous disadvantages. MPLS traffic engineering achieves the traffic engineering benefits of the overlay model without running a separate network, and without needing a nonscalable, full mesh of router interconnects.

How MPLS Traffic Engineering Works

MPLS traffic engineering automatically establishes and maintains LSPs across the backbone by using RSVP. The path that an LSP uses is determined by the LSP resource requirements and network resources, such as bandwidth.

Available resources are flooded by means of extensions to a link-state based IGP.

Traffic engineering tunnels are calculated at the LSP head based on a fit between required and available resources (constraint-based routing). The IGP automatically routes the traffic onto these LSPs. Typically, a packet crossing the MPLS traffic engineering backbone travels on a single LSP that connects the ingress point to the egress point.

MPLS traffic engineering is built on the following Cisco IOS XE mechanisms:

- IP tunnel interfaces

From a Layer 2 standpoint, an MPLS tunnel interface represents the head of an LSP. It is configured with a set of resource requirements, such as bandwidth and media requirements, and priority.

From a Layer 3 standpoint, an LSP tunnel interface is the headend of a unidirectional virtual link to the tunnel destination.

- MPLS traffic engineering path calculation module

This calculation module operates at the LSP head. The module determines a path to use for an LSP. The path calculation uses a link-state database containing flooded topology and resource information.

- RSVP with traffic engineering extensions

RSVP operates at each LSP hop and is used to signal and maintain LSPs based on the calculated path.

- MPLS traffic engineering link management module

This module operates at each LSP hop, does link call admission on the RSVP signaling messages, and bookkeeping of topology and resource information to be flooded.

- Link-state IGP (IS-IS or OSPF--each with traffic engineering extensions)

These IGPs are used to globally flood topology and resource information from the link management module.

- Enhancements to the SPF calculation used by the link-state IGP (IS-IS or OSPF)

The IGP automatically routes traffic onto the appropriate LSP tunnel based on tunnel destination. Static routes can also be used to direct traffic onto LSP tunnels.

- Label switching forwarding

This forwarding mechanism provides routers with a Layer 2-like ability to direct traffic across multiple hops of the LSP established by RSVP signaling.

One approach to engineering a backbone is to define a mesh of tunnels from every ingress device to every egress device. The MPLS traffic engineering path calculation and signaling modules determine the path taken by the LSPs for these tunnels, subject to resource availability and the dynamic state of the network. The IGP, operating at an ingress device, determines which traffic should go to which egress device, and steers that traffic into the tunnel from ingress to egress.

A flow from an ingress device to an egress device might be so large that it cannot fit over a single link, so it cannot be carried by a single tunnel. In this case, multiple tunnels between a given ingress and egress can be configured, and the flow is load-shared among them.

Mapping Traffic into Tunnels

This section describes how traffic is mapped into tunnels; that is, how conventional hop-by-hop link-state routing protocols interact with MPLS traffic engineering capabilities. In particular, this section describes how the shortest path first (SPF) algorithm, sometimes called a Dijkstra algorithm, has been enhanced so that a link-state IGP can automatically forward traffic over tunnels that MPLS traffic engineering establishes.

Link-state protocols, like integrated IS-IS or OSPF, use an SPF algorithm to compute a shortest path tree from the headend node to all nodes in the network. Routing tables are derived from this shortest path tree. The routing tables contain ordered sets of destination and first-hop information. If a router does normal hop-by-hop routing, the first hop is over a physical interface attached to the router.

New traffic engineering algorithms calculate explicit routes to one or more nodes in the network. The originating router views these explicit routes as logical interfaces. In the context of this document, these explicit routes are represented by LSPs and referred to as traffic engineering tunnels (TE tunnels).

The following sections describe how link-state IGPs can use these shortcuts, and how they can install routes in the routing table that point to these TE tunnels. These tunnels use explicit routes, and the path taken by a TE tunnel is controlled by the router that is the headend of the tunnel. In the absence of errors, TE tunnels are guaranteed not to loop, but routers must agree on how to use the TE tunnels. Otherwise, traffic might loop through two or more tunnels. See the following sections:

Enhancement to the SPF Computation

During each step of the SPF computation, a router discovers the path to one node in the network.

- If that node is directly connected to the calculating router, the first-hop information is derived from the adjacency database.
- If the node is not directly connected to the calculating router, the node inherits the first-hop information from the parent(s) of that node. Each node has one or more parents, and each node is the parent of zero or more downstream nodes.

For traffic engineering purposes, each router maintains a list of all TE tunnels that originate at this headend router. For each of those TE tunnels, the router at the tailend is known to the head-end router.

During the SPF computation, the TENT (tentative) list stores paths that are possibly the best paths and the PATH list stores paths that are definitely the best paths. When it is determined that a path is the best possible path, the node is moved from TENT to PATH. PATH is thus the set of nodes for which the best path from the computing router has been found. Each PATH entry consists of ID, path cost, and forwarding direction.

The router must determine the first-hop information. There are several ways to do this:

- Examine the list of tailend routers directly reachable by a TE tunnel. If there is a TE tunnel to this node, use the TE tunnel as the first hop.
- If there is no TE tunnel and the node is directly connected, use the first-hop information from the adjacency database.
- If the node is not directly connected and is not directly reachable by a TE tunnel, copy the first-hop information from the parent node(s) to the new node.

As a result of this computation, traffic to nodes that are the tail end of TE tunnels flows over the TE tunnels. Traffic to nodes that are downstream of the tail-end nodes also flows over the TE tunnels. If there is more than one TE tunnel to different intermediate nodes on the path to destination node X, traffic flows over the TE tunnel whose tail-end node is closest to node X.

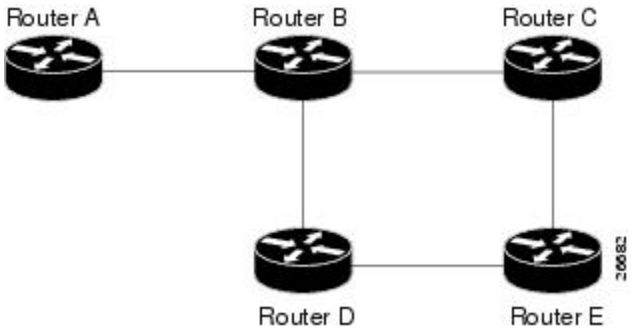
Special Cases and Exceptions for SPF Calculations

The SPF algorithm finds equal-cost parallel paths to destinations. The enhancement previously described does not change this. Traffic can be forwarded over any of the following:

- One or more native IP paths
- One or more traffic engineering tunnels
- A combination of native IP paths and traffic engineering tunnels

A special situation occurs in the topology shown in the figure below.

Figure 146: Sample Topology of Parallel Native Paths and Paths Over TE Tunnels



If parallel native IP paths and paths over TE tunnels are available, the following implementations allow you to force traffic to flow over TE tunnels only or only over native IP paths. Assume that all links have the same cost and that a TE tunnel is set up from Router A to Router D.

- When the SPF calculation puts Router C on the TENT list, it realizes that Router C is not directly connected. It uses the first-hop information from the parent, which is Router B.
- When the SPF calculation on Router A puts Router D on the TENT list, it realizes that Router D is the tail end of a TE tunnel. Thus Router A installs a route to Router D by the TE tunnel, and not by Router B.
- When Router A puts Router E on the TENT list, it realizes that Router E is not directly connected, and that Router E is not the tail end of a TE tunnel. Therefore Router A copies the first-hop information from the parents (Router C and Router D) to the first-hop information of Router E.

Traffic to Router E now load balances over

- The native IP path by Router A to Router B to Router C
- The TE tunnel Router A to Router D

Additional Enhancements to SPF Computation Using Configured Tunnel Metrics

When traffic engineering tunnels install an IGP route in a Router Information Base (RIB) as next hops, the distance or metric of the route must be calculated. Normally, you could make the metric the same as the IGP metric over native IP paths as if the TE tunnels did not exist. For example, Router A can reach Router C with the shortest distance of 20. X is a route advertised in IGP by Router C. Route X is installed in Router A's RIB with the metric of 20. When a TE tunnel from Router A to Router C comes up, by default the route is installed with a metric of 20, but the next-hop information for X is changed.

Although the same metric scheme can work well in other situations, for some applications it is useful to change the TE tunnel metric (for instance, when there are equal cost paths through TE tunnel and native IP links). You can adjust TE tunnel metrics to force the traffic to prefer the TE tunnel, to prefer the native IP paths, or to load share among them.

Suppose that multiple TE tunnels go to the same destination or different destinations. TE tunnel metrics can force the traffic to prefer some TE tunnels over others, regardless of IGP distances to those destinations.

Setting metrics on TE tunnels does not affect the basic SPF algorithm. It affects only two questions:

1. Is the TE tunnel installed as one of the next hops to the destination routers?
2. What is the metric value of the routes being installed into the RIB?

You can modify the metrics for determining the first-hop information in one of the following ways:

- If the metric of the TE tunnel to the tailend routers is higher than the metric for the other TE tunnels or native hop-by-hop IGP paths, this tunnel is not installed as the next hop.
- If the metric of the TE tunnel is equal to the metric of either other TE tunnels or native hop-by-hop IGP paths, this tunnel is added to the existing next hops.
- If the metric of the TE tunnel is lower than the metric of other TE tunnels or native hop-by-hop IGP paths, this tunnel replaces them as the only next hop.

In each of the above cases, the IGP assigns metrics to routes associated with those tailend routers and their downstream routers.

The SPF computation is loop free because the traffic through the TE tunnels is basically source routed. The end result of TE tunnel metric adjustment is the control of traffic loadsharing. If there is only one way to reach the destination through a single TE tunnel, then no matter what metric is assigned, the traffic has only one way to go.

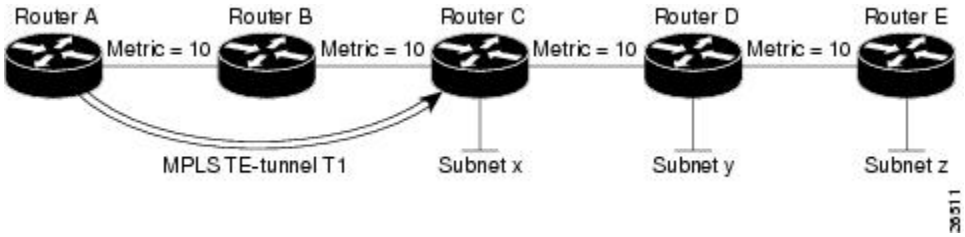
You can represent the TE tunnel metric in two different ways: (1) as an absolute (or fixed) metric or (2) as a relative (or floating) metric.

If you use an absolute metric, the routes assigned with the metric are fixed. This metric is used not only for the routes sourced on the TE tunnel tailend router, but also for each route downstream of this tailend router that uses this TE tunnel as one of its next hops.

For example, if you have TE tunnels to two core routers in a remote point of presence (POP), and one of them has an absolute metric of 1, all traffic going to that POP traverses this low-metric TE tunnel.

If you use a relative metric, the actual assigned metric value of routes is based on the IGP metric. This relative metric can be positive or negative, and is bounded by minimum and maximum allowed metric values. For example, assume the topology shown in the figure below.

Figure 147: Topology That Has No Traffic Engineering Tunnel



If there is no TE tunnel, Router A installs routes x, y, and z and assigns metrics 20, 30, and 40 respectively. Suppose that Router A has a TE tunnel T1 to Router C. If the relative metric -5 is used on tunnel T1, the routers x, y, and z have the installed metrics of 15, 25, and 35. If an absolute metric of 5 is used on tunnel T1, routes x, y and z have the same metric 5 installed in the RIB for Router A. The assigning of no metric on the TE tunnel is a special case, a relative metric scheme where the metric is 0.

Transition of an IS-IS Network to a New Technology

IS-IS, as specified in RFC 1142, includes extensions for MPLS traffic engineering and for other purposes. Running MPLS traffic engineering over IS-IS or taking advantage of these other extensions requires transitioning an IS-IS network to this new technology. This section describes these extensions and discusses two ways to migrate an existing IS-IS network from the standard ISO 10589 protocol towards the version of IS-IS specified in RFC 1142.Running MPLS traffic engineering over an existing IS-IS network requires a transition to the version of IS-IS specified in RFC 1142. However, running MPLS traffic engineering over OSPF does **not** require any similar network transition.

Extensions for the IS-IS Routing Protocol

Extensions for the IS-IS routing protocol serve the following purposes:

- Remove the 6-bit limit on link metrics.
- Allow interarea IP routes.
- Enable IS-IS to carry different kinds of information for traffic engineering. In the future, more extensions might be needed.

To serve these purposes, two new TLVs (type, length, and value objects) have been defined:

- TLV 22 describes links (or rather adjacencies). It serves the same purpose as the “IS neighbor option” in ISO 10589 (TLV 2).
- TLV 135 describes reachable IP prefixes. It is similar to the IP Neighbor options from RFC 1195 (TLVs 128 and 130).

Note For the purpose of briefness, these two new TLVs, 22 and 135, are referred to as “new-style TLVs.” TLVs 2, 128, and 130 are referred to as “old-style TLVs.”

Both new TLVs have a fixed length part, followed by optional sub-TLVs. The metric space in these new TLVs has been enhanced from 6 bits to 24 or 32 bits. The sub-TLVs allow you to add new properties to links and prefixes. Traffic engineering is the first technology to use this ability to add new properties to a link.

Problems with Old and New TLVs in Theory and in Practice

Link-state routing protocols compute loop-free routes. This is guaranteed because all routers calculate their routing tables based on the same information from the link-state database (LSPDB).

There is a problem when some routers look at old-style TLVs and some routers look at new-style TLVs because the routers can base their SPF calculations on different information. This can cause routing loops.

The easiest way to migrate from old-style TLVs towards new-style TLVs would be to introduce a “flag day.” A flag day means that you reconfigure all routers during a short period of time, during which service is interrupted. If the implementation of a flag day is not acceptable, a network administrator needs to find a viable solution for modern existing networks.

Network administrators have the following problems related to TLVs:

- They need to run an IS-IS network where some routers are advertising and using the new-style TLVs and, at the same time, other routers are capable only of advertising and using old-style TLVs.
- They need to test new traffic engineering software in existing networks on a limited number of routers. They cannot upgrade all their routers in their production networks or in their test networks before they start testing.

The new extensions allow a network administrator to use old-style TLVs in one area, and new-style TLVs in another area. However, this is not a solution for administrators who need or want to run their network in one single area.

The following sections describe two solutions to the network administrator’s problems.

First Solution for Transitioning an IS-IS Network to a New Technology

When you migrate from old-style TLVs towards new-style TLVs, you can advertise the same information twice--once in old-style TLVs and once in new-style TLVs. This ensures that all routers can understand what is advertised.

There are three disadvantages to using that approach:

- Size of the LSPs--During the transition, the LSPs grow to about twice their original size. This might be a problem in networks where the LSPDB is large. An LSPDB might be large because
  - There are many routers, and thus LSPs.
  - There are many neighbors or IP prefixes per router. A router that advertises lots of information causes the LSPs to be fragmented.
- Unpredictable results--In a large network, this solution can produce unpredictable results. A large network in transition pushes the limits regarding LSP flooding and SPF scaling. During the transition
  - You can expect some extra network instability. At this time, you especially do not want to test how far you can push an implementation.
  - Traffic engineering extensions might cause LSPs to be reflooded frequently.

- Ambiguity--If a router encounters different information in the old-style TLVs and the new-style TLVs, it may not be clear what the router should do.

These problems can be largely solved easily by using

- All information in old-style and new-style TLVs in an LSP
- The adjacency with the lowest link metric if an adjacency is advertised more than once

The main benefit to advertising the same information twice is that network administrators can use new-style TLVs before all routers in the network can understand them.

Transition Actions During the First Solution

When transitioning from using IS-IS with old-style TLVs to new-style TLVs, you can perform the following actions:

- If all routers run old software, advertise and use only old-style TLVs.
- Upgrade some routers to newer software.
- Configure some routers with new software to advertise both old-style and new-style TLVs. They accept both styles of TLVs. Configure other routers (with old software) to continue advertising and using only old-style TLVs.
- Test traffic engineering in parts of your network; however, new-style TLVs cannot be used yet.
- If the whole network needs to migrate, upgrade and configure all remaining routers to advertise and accept both styles of TLVs.
- Configure all routers to advertise and accept only new-style TLVs.
- Configure metrics larger than 63.

For more information about how to perform these actions, see the TLV Configuration Commands section.

Second Solution for Transitioning an IS-IS Network to a New Technology

Routers advertise only one style of TLVs at the same time, but can understand both types of TLVs during migration. There are two main benefits to this approach:

- LSPs stay approximately the same size during migration.
- There is no ambiguity when the same information is advertised twice inside one LSP.

This method is useful when you are transitioning the whole network (or a whole area) to use wider metrics (that is, you want a router running IS-IS to generate and accept only new-style TLVs). For more information, see the **metric-style wide**command.

The disadvantage is that all routers must understand the new-style TLVs before any router can start advertising new-style TLVs. It does not help the second problem, where network administrators want to use the new-style TLVs for traffic engineering, while some routers are capable of understanding only old-style TLVs.

Transition Actions During the Second Solution

If you use the second solution, you can perform the following actions:

- If all routers run old software, advertise and use only old-style TLVs.
- Upgrade all routers to newer software.
- Configure all routers one-by-one to advertise old-style TLVs, but to accept both styles of TLVs.
- Configure all routers one-by-one to advertise new-style TLVs, but to accept both styles of TLVs.
- Configure all routers one-by-one to advertise and to accept only new-style TLVs.
- Configure metrics larger than 63.

TLV Configuration Commands

Cisco IOS XE has a **router isis**command-line interface (CLI) command called **metric-style**. Once the router is in IS-IS configuration mode, you have the option to choose the following:

- **metric-style narrow** --Enables the router to generate and accept only old-style TLVs
- **metric-style transition** --Enables the router to generate and accept both old-style and new-style TLVs
- **metric-style wide** --Enables the router to generate and accept only new-style TLVs

You can use either of the following two transition schemes when you use the **metric-style**command to configure:

- Narrow to transition to wide
- Narrow to narrow transition to wide transition to wide

Implementation in Cisco IOS XE Software

Cisco IOS XE implements both transitions solution. Network administrators can choose the solution that suits them best. For test networks, the first solution is best. For a full transition, both solutions can be used. The first solution requires fewer steps and less configuration. You would use the second solution for the largest networks where a risk of doubling the LSPDB during transition exists.

How to Configure MPLS Traffic Engineering and Enhancements

Configuring a Device to Support Tunnels

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**

4. **mpls traffic-eng tunnels**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Router(config)# ip cef`	Enables standard Cisco Express Forwarding operation.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config)# mpls traffic-eng tunnels`	Enables the MPLS traffic engineering tunnel feature on a device.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring an Interface to Support RSVP-Based Tunnel Signaling and IGP Flooding

Note You must enable the tunnel feature on interfaces that you want to support MPLS traffic engineering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*]
4. **mpls traffic-eng tunnels**
5. **ip rsvp bandwidth** *bandwidth*
6. **exit**

7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*] **Example:** `Router(config)# interface serial 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config-if)# mpls traffic-eng tunnels`	Enables MPLS traffic engineering tunnels on an interface.
Step 5	**ip rsvp bandwidth** *bandwidth* **Example:** `Router(config-if)# ip rsvp bandwidth 1000`	Enables RSVP for IP on an interface and specifies the amount of bandwidth that will be reserved.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IS-IS for MPLS Traffic Engineering

To configure IS-IS for MPLS traffic engineering, perform the following steps.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. Router(config)# **router isis**
2. Router(config-router)# **mpls traffic-eng level-1**
3. Router(config-router)# **mpls traffic-eng level-2**
4. Router(config-router)# **mpls traffic-eng router-id loopback 0**
5. Router(config-router)# **metric-style wide**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router isis**	Enables IS-IS routing and specifies an IS-IS process for IP. The router is placed in configuration mode.
Step 2	Router(config-router)# **mpls traffic-eng level-1**	Turns on MPLS traffic engineering for IS-IS level 1.
Step 3	Router(config-router)# **mpls traffic-eng level-2**	Turns on MPLS traffic engineering for IS-IS level 2.
Step 4	Router(config-router)# **mpls traffic-eng router-id loopback 0**	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface loopback0.
Step 5	Router(config-router)# **metric-style wide**	Configures a router to generate and accept only new-style type, length, value objects (TLVs).

Configuring OSPF for MPLS Traffic Engineering

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **mpls traffic-eng area** *number*
5. **mpls traffic-eng router-id loopback0**
6. **exit**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 200`	Configures an OSPF routing process for IP and enters router configuration mode. • The *process-id* is an internally used identification parameter for an OSPF routing process. It is locally assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**mpls traffic-eng area** *number* **Example:** `Router(config-router)# mpls traffic-eng area 0`	Turns on MPLS traffic engineering for the indicated OSPF area.
Step 5	**mpls traffic-eng router-id loopback0** **Example:** `Router(config-router)# mpls traffic-eng router-id loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface loopback0.
Step 6	**exit** **Example:** `Router(config-router)# exit`	Exits to global configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel

This tunnel has two path setup options: a preferred explicit path and a backup dynamic path.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | **identifier** *path-number*}} [**lockdown**]
9. **exit**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface Tunnel0`	Configures an interface type and enters interface configuration mode. • The *number*argument is the number of the tunnel.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered loopback0`	Enables IP processing on an interface without assigning an explicit IP address to the interface. • The *type* and *number* arguments name the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface. • An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.4.4`	Specifies the destination for a tunnel interface. • The *ip-address* argument must be the MPLS traffic engineering router ID of the destination device.
Step 6	**tunnel mode mpls traffic-eng** **Example:**	Sets the tunnel encapsulation mode to MPLS traffic engineering.

	Command or Action	Purpose
	`Router(config-if)# tunnel mode mpls traffic-eng`	
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 250`	Configures the bandwidth for the MPLS traffic engineering tunnel. • The *bandwidth* argument is a number in kilobits per second that is set aside for the MPLS traffic engineering tunnel. Range is from 1 to 4294967295. **Note** If automatic bandwidth is configured for the tunnel, use the **tunnel mpls traffic-eng bandwidth** command to configure the initial tunnel bandwidth, which is adjusted by the autobandwidth mechanism.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | **identifier** *path-number*}} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 10 explicit identifier 321`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. • The *number* argument is the preference for this path option. When you configure multiple path options, lower numbered options are preferred. Valid values are from 1 to 1000. • The **dynamic** keyword indicates that the path of the label switched path (LSP) is dynamically calculated. • The **explicit** keyword indicates that the path of the LSP is an IP explicit path. • The **name** *path-name* keyword and argument are the path name of the IP explicit path that the tunnel uses with this option. • The **identifier** *path-number*keyword and argument pair names the path number of the IP explicit path that the tunnel uses with this option. The range is from 1 to 65535. • The **lockdown** keyword specifies that The LSP cannot be reoptimized. **Note** A dynamic path is used if an explicit path is currently unavailable.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

DEFAULT STEPS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name*} | **identifier** *path-number*} [**lockdown**]
9. **exit**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel10`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Gives the tunnel interface an IP address. • An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address*	Specifies the destination for a tunnel.

	Command or Action	Purpose
	Example: `Router(config-if)# tunnel destination 10.20.1.1`	• The *ip-address* keyword is the IP address of the host destination expressed in dotted decimal notation.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 1000`	Configures the bandwidth for the MPLS traffic engineering tunnel.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name*} | **identifier** *path-number*} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit identifier 1`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. • A dynamic path is used if an explicit path is currently unavailable.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel that an IGP Can Use

This tunnel has two path setup options: a preferred explicit path and a backup dynamic path.

DEFAULT STEPS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng autoroute announce**
5. **exit**
6. **exit**

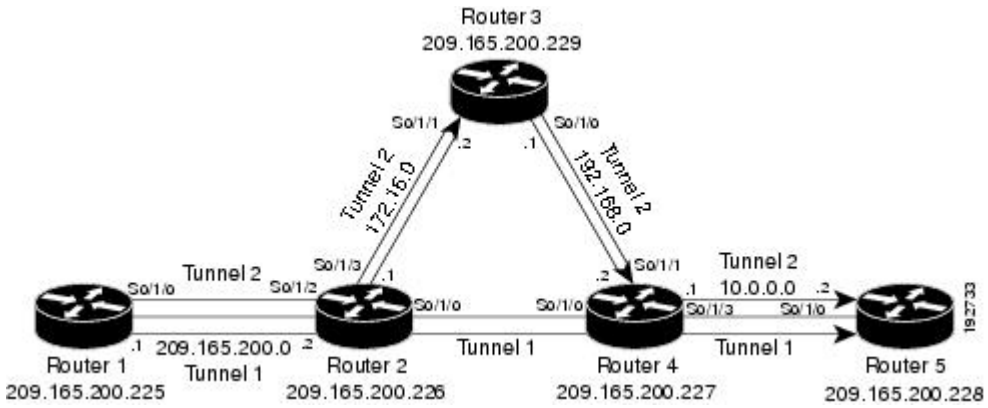
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel1`	Configures an interface type and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng autoroute announce** **Example:** `Router(config-if)# tunnel mpls traffic-eng autoroute announce`	Causes the IGP to use the tunnel in its enhanced SPF calculation.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for MPLS Traffic Engineering and Enhancements

The figure below illustrates a sample MPLS topology. This example specifies point-to-point outgoing interfaces. The next sections contain sample configuration commands you enter to implement MPLS traffic engineering and the basic tunnel configuration shown in Figure 3.

Figure 148: Sample MPLS Traffic Engineering Tunnel Configuration



Example Configuring MPLS Traffic Engineering Using IS-IS

This example lists the commands you enter to configure MPLS traffic engineering with IS-IS routing enabled (see the figure above).

Note You must enter the following commands on every router in the traffic-engineered portion of your network.

Router 1--MPLS Traffic Engineering Configuration

To configure MPLS traffic engineering, enter the following commands:

```
ip cef
mpls traffic-eng tunnels
interface loopback 0
ip address 10.0.0.0 255.255.255.254
ip router isis
interface s1/0/0
ip address 209.165.200.1 255.255.0.0
ip router isis
mpls traffic-eng tunnels
ip rsvp bandwidth 1000
```

Router 1--IS-IS Configuration

To enable IS-IS routing, enter the following commands:

```
router isis
network 47.0000.0011.0011.00
is-type level-1
metric-style wide
mpls traffic-eng router-id loopback0
mpls traffic-eng level-1
```

Example Configuring MPLS Traffic Engineering Using OSPF

This example lists the commands you enter to configure MPLS traffic engineering with OSPF routing enabled (see the figure above).

Note You must enter the following commands on every router in the traffic-engineered portion of your network.

Router 1--MPLS Traffic Engineering Configuration

To configure MPLS traffic engineering, enter the following commands:

```
ip cef
mpls traffic-eng tunnels
interface loopback 0
ip address 209.165.200.225 255.255.255.255
interface s1/0/0
ip address 209.165.200.1 255.255.0.0
mpls traffic-eng tunnels
  ip rsvp bandwidth 1000
```

Router 1--OSPF Configuration

To enable OSPF, enter the following commands:

```
router ospf 0
network 209.165.200.0.0.0.255.255 area 0
mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
```

Example Configuring an MPLS Traffic Engineering Tunnel

This example shows you how to configure a dynamic path tunnel and an explicit path in the tunnel. Before you configure MPLS traffic engineering tunnels, you must enter the appropriate global and interface commands on the specified router (in this case, Router 1).

Router 1--Dynamic Path Tunnel Configuration

In this section, a tunnel is configured to use a dynamic path.

```
interface tunnel1
  ip unnumbered loopback 0
  tunnel destination 209.165.200.228
  tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 100
  tunnel mpls traffic-eng priority 1 1
  tunnel mpls traffic-eng path-option 1 dynamic
```

Router 1--Dynamic Path Tunnel Verification

This section includes the commands you use to verify that the tunnel is up.

```
show mpls traffic-eng tunnels
show ip interface tunnel1
```

Router 1--Explicit Path Configuration

In this section, an explicit path is configured.

```
ip explicit-path identifier 1
 next-address 209.165.200.1
 next-address 172.16.0.1
 next-address 192.168.0.1
 next-address 10.0.0.1
```

Router 1--Explicit Path Tunnel Configuration

In this section, a tunnel is configured to use an explicit path.

```
interface tunnel2
  ip unnumbered loopback 0
  tunnel destination 209.165.200.228
  tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 100
  tunnel mpls traffic-eng priority 1 1
  tunnel mpls traffic-eng path-option 1 explicit identifier 1
```

Router 1--Explicit Path Tunnel Verification

This section includes the commands you use to verify that the tunnel is up.

```
show mpls traffic-eng tunnels
show ip interface tunnel2
```

Example Configuring Enhanced SPF Routing over a Tunnel

This section includes the commands that cause the tunnel to be considered by the IGP's enhanced SPF calculation, which installs routes over the tunnel for appropriate network prefixes.

Router 1--IGP Enhanced SPF Consideration Configuration

In this section, you specify that the IGP should use the tunnel (if the tunnel is up) in its enhanced shortest path first (SPF) calculation.

```
interface tunnel1
tunnel mpls traffic-eng autoroute announce
```

Router 1--Route and Traffic Verification

This section includes the commands you use to verify that the tunnel is up and that the traffic is routed through the tunnel.

```
show traffic-eng tunnels tunnel1 brief
show ip route 209.165.200.228
show mpls traffic-eng autoroute
ping 209.165.200.228
```

```
show interface tunnel1 accounting
show interface s1/0/0 accounting
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuring Integrated IS-IS	*Cisco IOS XE IP Routing Protocols Configuration Guide*
IS-IS commands	*Cisco IOS IP Routing Protocols Command Reference*
Configuring OSPF	*Cisco IOS XE IP Routing Protocols Configuration Guide*
OSPF command	*Cisco IOS IP Routing Protocols Command Reference*
Configuring Multiprotocol Label Switching	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*
MPLS TE commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
RSVP commands	*Cisco IOS Quality of Service Solutions Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
1142	*IS-IS*
1195	*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*
2205	*Resource ReSerVation Protocol (RSVP)*

RFC	Title
2328	*OSPF Version 2*
2370	*The OSPF Opaque LSA Option*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering and Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 163: Feature Information for MPLS Traffic Engineering and Enhancements

Feature Name	Releases	Feature Information
MPLS Traffic Engineering and Enhancements	Cisco IOS XE Release 2.3	Multiprotocol Label Switching (MPLS) traffic engineering software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what previously could be achieved only by overlaying a Layer 3 network on a Layer 2 network. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Feature Name	Releases	Feature Information
		The following commands were introduced or modified:**ip explicit-path**, **metric-style narrow**, **metric-style transition**, **metric-style wide**, **mpls traffic-eng**, **mpls traffic-eng area**, **mpls traffic-eng router-id**, **mpls traffic-eng tunnels**(configuration), **mpls traffic-eng tunnels**(interface), **show mpls traffic-eng autoroute**, **show mpls traffic-eng tunnels**, **tunnel mode mpls traffic-eng**, **tunnel mode mpls traffic-eng autoroute announce**, **tunnel mpls traffic-eng bandwidth**, **tunnel mpls traffic-eng path-option**, **tunnel mpls traffic-eng priority.**

Glossary

affinity --An MPLS traffic engineering tunnel's requirements on the attributes of the links it will cross. The tunnel's affinity bits and affinity mask bits must match the attribute bits of the various links carrying the tunnel.

call admission precedence --An MPLS traffic engineering tunnel with a higher priority will, if necessary, preempt an MPLS traffic engineering tunnel with a lower priority. Tunnels that are harder to route are expected to have a higher priority and to be able to preempt tunnels that are easier to route. The assumption is that lower-priority tunnels will be able to find another path.

constraint-based routing --Procedures and protocols that determine a route across a backbone take into account resource requirements and resource availability instead of simply using the shortest path.

flow --A traffic load entering the backbone at one point--point of presence (POP)--and leaving it from another, that must be traffic engineered across the backbone. The traffic load is carried across one or more LSP tunnels running from the entry POP to the exit POP.

headend --The upstream, transmit end of a tunnel.

IGP --Interior Gateway Protocol. The Internet protocol used to exchange routing information within an autonomous system. Examples of common IGPs include IGRP, OSPF, and RIP.

ip explicit path --A list of IP addresses, each representing a node or link in the explicit path.

IS-IS --Intermediate System-to-Intermediate System. OSI link-state hierarchical routing protocol that calls for intermediate system (IS) routers to exchange routing information based on a single metric to determine network topology.

label switched path (LSP) --A sequence of hops (R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A label switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

label switched path (LSP) tunnel --A configured connection between two routers, in which label switching is used to carry the packets.

label switching router (LSR) --A Layer 3 router that forwards packets based on the value of a label encapsulated in the packets.

LCAC --Link-level (per hop) call admission control.

LSA --Link-state advertisement. Flooded packet used by OSPF that contains information about neighbors and path costs. In IS-IS, receiving routers use LSAs to maintain their routing tables.

LSP--See label switched path.

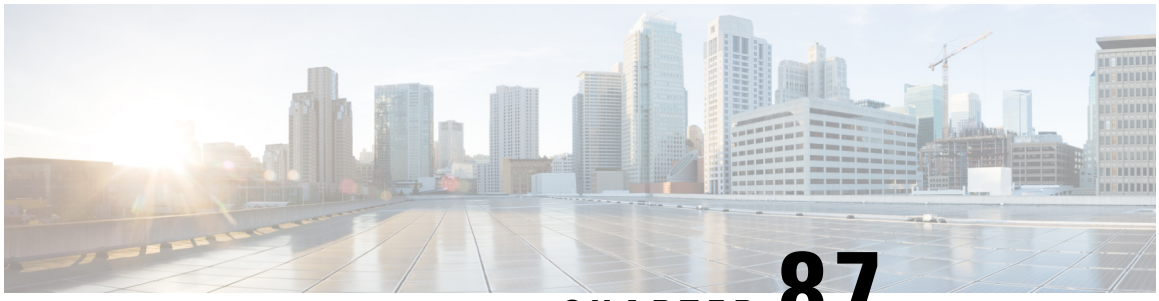
OSPF protocol --Open Shortest Path First. A link state routing protocol used for routing IP.

reoptimization--Reevaluation of the most suitable path for a tunnel to use, given the specified constraints.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

tailend --The downstream, receive end of a tunnel.

traffic engineering --Techniques and processes that cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.



CHAPTER 87

MPLS Traffic Engineering Configurable Path Calculation Metric for Tunnels

The MPLS Traffic Engineering--Configurable Path Calculation Metric for Tunnels feature enables the user to control the metric used in path calculation for traffic engineering (TE) tunnels on a per-tunnel basis. Certain tunnels are used to carry voice traffic, which requires low delay, and other tunnels are used to carry data. A TE link metric can be used to represent link delay and configure tunnels that carry voice traffic for path calculation and configure tunnels that carry data to use the Interior Gateway Protocol (IGP) metric for path calculation.

Prerequisites for MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

Before you configure tunnel path calculation metrics, your network must support the following Cisco IOS XE features:

- Multiprotocol Label Switching (MPLS) traffic engineering tunnels
- IP Cisco Express Forwarding
- Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS)

Restrictions for MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

- Unless explicitly configured, the TE link metric for a given link is the IGP link metric. When the TE link metric is used to represent a link property that is different from cost/distance, you must configure every network link that can be used for TE tunnels with a TE link metric that represents that property by using the **mpls traffic-eng administrative-weight** command. Failure to do so might cause tunnels to use unexpected paths.
- MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

Information About MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

Overview

When MPLS TE is configured in a network, the IGP floods two metrics for every link: the normal IGP (OSPF or IS-IS) link metric and a TE link metric. The IGP uses the IGP link metric in the normal way to compute routes for destination networks.

You can specify that the path calculation for a given tunnel be based on either of the following:

- IGP link metrics.
- TE link metrics, which you can configure so that they represent the needs of a particular application. For example, the TE link metrics can be configured to represent link transmission delay.

Benefits

When TE tunnels are used to carry two types of traffic, the Configurable Path Calculation Metric for Tunnels feature allows you to tailor tunnel path selection to the requirements of each type of traffic.

For example, suppose certain tunnels are to carry voice traffic (which requires low delay) and other tunnels are to carry data. In this situation, you can use the TE link metric to represent link delay and do the following:

- Configure tunnels that carry voice to use the TE link metric set to represent link delay for path calculation.
- Configure tunnels that carry data to use the IGP metric for path calculation.

How to Configure MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

Configuring a Platform to Support Traffic Engineering Tunnels

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **mpls traffic-eng tunnels**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding operation.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config)# mpls traffic-eng tunnels`	Enables the MPLS traffic engineering tunnel feature on a device.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IS-IS for MPLS Traffic Engineering

To configure IS-IS for MPLS traffic engineering, perform the following steps.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. Router(config)# **router isis**
2. Router(config-router)# **mpls traffic-eng level-1**
3. Router(config-router)# **mpls traffic-eng level-2**
4. Router(config-router)# **mpls traffic-eng router-id loopback 0**
5. Router(config-router)# **metric-style wide**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router isis**	Enables IS-IS routing and specifies an IS-IS process for IP. The router is placed in configuration mode.
Step 2	Router(config-router)# **mpls traffic-eng level-1**	Turns on MPLS traffic engineering for IS-IS level 1.
Step 3	Router(config-router)# **mpls traffic-eng level-2**	Turns on MPLS traffic engineering for IS-IS level 2.
Step 4	Router(config-router)# **mpls traffic-eng router-id loopback 0**	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface loopback0.
Step 5	Router(config-router)# **metric-style wide**	Configures a router to generate and accept only new-style type, length, value objects (TLVs).

Configuring OSPF for MPLS Traffic Engineering

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **mpls traffic-eng area** *number*
5. **mpls traffic-eng router-id loopback0**
6. **exit**

7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 200`	Configures an OSPF routing process for IP and enters router configuration mode. • The *process-id* is an internally used identification parameter for an OSPF routing process. It is locally assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**mpls traffic-eng area** *number* **Example:** `Router(config-router)# mpls traffic-eng area 0`	Turns on MPLS traffic engineering for the indicated OSPF area.
Step 5	**mpls traffic-eng router-id loopback0** **Example:** `Router(config-router)# mpls traffic-eng router-id loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface loopback0.
Step 6	**exit** **Example:** `Router(config-router)# exit`	Exits to global configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring Traffic Engineering Link Metrics

Unless explicitly configured, the TE link metric is the IGP link metric.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / subslot / port* [**.** *subinterface-number*]
4. **mpls traffic-eng administrative-weight** *weight*
5. **exit**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / subslot / port* [**.** *subinterface-number*] **Example:** `Router(config)# interface pos2/0/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the chassis slot number. Refer to the appropriate hardware manual for slot information. For SIPs, refer to the platform-specific SPA hardware installation guide or the corresponding "Identifying Slots and Subslots for SIPs and SPAs" topic in the platform-specific SPA software configuration guide. • The **/** *subslot* keyword and argument pair is the secondary slot number on a SIP where a SPA is installed. The slash (/) is required. Refer to the platform-specific SPA hardware installation guide and the corresponding "Specifying the Interface Address on a SPA" topic in the platform-specific SPA software configuration guide for subslot information. • The **/** *port* keyword and argument pair is the port or interface number. The slash (/) is required. Refer to the appropriate hardware manual for port information. For SPAs, refer to the corresponding "Specifying the Interface Address on a SPA" topics in the platform-specific SPA software configuration guide

	Command or Action	Purpose
		• The **.** *subinterface-number*keyword and argument pair is the subinterface number in the range 1 to 4294967293. The number that precedes the period (.) must match the number to which this subinterface belongs.
Step 4	**mpls traffic-eng administrative-weight** *weight* **Example:** `Router(config-if)# mpls traffic-eng administrative-weight 20`	Overrides the IGP administrative weight (cost) of the link. • The *weight* argument is the cost of the link.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel

This tunnel has two path setup options: a preferred explicit path and a backup dynamic path.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | **identifier** *path-number*}} [**lockdown**]
9. **exit**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface Tunnel0`	Configures an interface type and enters interface configuration mode. • The *number*argument is the number of the tunnel.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered loopback0`	Enables IP processing on an interface without assigning an explicit IP address to the interface. • The *type* and *number* arguments name the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface. • An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.4.4`	Specifies the destination for a tunnel interface. • The *ip-address* argument must be the MPLS traffic engineering router ID of the destination device.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 250`	Configures the bandwidth for the MPLS traffic engineering tunnel. • The *bandwidth* argument is a number in kilobits per second that is set aside for the MPLS traffic engineering tunnel. Range is from 1 to 4294967295. **Note** If automatic bandwidth is configured for the tunnel, use the **tunnel mpls traffic-eng bandwidth** command to configure the initial tunnel bandwidth, which is adjusted by the autobandwidth mechanism.

	Command or Action	Purpose
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | **identifier** *path-number*}} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 10 explicit identifier 321`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. • The *number* argument is the preference for this path option. When you configure multiple path options, lower numbered options are preferred. Valid values are from 1 to 1000. • The **dynamic** keyword indicates that the path of the label switched path (LSP) is dynamically calculated. • The **explicit** keyword indicates that the path of the LSP is an IP explicit path. • The **name** *path-name* keyword and argument are the path name of the IP explicit path that the tunnel uses with this option. • The **identifier** *path-number*keyword and argument pair names the path number of the IP explicit path that the tunnel uses with this option. The range is from 1 to 65535. • The **lockdown** keyword specifies that The LSP cannot be reoptimized. **Note** A dynamic path is used if an explicit path is currently unavailable.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Metric Type for Tunnel Path Calculation

Unless explicitly configured, the TE link metric type is used for tunnel path calculation. Two commands are provided for controlling the metric type to be used: an interface configuration command that specifies the metric type to be used for a particular TE tunnel and a global configuration command that specifies the metric type to be used for TE tunnels for which a metric type has not been specified by the interface configuration command.

Note If you do not enter either of the path selection metrics commands, the traffic engineering (TE) metric is used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng path-selection metric** {**igp** | **te**}
5. **exit**
6. **mpls traffic-eng path-selection metric** {**igp** | **te**}
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface Tunnel0`	Configures an interface type and enters interface configuration mode. • The *number*argument is the number of the tunnel.
Step 4	**tunnel mpls traffic-eng path-selection metric** {**igp**	**te**} **Example:** `Router(config-if)# tunnel mpls traffic-eng path-selection metric igp`	Specifies the metric type to use for path calculation for a tunnel. • The **igp** keyword specifies the use of the Interior Gateway Protocol (IGP) metric. • The **te** keyword specifies the use of the traffic engineering (TE) metric. This is the default.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**mpls traffic-eng path-selection metric** {**igp**	**te**} **Example:**	Specifies the metric type to use if a metric type was not explicitly configured for a given tunnel.

	Command or Action	Purpose
	`Router(config)# mpls traffic-eng path-selection metric igp`	• The **igp** keyword specifies the use of the Interior Gateway Protocol (IGP) metric. • The **te** keyword specifies the use of the traffic engineering (TE) metric. This is the default.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the Tunnel Path Metric Configuration

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng topolog** y
3. **show mpls traffic-eng tunnels**
4. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show mpls traffic-eng topolog** y

Use the **show mpls traffic-eng topology** command, which displays TE and IGP metrics for each link, to verify that link metrics have been correctly configured for a network. For example:

Example:

```
Router# show mpls traffic-eng topology
My_System_id: 1440.0000.0044.00 (isis  level-1)
IGP Id: 0090.0000.0009.00, MPLS TE Id:192.168.9.9 Router Node  (isis  level-1)
      link[0 ]:Nbr IGP Id: 0090.0000.0009.03, gen:7
          frag_id 0, Intf Address:10.0.0.99
          TE metric:100, IGP metric:48, attribute_flags:0x0     !!Note TE and IGP metrics
          physical_bw: 10000 (kbps), max_reservable_bw_global: 0 (kbps)
          max_reservable_bw_sub: 0 (kbps)
.
.
.
      link[1 ]:Nbr IGP Id: 0055.0000.0055.00, gen:7
          frag_id 0, Intf Address:10.205.0.9, Nbr Intf Address:10.205.0.55
```

```
          TE metric:120, IGP metric:10, attribute_flags:0x0     !!Note TE and IGP metrics
          physical_bw: 155000 (kbps), max_reservable_bw_global: 500000 (kbps)
          max_reservable_bw_sub: 0 (kbps)
.
.
.
```

Step 3 **show mpls traffic-eng tunnels**

Use the **show mpls traffic-eng tunnels** command, which displays the link metric used for tunnel path calculation, to verify that the desired link metrics are being used for each tunnel. For example:

Example:

```
Router# show mpls traffic-eng tunnels
Name: te3640-17-c_t221              (Tunnel22) Destination: 192.168.100.22
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type dynamic (Basis for Setup, path weight 10)
  Config Parameters:
    Bandwidth: 400 kps (Global)    Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: IGP                                          !!Note metric type
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 0   bw-based
    auto-bw: disabled(0/115) 0  Bandwidth Requested: 0
.
.
.
Name: te3640-17-c_t222              (Tunnel33) Destination: 192.168.100.22
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type dynamic (Basis for Setup, path weight 10)
  Config Parameters:
    Bandwidth: 200 kbps (Global)   Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: TE                                           !!Note metric type
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 0   bw-based
    auto-bw: disabled(0/115) 0  Bandwidth Requested: 0
.
.
.
```

Step 4 **exit**

Use this command to return to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

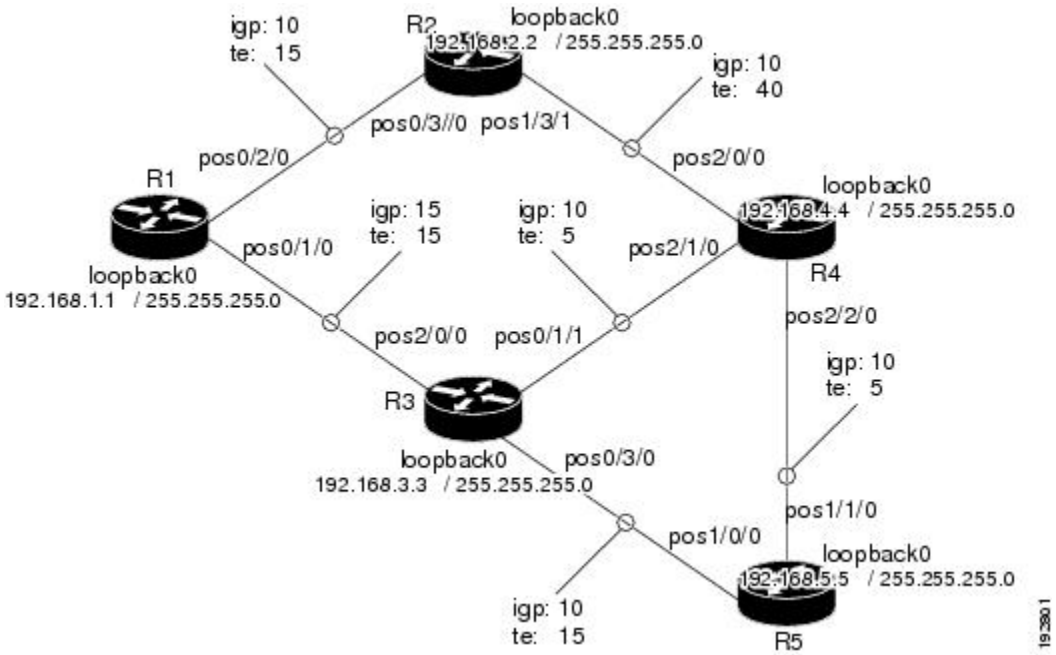
Configuration Examples for Configuring a Path Calculation Metric for Tunnels

Example Configuring Link Type and Metrics for Tunnel Path Selection

The section illustrates how to configure the link metric type to be used for tunnel path selection, and how to configure the link metrics themselves. The configuration commands included focus on specifying the metric type for path calculation and assigning metrics to links. Additional commands are required to fully configure the example scenario: for example, the IGP commands for traffic engineering and the link interface commands for enabling traffic engineering and specifying available bandwidth.

The examples in this section support the simple network technology shown in the figure below.

Figure 149: Network Topology



In the figure above:

- Tunnel1 and Tunnel2 run from R1 (headend) to R4 (tailend).
- Tunnel3 runs from R1 to R5.
- Path calculation for Tunnel1 and Tunnel3 should use a metric that represents link delay because these tunnels carry voice traffic.
- Path calculation for Tunnel2 should use IGP metrics because MPLS TE carries data traffic with no delay requirement.

Configuration fragments follow for each of the routers that illustrate the configuration relating to link metrics and their use in tunnel path calculation. TE metrics that represent link delay must be configured for the network links on each of the routers, and the three tunnels must be configured on R1.

These configuration fragments force Tunnel1 to take path R1-R3-R4, Tunnel2 to take path R1-R2-R4, and Tunnel3 to take path R1-R3-R4-R5 (assuming the links have sufficient bandwidth to accommodate the tunnels).

R1 Configuration

The following example shows how to configure the tunnel headend (R1) for Tunnel1, Tunnel2, and Tunnel3 in the figure above:

```
interface pos0/1/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface pos0/2/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface Tunnel1                               !Tunnel1 uses TE metric (default)
                                                !for path selection
ip unnumbered loopback0
tunnel destination 192.168.4.4 255.255.255.0
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 1000
tunnel mpls traffic-eng path-option 1 dynamic
interface Tunnel2                                !Tunnel2 uses IGP metric
                                                 !for path selection
ip unnumbered loopback0
tunnel destination 192.168.4.4 255.255.255.0
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 1000
tunnel mpls traffic-eng path-option 1 dynamic
tunnel mpls traffic-eng path-selection-metric igp !Use IGP cost for path selection.
interface Tunnel3                                  !Tunnel3 uses TE metric (default)
                                                   !for path selection
ip unnumbered loopback0
tunnel destination 192.168.5.5 255.255.255.0
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 1000
tunnel mpls traffic-eng path-option 1 dynamic
```

R2 Configuration

The following example shows how to configure R2 in the figure above:

```
interface pos0/3/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface pos1/3/1
mpls traffic-eng administrative-weight 40       !TE metric different from IGP metric
```

R3 Configuration

The following example shows how to configure R3 in the figure above:

```
interface pos2/0/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface pos0/3/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface pos0/1/1
mpls traffic-eng administrative-weight 5        !TE metric different from IGP metric
```

R4 Configuration

The following example shows how to configure R4 in the figure above:

```
interface pos2/0/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface pos2/1/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface pos2/2/0
mpls traffic-eng administrative-weight 5        !TE metric different from IGP metric
```

R5 Configuration

The following example shows how to configure R5 in the figure above:

```
interface pos1/0/0
mpls traffic-eng administrative-weight 15       !TE metric different from IGP metric
interface pos1/1/0
mpls traffic-eng administrative-weight 5        !TE metric different from IGP metric
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuration tasks for IS-IS and OSPF	*Cisco IOS XE IP Routing Protocols Configuration Guide*
IS-IS and OSPF commands	*Cisco IOS IP Routing Protocols Command Reference*
Configuration tasks for MPLS and MPLS TE	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*
MPLS TE commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Configuration tasks for tunnels	• *Cisco IOS XE Interface and Hardware Component Configuration Guide* • *Cisco IOS XE Multiprotocol Label Switching Configuration Guide*
Tunnel configuration commands	• *Cisco IOS Interface and Hardware Component Command Reference* • *Cisco IOS XE Multiprotocol Label Switching Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	-

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified.	-

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

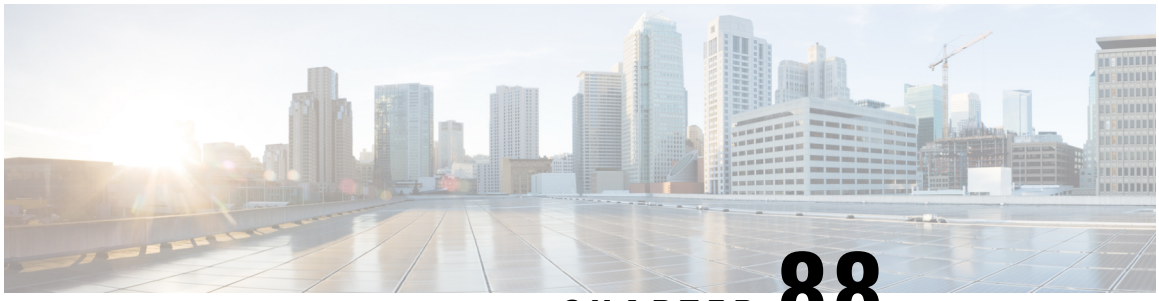
Feature Information for MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 164: Feature Information for MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

Feature Name	Releases	Feature Information
MPLS Traffic Engineering:Configurable Path Calculation Metric for Tunnels	12.0(18)ST 12.2(11)S 12.2(14)S 12.2(28)SB 12.4(20)T Cisco IOS XE Release 2.3	The MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels feature enables the user to control the metric used in path calculation for traffic engineering (TE) tunnels on a per-tunnel basis. Certain tunnels are used to carry voice traffic, which requires low delay, and other tunnels are used to carry data. A TE link metric can be used to represent link delay and configure tunnels that carry voice traffic for path calculation and configure tunnels that carry data to use the Interior Gateway Protocol (IGP) metric for path calculation. The following commands were introduced or modified: **mpls traffic-eng path-selection metric**, **tunnel mpls traffic-eng path-selection metric**.



CHAPTER 88

MPLS Traffic Engineering--Scalability Enhancements

The MPLS Traffic Engineering--Scalability Enhancement feature improves scalability performance for large numbers of traffic engineering tunnels.

These improvements allow an increase in the number of traffic engineering (TE) tunnels a router can support when the router is configured as a tunnel headend. Additionally, when the router is configured as a tunnel midpoint, the enhancements reduce the time required to establish large numbers of TE tunnels.

This feature module contains information about and instructions on how to configure the Multiprotocol Label Switching (MPLS) traffic engineering scalability enhancements.

Prerequisites for MPLS Traffic Engineering--Scalability Enhancements

Your network must support the following Cisco IOS XE features before you enable MPLS traffic engineering:

- MPLS
- Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)

Restrictions for MPLS Traffic Engineering--Scalability Enhancements

The number of tunnels that a particular platform can support can vary depending on:

- The types of interfaces that the tunnels traverse
- The manner in which the Resource Reservation Protocol (RSVP) message pacing feature is configured
- MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

Information About MPLS Traffic Engineering--Scalability Enhancements

Scalability Enhancements for Traffic Engineering Tunnels

Scalability performance is improved for large numbers of traffic engineering tunnels, and includes the following enhancements:

- Increase the number of traffic engineering tunnels a router can support when configured as a tunnel headend and when configured as a tunnel midpoint
- Reduce the time required to establish large numbers of traffic engineering tunnels

RSVP Rate Limiting

A burst of RSVP traffic engineering signaling messages can overflow the input queue of a receiving router, causing some messages to be dropped. Dropped messages cause a substantial delay in completing label switched path (LSP) signaling.

This MPLS Traffic Engineering--Scalability Enhancements feature provides an enhancement mechanism that controls the transmission rate for RSVP messages and reduces the likelihood of input drops on the receiving router. The default transmission rate is 200 RSVP messages per second to a given neighbor. The rate is configurable.

Improved Recovery Response for Signaling and Management of MPLS Traffic Engineering Tunnels

The MPLS Traffic Engineering--Scalability Enhancements feature improves the recovery response for signaling and management of MPLS TE tunnels. LSP recovery responsiveness is improved when a link used by an LSP fails:

- When the upstream end of a failed link detects the failure, the software generates an RSVP No Route path error message. This enables the LSP headend to detect the link failure and initiate recovery, even when the Interior Gateway Protocol (IGP) update announcing the link failure is delayed.
- The LSP headend marks the link in question so that subsequent constraint-based shortest path first (SPF) calculations ignore the link until either a new IGP update arrives or a configurable timeout occurs. This ensures that resignaling to restore the LSP avoids the failed link.

IS-IS and MPLS Traffic Engineering Topology Database Interactions

The MPLS Traffic Engineering--Scalability Enhancements feature reduces the interval between when the IS-IS protocol receives an IGP update and when it delivers the update to the MPLS traffic engineering topology database.

Before the MPLS Traffic Engineering--Scalability Enhancements feature was introduced, when IS-IS received a new LSP that contained traffic engineering type, length, value (TLV) objects, a delay of several seconds could occur before IS-IS passed the traffic engineering TLVs to the traffic engineering database. The purpose of the delay was to provide better scalability during periods of network instability and to give the router an opportunity to receive more fragments of the LSP before passing the information to the traffic engineering database. However, this delay increased the convergence time for the traffic engineering database.

With the MPLS Traffic Engineering--Scalability Enhancements feature, IS-IS extracts traffic engineering TLVs from received LSPs and passes them to the traffic engineering database immediately. The exception to this occurs when there are large numbers of LSPs to process and it is important to limit CPU consumption, such as during periods of network instability. The parameters that control IS-IS delivery of traffic engineering TLVs to the traffic engineering topology database are configurable.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

Improved Counter Capabilities for MPLS TE Tunnels Events and RSVP Signaling

With the MPLS Traffic Engineering--Scalability Enhancements feature, diagnostic and troubleshooting capabilities for MPLS traffic engineering tunnels and RSVP are improved:

- Counters record tunnel headend error events such as no route (link down), preemption, and insufficient bandwidth on a per-tunnel basis.
- Counters record RSVP messages. The counters are per-interface and record the number of RSVP messages of each type sent and received on the interface.

Benefits of MPLS Traffic Engineering--Scalability Enhancements

The MPLS Traffic Engineering--Scalability Enhancements feature provides the following benefits:

- Increased scalability--Up to 600 MPLS traffic engineering tunnel headends are supported. Up to 10,000 traffic engineering tunnel midpoints are supported, with up to 5000 midpoints per interface.

- Faster recovery after failure conditions--Message pacing provides a mechanism to throttle RSVP control messages so that they are less likely to be dropped. This results in a faster recovery from failure conditions when many MPLS traffic engineering tunnels are being set up.
- Improved reroute time--When a traffic engineering tunnel is down, the headend router needs to be notified so that it can signal for a new LSP for the tunnel along an alternate path. The headend router does not have to wait for an IGP update to signal for a new LSP for the tunnel along an alternate path.
- Improved tunnel setup time--Fewer control messages and tunnel setup messages are dropped. This reduces the average time required to set up tunnels.

How to Configure MPLS Traffic Engineering--Scalability Enhancements

Enabling RSVP Rate Limiting for MPLS Traffic Engineering Scalability Enhancements

Perform the following task to enable RSVP rate limiting for MPLS traffic engineering scalability enhancements. RSVP rate limiting maintains, on an outgoing interface basis, a count of messages that were dropped because the output queue for the interface used for rate limiting was full.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling rate-limit** [**burst** *number*] [**limit** *number*] [**maxsize** *bytes*] [**period** *ms*]
4. **end**
5. **show ip rsvp neighbor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling rate-limit** [**burst** *number*] [**limit** *number*] [**maxsize** *bytes*] [**period** *ms*] **Example:**	Controls the transmission rate for RSVP messages sent to a neighboring router during a specified amount of time.

	Command or Action	Purpose
	`Router(config)# ip rsvp signalling rate-limit burst 5 maxsize 3 period 2`	• The **burst** *number*keyword and argument pair indicates the maximum number of RSVP messages sent to a neighboring router during each interval. The range is from 1 to 5000. The default is 8. • The **limit** *number* keyword and argument pair indicates the maximum number of messages to send per queue interval when the number of messages sent is less than the number of messages to be sent normally. The range is 1 to 5000. The default is 37. • The **maxsize** *bytes* keyword and argument pair indicates the maximum size of the message queue, in bytes. The range is 1 to 5000. The default is 2000. • The **period** *ms* keyword and argument pair indicates the length of the interval (time frame) in milliseconds (ms). The range is 10 to 5000. The default is 20.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 5	**show ip rsvp neighbor** **Example:** `Router# show ip rsvp neighbor`	Displays current RSVP neighbors. Use this command to verify that RSVP message pacing is enabled.

Managing Link Failure Timeouts for MPLS Traffic Engineering Tunnels

Perform this task to manage link failure timeouts for MPLS traffic engineering tunnels.

This allows the configuration of a timeout during which the router ignores a link in its path calculation to avoid paths that contain a failed link and are likely to fail when signaled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng topology holddown sigerr** *seconds*
4. **end**
5. **show mpls traffic-eng topology** [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng topology holddown sigerr** *seconds* **Example:** `Router(config)# mpls traffic-eng topology holddown sigerr 15`	Specifies the amount of time that a router ignores a link in its traffic engineering topology database in tunnel path Constrained Shortest Path First (CSPF) computations following a traffic engineering tunnel error on the link. • The *seconds* argument specifies the length of time (in seconds) a router should ignore a link during tunnel path calculations following a traffic engineering tunnel error on the link. The range is 0 to 300. The default is 10.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 5	**show mpls traffic-eng topology** [**brief**] **Example:** `Router# show mpls traffic-eng topology brief`	Displays the MPLS traffic engineering global topology as currently known at this node. • The **brief** keyword provides a less detailed version of the topology.

Controlling IS-IS Communication with the MPLS Traffic Engineering Topology Database

Perform the following task to control IS-IS and MPLS traffic engineering topology database interactions. This reduces the interval time between when the IS-IS protocol receives an IGP update and when IS-IS delivers the update to the MPLS traffic engineering topology database, which reduces convergence time for the database.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **router isis** [*area-tag*]
4. **mpls traffic-eng scanner** [**interval** *seconds*] [**max-flash** *LSPs*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Router(config)# router isis`	Enables the IS-IS routing protocol and specifies an IS-IS process. • The *area-tag* argument is a meaningful name for a routing process. If it is not specified, a null tag is assumed and the process is referenced with a null tag. This name must be unique among all IP or Connectionless Network Service (CLNS) router processes for a given router. **Note** This argument is Required for multiarea IS-IS configuration and optional for conventional IS-IS configuration.
Step 4	**mpls traffic-eng scanner** [**interval** *seconds*] [**max-flash** *LSPs*] **Example:** `Router(config-router)# mpls traffic-eng scanner interval 5 max-flash 100`	Specifies how often IS-IS extracts traffic engineering TLVs from flagged LSPs and passes them to the traffic engineering topology database, and specifies the maximum number of LSPs that the router can process immediately. • The **interval** *seconds* keyword and argument specify the frequency, in seconds, at which IS-IS sends traffic engineering TLVs into the traffic engineering database. The range is 1 to 60. The default is 5. • The **max-flash** *LSPs* keyword and argument specify the maximum number of LSPs that the router can process immediately without incurring a delay. The range is 0 to 200. The default is 15.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits to privileged EXEC mode.

Monitoring and Maintaining MPLS TE Scalability Enhancements

SUMMARY STEPS

1. **enable**
2. **show ip rsvp neighbor** [**detail**]
3. **show ip rsvp counters** [**summary**]
4. **clear ip rsvp counters**
5. **clear ip rsvp signalling rate-limit**
6. **show mpls traffic-eng tunnels statistics**
7. **clear mpls traffic-eng tunnels counters**
8. **show mpls traffic-eng topology** [**brief**]
9. **exit**

DETAILED STEPS

Step 1 enable

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 show ip rsvp neighbor [detail]

Use this command to verify that RSVP message pacing is turned on. For example:

Example:

```
Router# show ip rsvp neighbor detail
Neighbor:10.0.0.1
    Encapsulation:RSVP
    Rate-Limiting:
      Dropped messages:0
Refresh Reduction:
      Remote epoch:0x1BFEA5
      Out of order messages:0
      Retransmitted messages:0
      Highest rcvd message id:1059
      Last rcvd message:00:00:04
Neighbor:10.0.0.2
    Encapsulation:RSVP
    Rate-Limiting:
      Dropped messages:0
    Refresh Reduction:
      Remote epoch:0xB26B1
      Out of order messages:0
      Retransmitted messages:0
      Highest rcvd message id:945
      Last rcvd message:00:00:05
```

Step 3 show ip rsvp counters [summary]

Use this command to display the counts of RSVP messages that were sent and received. For example:

Example:

```
Router# show ip rsvp counters summary
All Interfaces          Recv      Xmit                          Recv      Xmit
    Path                  110        15    Resv                    50        28
    PathError               0         0    ResvError                0         0
    PathTear                0         0    ResvTear                 0         0
    ResvConf                0         0    RTearConf                0         0
    Ack                     0         0    Srefresh                 0         0
    Hello                5555      5554    IntegrityChalle          0         0
    IntegrityRespon         0         0    DSBM_WILLING             0         0
    I_AM_DSBM               0         0
    Unknown                 0         0    Errors                   0         0
Recv Msg Queues                Current       Max
    RSVP                             0         2
    Hello (per-I/F)                  0         1
    Awaiting Authentication          0         0
```

Step 4 **clear ip rsvp counters**

Use this command to clear (set to zero) all IP RSVP counters that are being maintained. For example:

Example:

```
Router# clear ip rsvp counters
Clear rsvp counters [confirm]
```

Step 5 **clear ip rsvp signalling rate-limit**

Use this command to clear (set to zero) counts of the messages that message pacing was forced to drop because the output queue for the interface used for message pacing was full. For example:

Example:

```
Router# clear ip rsvp signalling rate-limit
```

Step 6 **show mpls traffic-eng tunnels statistics**

Use this command to display event counters for one or more MPLS traffic engineering tunnels. For example:

Example:

```
Router# show mpls traffic-eng tunnels statistics
Tunnel1001 (Destination 10.8.8.8; Name Router_t1001)
  Management statistics:
    Path: 25 no path, 1 path no longer valid, 0 missing ip exp path
 5 path changes
    State: 3 transitions, 0 admin down, 1 oper down
  Signalling statistics:
    Opens: 2 succeeded, 0 timed out, 0 bad path spec
 0 other aborts
    Errors: 0 no b/w, 0 no route, 0 admin
 0 bad exp route, 0 rec route loop, 0 other

...
```

Example:

```
Tunnel7050 (Destination 10.8.8.8; Name Router_t7050)
  Management statistics:
    Path:  19 no path, 1 path no longer valid, 0 missing ip exp path
 3 path changes
```

```
   State:  3 transitions, 0 admin down, 1 oper down
 Signalling statistics:
   Opens:  2 succeeded, 0 timed out, 0 bad path spec
0 other aborts
   Errors: 0 no b/w, 0 no route, 0 admin
0 bad exp route, 0 rec route loop, 0 other
```

Step 7 **clear mpls traffic-eng tunnels counters**

Use this command to clear counters for all MPLS traffic engineering tunnels. For example:

Example:

```
Router# clear mpls traffic-eng tunnels counters
Clear traffic engineering tunnel counters [confirm]
```

Step 8 **show mpls traffic-eng topology** [**brief**]

Use this command to display the MPLS traffic engineering topology database. For example:

Example:

```
Router# show mpls traffic-eng topology brief
My_System_id:0000.0000.0003.00 (isis  level-2)
Signalling error holddown:10 sec Global Link Generation 9
IGP Id:0000.0000.0003.00, MPLS TE Id:10.0.3.1 Router Node  (isis
level-2)
      link[0]:Point-to-Point, Nbr IGP Id:0000.0000.0004.00,
nbr_node_id:2, gen:9
          frag_id 0, Intf Address:10.0.0.33, Nbr Intf Address:10.0.0.34
          TE metric:10, IGP metric:10, attribute_flags:0x0
          SRLGs:1 2
```

Step 9 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS Traffic Engineering--Scalability Enhancements

Example Enabling RSVP Rate Limiting for MPLS Traffic Engineering Scalability Enhancements

The following examples show how to enable RSVP rate limiting for MPLS traffic engineering scalability enhancements:

```
configure terminal
```

```
 ip rsvp signalling rate-limit
 end
```

The following is sample output that traffic engineering displays when RSVP rate limiting is enabled:

```
Router# show ip rsvp signalling rate-limit
Rate Limiting: enabled
  Burst: 10
  Limit: 37
  Maxsize: 5000
  Period (msec): 100
  Max rate (msgs/sec): 100
```

The following example shows how to configure a router to send a maximum of 5 RSVP traffic engineering signaling messages in 1 second to a neighbor. The size of the output queue is 35.

```
configure terminal
 ip rsvp signalling rate-limit
period 1 burst 5 maxsize 35
```

Example Managing Link Failure Timeouts for MPLS Traffic Engineering Tunnels

The following example shows how to manage link failure timeouts for MPLS traffic engineering tunnels:

```
configure terminal
 mpls traffic-eng topology holddown sigerr 15
 end
```

In this example, the link hold-down time for signaling errors is set to 15 seconds.

Example Controlling IS-IS Communication with the MPLS Traffic Engineering Topology Database

The following example shows how to control IS-IS communication with the MPLS traffic engineering topology database:

```
configure terminal
router isis
 mpls traffic-eng scanner interval 5 max-flash 50
 end
```

In this example, the router is enabled to process up to 50 IS-IS LSPs without any delay.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
Quality of service	• *Cisco IOS Quality of Service Solutions Command Reference* • *Cisco IOS XE Quality of Service Solutions Configuration Guide,* Release 2
MPLS	• *Cisco IOS Multiprotocol Label Switching Command Reference* • *Cisco IOS XE Multiprotocol Label Switching Configuration Guide,* Release 2

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering Scalability Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 165: Feature Information for MPLS Traffic Engineering Scalability Enhancements

Feature Name	Releases	Feature Information
MPLS Traffic Engineering: Scalability Enhancements	Cisco IOS XE Release 2.3	The MPLS Traffic Engineering--Scalability Enhancements feature improves scalability performance for large numbers of traffic engineering tunnels. These improvements allow an increase in the number of traffic engineering (TE) tunnels a router can support when the router is configured as a tunnel headend. Additionally, when the router is configured as a tunnel midpoint, the enhancements reduce the time required to establish large numbers of TE tunnels. This feature module contains information about and instructions on how to configure the Multiprotocol Label Switching (MPLS) traffic engineering scalability enhancements. The following commands were introduced or modified: **clear ip rsvp counters**, **clear ip rsvp signalling rate-limit**, **clear mpls traffic-eng tunnel counters**, **ip rsvp signalling rate-limit**, **mpls traffic-eng scanner**, **mpls traffic-eng topology holddown sigerr**, **show ip rsvp counters**, and **show mpls traffic-eng tunnels statistics**.

Glossary

bundled interface—Generic terms to represent port-channel, multilink, and VLAN interfaces.

Cisco express forwarding —A means for accelerating the forwarding of packets within a router, by storing route lookup information in several data structures instead of in a route cache.

CLNS —Connectionless Network Service. The Open Systems Interconnection (OSI) network layer service that does not require a circuit to be established before data is transmitted. CLNS routes messages to their destination independently of any other messages.

CSPF —Constrained Shortest Path First. A routing protocol that calculates the shortest path based on a set of constraints, such as a minimum bandwidth requirement, maximum number of nodes, or nodes to include or exclude.

enterprise network —A large and diverse network connecting most major points in a company or other organization.

FRR—Fast ReRoute.

headend —The endpoint of a broadband network. All stations send toward the headend; the headend then sends toward the destination stations.

IGP —Interior Gateway Protocol. An Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

interface —A network connection.

IS-IS —Intermediate System to Intermediate System. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, where ISs (routers) exchange routing information based on a single metric, to determine the network topology.

LDN— Link Down Notification.

LSP —Label-Switched Path. A sequence of hops (R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A label-switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

member links—Individual interfaces that are grouped into a bundled interface.

message-pacing —The former name of the rate limiting feature.

MPLS —Formerly known as tag switching, Multiprotocol Label Switching is a method for directing packets primarily through Layer 2 switching rather than Layer 3 routing. In MPLS, packets are assigned short fixed-length labels at the ingress to an MPLS cloud by using the concept of forwarding equivalence classes. Within the MPLS domain, the labels are used to make forwarding decisions mostly without recourse to the original packet headers.

OSPF —Open Shortest Path First. A link-state, hierarchical Interior Gateway Protocol (IGP) routing protocol. derived from the Intermediate System-Intermediate System (IS-IS) protocol. OSPF features are least-cost routing, multipath routing, and load balancing.

router —A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RSVP —Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network.

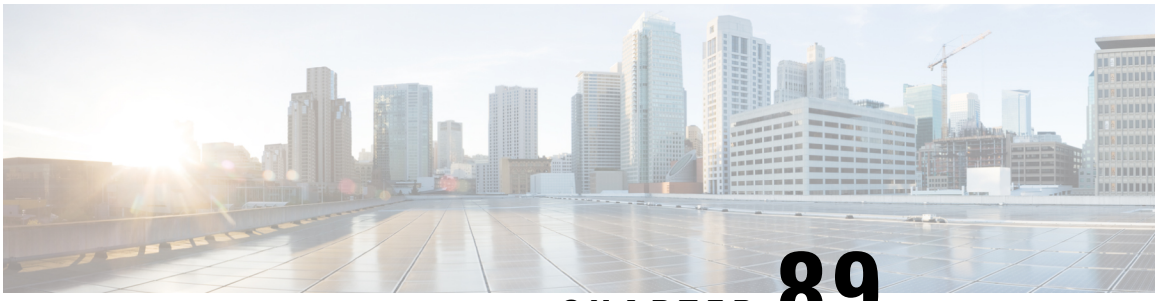
scalability —An indicator showing how quickly some measure of resource usage increases as a network gets larger.

TLV —type, length, value. TLV objects are used in data communication to provide optional information. The type field indicates the type of items in the value field. The length field indicates the length of the value field. The value field is the data portion of the packet.

topology —The physical arrangement of network nodes and media within an enterprise networking structure.

TE (traffic engineering) —Techniques and processes that cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

traffic engineering tunnel —A label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing would cause the tunnel to take.



CHAPTER 89

MPLS Traffic Engineering--LSP Attributes

This document describes how to configure label switched path (LSP) attributes for path options associated with Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels.

The MPLS Traffic Engineering--LSP Attributes feature is an extension to MPLS TE that provides an LSP Attribute list feature and a Path Option for Bandwidth Override feature. These features provide flexibility in the configuration of LSP attributes for MPLS TE tunnel path options. Several LSP attributes can be applied to path options for TE tunnels using an LSP attribute list. If bandwidth is the only LSP attribute you require, then you can configure a Path Option for Bandwidth Override.

Prerequisites for MPLS Traffic Engineering--LSP Attributes

TheMPLS Traffic Engineering--LSP Attributes feature requires that you configure an MPLS TE tunnel before you configure either an LSP Attribute List or a Path Option for Bandwidth Override feature.

Restrictions for MPLS Traffic Engineering--LSP Attributes

Reoptimization between path options with different bandwidth pool types (subpool versus global pool) and different priorities is not supported. Specifically,

- With the Path Option for Bandwidth Override feature, you need to configure bandwidth for path options with the same bandwidth pool as configured for the tunnel.
- With the LSP Attribute List feature, you need to configure both a bandwidth pool and priority for path options that are consistent with the bandwidth pool and priority configured on the tunnel or in other path options used by the tunnel.

Information About MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

MPLS Traffic Engineering--LSP Attributes Benefits

The MPLS Traffic Engineering--LSP Attributes feature provides an LSP Attribute List feature and a Path Option for Bandwidth Override feature. These features have the following benefits:

- The LSP Attributes List feature provides the ability to configure values for several LSP-specific path options for TE tunnels.
- One or more TE tunnels can specify specific path options by referencing an LSP Attribute List.
- LSP attribute lists make the MPLS TE user interface more flexible, easier to use, and easier to extend and maintain.
- The Path Option for Bandwidth Override feature provides a single command that allows a TE tunnel to fall back temporarily to path options that can reduce bandwidth constraints.

Traffic Engineering Bandwidth and Bandwidth Pools

MPLS traffic engineering allows constraint-based routing (CBR) of IP traffic. One of the constraints satisfied by CBR is the availability of required bandwidth over a selected path. Regular TE tunnel bandwidth is called the global pool. Subpool bandwidth is a portion of the global pool. Subpool bandwidth is not reserved from the global pool if it is not in use. Therefore, subpool tunnels require a higher priority than nonsubpool tunnels.

You can configure the LSP Attribute bandwidth path option to use either global pool (default) or subpool bandwidth. The bandwidth value for the path option may be any valid value and the pool does not have to be the same as that configured on the tunnel.

Note When you configure bandwidth for path options with the **bandwidth** [**sub-pool** | **global**] *kbps* command, use either all subpool bandwidths or all global-pool bandwidths.

You can configure bandwidth on both dynamic and explicit path options using either the LSP Attribute List feature or the Path Option for Bandwidth Override feature. The commands that enable these features are exclusive of each other. If bandwidth is the only LSP attribute that you need to set on the path option, then use the command to enable the feature. This is the simplest way to configure multiple path options with decreasing bandwidth constraints. Once the **bandwidth** keyword is entered on the **tunnel mpls traffic-eng path-option** command in interface configuration mode, you cannot configure an LSP Attribute List for that path option.

Tunnel Attributes and LSP Attributes

Cisco IOS XE tunneling interfaces have many parameters associated with MPLS TE. Typically, you configure these parameters with **tunnel mpls traffic-eng** commands in interface configuration mode. Many of these commands determine tunnel-specific properties, such as the load-sharing factor for the tunnel. These commands

configure parameters that are unrelated to the particular LSP in use by the tunnel. However, some of the tunneling parameters apply to the LSP that the tunnel uses. You can configure the LSP-specific properties using an LSP Attribute list.

LSP Attributes and the LSP Attribute List

An LSP Attribute list can contain values for each LSP-specific parameter that is configurable for a TE tunnel. You configure an LSP attribute list with the **mpls traffic-eng lsp attributes** *string* command, where *string* identifies the attribute list. The LSP attributes that you can specify include the following:

- Attribute flags for links that make up the LSP (**affinity** command)
- Automatic bandwidth configuration (**auto-bw** command)
- LSP bandwidth--global pool or subpool (**bandwidth** command)
- Disable reoptimization of the LSP (**lockdown** command)
- LSP priority (**priority** command)
- Protection failure (**protection** command)
- Record the route used by the LSP (**record-route** command)

LSP Attribute Lists Management

The MPLS Traffic Engineering--LSP Attributes feature also provides commands that help you manage LSP Attribute lists. You can do the following:

- Relist all attribute list entries (**list** command)
- Remove a specific attribute from the list (**no***attribute* command)

The **exit** command exits from the LSP attributes configuration submode and returns you to global configuration mode.

Based on your requirements, you can configure LSP attributes lists with different sets of attributes for different path options. LSP attribute lists also provide an easy way to configure multiple TE tunnels to use the same LSP attributes. That is, you can reference the same LSP attribute list to configure LSP-specific parameters for one or more TE tunnels.

Constraint-Based Routing and Path Option Selection

MPLS traffic engineering automatically establishes and maintains LSPs across the backbone by using the Resource Reservation Protocol (RSVP). The path that an LSP uses is determined by the LSP resource requirements and network resources, such as bandwidth. Traffic engineering tunnels are calculated at the LSP head based on a fit between required and available resources (constraint-based routing).

Without the Path Option for Bandwidth Override feature, a TE tunnel establishes an LSP based on dynamic or explicit path options in order of preference. However, the bandwidth and other attributes configured on the TE tunnel allow the setup of an LSP only if LSP path options satisfy the constraints. If a path cannot be found that satisfies the configured path options, then the tunnel is not set up.

The Path Option for Bandwidth Override feature provides a fallback path option that allows overriding the bandwidth configured on the TE tunnel interface. For example, you can configure a path option that sets the bandwidth to zero (0) effectively removing the bandwidth constraint imposed by the constraint-based routing calculation.

Tunnel Reoptimization and Path Option Selection

Reoptimization occurs when a device with traffic engineering tunnels periodically examines tunnels with established LSPs to learn if better LSPs are available. If a better LSP seems to be available, the device attempts to signal the better LSP. If the signaling is successful, the device replaces the older LSP with the new, better LSP.

Reoptimization can be triggered by a timer, the issuance of an **mpls traffic-eng reoptimize** command, or a configuration change that requires the resignalling of a tunnel. The MPLS AutoBandwidth feature, for example, uses a timer to set the frequency of reoptimization based on the bandwidth path option attribute. The Path Option for Bandwidth Override feature allows for the switching between bandwidth configured on the TE tunnel interface and bandwidth configured on a specific path option. This increases the success of signaling an LSP for the TE tunnel.

With bandwidth override configured on a path option, the traffic engineering software attempts to reoptimize the bandwidth every 30 seconds to reestablish the bandwidth configured on the tunnel (see the Configuring a Path Option for Bandwidth Override section).

You can disable reoptimization of an LSP with the **lockdown** command in an LSP Attribute list. You can apply the LSP Attribute list containing the **lockdown** command to a path option with the **tunnel mpls traffic-eng path-option** command.

Note When you configure bandwidth for path options with the **bandwidth** [**sub-pool** | **global**] *kpbs* command, use either all subpool bandwidths or all global-pool bandwidths. Do not mix subpool and nonsubpool bandwidths, otherwise the path option does not reoptimize later.

Path Option Selection with Bandwidth Override

The Path Option for Bandwidth Override feature allows you to configure bandwidth parameters on a specific path option. The **tunnel mpls traffic-eng path-option** command's **bandwidth** keyword can be used for this purpose. When an LSP is signaled using a path option with a configured bandwidth, the bandwidth associated with the path option is signaled instead of the tunnel's configured bandwidth.

This feature also provides the ability to configure multiple path options that reduce the bandwidth constraint each time the headend of a tunnel fails to establish an LSP.

The following configuration uses the **tunnel mpls traffic-eng bandwidth** command to configure the bandwidth of the tunnel and three **tunnel mpls traffic-eng path-option** commands that define the signalling path options for the LSP:

```
tunnel mpls traffic-eng bandwidth 1000
tunnel mpls traffic-eng path-option 1 explicit name path1
tunnel mpls traffic-eng path-option 2 explicit name path2 bandwidth 500
tunnel mpls traffic-eng path-option 3 dynamic bandwidth 0
```

The device selects a path option for an LSP in order of preference, as follows:

- The device attempts to signal an LSP using path options starting with path option 1.

The device attempts to signal an LSP with the 1000 kbps bandwidth configured on the tunnel interface because path-option 1 has no bandwidth configured.

- If 1000 kbps bandwidth is not available over the network, the device attempts to establish an LSP using path-option 2.

Path option 2 has a bandwidth of 500 kbps configured. This reduces the bandwidth constraint from the original 1000 kbps configured on the tunnel interface.

- If 500 kbps is not available, the device attempts to establish an LSP using path-option 3.

Path-option 3 is configured as dynamic and has bandwidth 0. The device establishes the LSP if an IP path exists to the destination and all other tunnel constraints are met.

Default Path Option Attributes for TE Tunnels Using LSP Attribute Lists

Values for path option attributes for a TE tunnel are determined in this manner:

- LSP attribute list values referenced by the path option take precedence over the values configured on the tunnel interface.
- If an attribute is not specified in the LSP attribute list, the device uses the attribute in the tunnel configuration. LSP attribute lists do not have defaults.
- If the attribute is not configured on the tunnel, then the device uses the tunnel default value, as follows:

{affinity= affinity 0 mask 0,

auto-bw= no auto-bw,

bandwidth= bandwidth 0,

lockdown= no lockdown,

priority= priority 7 7,

protection fast-reroute= no protection fast-reroute,

record-route= no record-route

.

.

.

}

How to Configure MPLS Traffic Engineering--LSP Attributes

Configuring an LSP Attribute List

Perform this task to configure a label switched path (LSP) attribute list with the desired attributes to be applied on a path option. Based on your requirements, you can configure LSP attributes lists with different sets of attributes for different path options. The LSP attribute list provides a user interface that is flexible, easy to use, and easy to extend and maintain for the configuration of MPLS TE tunnel path options.

LSP attribute lists also provide an easy way to configure multiple TE tunnels to use the same LSP attributes. That is, you can reference the same LSP attribute list to configure LSP-specific parameters for one or more TE tunnels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng lsp attributes** *string*
4. **affinity** *value* [**mask** *value*]
5. **auto-bw** [**frequency** *secs*] [**max-bw** *kbps*] [**min-bw** *kbps*] [**collect-bw**]
6. **bandwidth** [**sub-pool**| **global**] *kbps*
7. **list**
8. **lockdown**
9. **priority** *setup-priority* [*hold-priority*]
10. **protection fast-reroute**
11. **record-route**
12. **no** *sub-command*
13. **exit**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng lsp attributes** *string* **Example:**	Configures an LSP attribute list and enters LSP Attributes configuration mode.

	Command or Action	Purpose
	`Router(config)# mpls traffic-eng lsp attributes 1`	• The *string* argument identifies a specific LSP attribute list.

Step 4	**affinity** *value* [**mask** *value*]
	Example:
	`Router(config-lsp-attr)# affinity 0 mask 0`
		(Optional) Specifies attribute flags for links comprising an LSP.
		• The *value* argument is a value required for links that make up an LSP. Values of the bits are either 0 or 1.
		• The **mask** *value* keyword argument combination indicates which attribute values should be checked.
		  • If a bit in the mask is 0, an attribute value of the link or that bit is irrelevant.
		  • If a bit in the mask is 1, the attribute value of that link and the required affinity of the LSP for that bit must match.

Step 5	**auto-bw** [**frequency** *secs*] [**max-bw** *kbps*] [**min-bw** *kbps*] [**collect-bw**] **Example:** `Router(config-lsp-attr)# auto-bw`	(Optional) Specifies automatic bandwidth configuration. • The **frequency** *secs* keyword argument combination specifies the interval between bandwidth adjustments. The specified interval can be from 300 to 604800 seconds. • The **max-bw** *kbps* keyword argument combination specifies the maximum automatic bandwidth, in kbps, for this path option. The value can be from 0 to 4294967295. • The **min-bw** *kpbs* keyword argument combination specifies the minimum automatic bandwidth, in kbps, for this path option. The value can be from 0 to 4294967295. • The **collect-bw** keyword collects output rate information for the path option, but does not adjust the bandwidth of the path option.
Step 6	**bandwidth** [**sub-pool**| **global**] *kbps* **Example:** `Router(config-lsp-attr)# bandwidth 5000`	(Optional) Specifies LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option. • The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all path options are from the global pool in the absence of the **sub-pool** keyword. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295.

	Command or Action	Purpose
Step 7	**list** **Example:** `Router(config-lsp-attr)# list`	(Optional) Displays the contents of the LSP attribute list.
Step 8	**lockdown** **Example:** `Router(config-lsp-attr)# lockdown`	(Optional) Disables reoptimization of the LSP.
Step 9	**priority** *setup-priority* [*hold-priority*] **Example:** `Router(config-lsp-attr)# priority 1 1`	(Optional) Specifies the LSP priority. • The *setup-priority* argument is used when signaling an LSP to determine which existing LSPs can be preempted. Valid values are from 0 to 7, where a lower number indicates a higher priority. Therefore, an LSP with a setup priority of 0 can preempt any LSP with a non-0 priority. • The *hold-priority* argument is associated with an LSP to determine if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.
Step 10	**protection fast-reroute** **Example:** `Router(config-lsp-attr)# protection fast-reroute`	(Optional) Enables failure protection on the LSP.
Step 11	**record-route** **Example:** `Router(config-lsp-attr)# record-route`	(Optional) Records the route used by the LSP.
Step 12	**no** *sub-command* **Example:** `Router(config-lsp-attr)# no record-route`	(Optional) Removes a specific attribute from the LSP attributes list. • The *sub-command*argument names the LSP attribute to remove from the attributes list.
Step 13	**exit** **Example:** `Router(config-lsp-attr)# exit`	(Optional) Exits from LSP Attributes configuration mode.
Step 14	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.

Adding Attributes to an LSP Attribute List

Perform this task to add attributes to an LSP attribute list. The LSP attribute list provides a user interface that is flexible, easy to use, and that can be extended or changed at any time to meet the requirements of your MPLS TE tunnel traffic. LSP Attributes configuration mode is used to display the specific LSP attributes list and to add or change the required path option attribute.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng lsp attributes** *string*
4. **affinity** *value* [**mask***value*]
5. **bandwidth** [**sub-pool** | **global**] *kbps*
6. **priority** *setup-priority* [*hold-priority*]
7. **list**
8. **exit**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng lsp attributes** *string* **Example:** `Router(config)# mpls traffic-eng lsp attributes 1`	Configures an LSP Attribute list and enters LSP Attributes configuration mode. • The *string* argument identifies a specific LSP Attribute list.
Step 4	**affinity** *value* [**mask***value*] **Example:** `Router(config-lsp-attr)# affinity 0 mask 0`	(Optional) Specifies attribute flags for links comprising an LSP. • The *value* argument is a value required for links that make up an LSP. Values of the bits are either 0 or 1. • The **mask***value* keyword argument combination indicates which attribute values should be checked. • If a bit in the mask is 0, an attribute value of the link or that bit is irrelevant.

	Command or Action	Purpose
		• If a bit in the mask is 1, the attribute value of that link and the required affinity of the LSP for that bit must match.
Step 5	**bandwidth** [**sub-pool** | **global**] *kbps* **Example:** `Router(config-lsp-attr)# bandwidth 1000`	Specifies an LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option. • The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all path options are from the global pool in the absence of the **sub-pool** keyword. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295.
Step 6	**priority** *setup-priority* [*hold-priority*] **Example:** `Router(config-lsp-attr)# priority 2 2`	Specifies the LSP priority. • The *setup-priority* argument is used when signaling an LSP to determine which existing LSPs can be preempted. Valid values are from 0 to 7, where a lower number indicates a higher priority. Therefore, an LSP with a setup priority of 0 can preempt any LSP with a non-0 priority. • The *hold-priority* argument is associated with an LSP to determine if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.
Step 7	**list** **Example:** `Router(config-lsp-attr)# list`	(Optional) Displays the contents of the LSP attribute list. • Use the **list** command to display the path option attributes added to the attribute list.
Step 8	**exit** **Example:** `Router(config-lsp-attr)# exit`	(Optional) Exits LSP Attributes configuration mode.
Step 9	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.

Removing an Attribute from an LSP Attribute List

Perform this task to remove an attribute from an LSP attribute list. The LSP attributes list provides a means to easily remove a path option attribute that is no longer required for your MPLS TE tunnel traffic. LSP Attributes configuration mode is used to display the specific LSP attribute list and for the **no***sub-command*command, which is used to remove the specific attribute from the list. Replace the*sub-command* argument with the command that you want to remove from the list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng lsp attributes** *string*
4. **no** *sub-command*
5. **list**
6. **exit**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng lsp attributes** *string* **Example:** `Router(config)# mpls traffic-eng lsp attributes 1`	Configures an LSP Attribute list and enters LSP Attributes configuration mode. • The *string* argument identifies a specific LSP attribute list.
Step 4	**no** *sub-command* **Example:** `Router(config-lsp-attr)# no priority`	Removes a specific attribute from the LSP Attribute list. • The *sub-command*argument names the LSP attribute to remove from the attributes list.
Step 5	**list** **Example:** `Router(config-lsp-attr)# list`	(Optional) Displays the contents of the LSP attribute list. • Use the **list** command to verify that the path option attribute is removed from the attribute list.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-lsp-attr)# exit`	(Optional) Exits LSP Attributes configuration mode.
Step 7	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.

Modifying an Attribute in an LSP Attribute List

Perform this task to modify an attribute in an LSP attribute list. The LSP attribute list provides a flexible user interface that can be extended or modified an any time to meet the requirements of your MPLS TE tunnel traffic. LSP Attributes configuration mode is used to display the specific LSP attributes list and to modify the required path option attribute.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng lsp attributes** *string*
4. **affinity** *value* [**mask***value*]
5. **list**
6. **affinity** *value* [**mask***value*]
7. **list**
8. **exit**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng lsp attributes** *string* **Example:**	Configures an LSP Attribute list and enters LSP Attributes configuration mode.

	Command or Action	Purpose
	`Router(config)# mpls traffic-eng lsp attributes 1`	• The *string* argument identifies a specific LSP attribute list.

Step 4	**affinity** *value* [**mask***value*]
	Example:
	`Router(config-lsp-attr)# affinity 1 mask 1`
		Specifies attribute flags for links comprising an LSP.
		• The *value* argument is a value required for links comprising an LSP. Values of bits are either 0 or 1.
		• The **mask***value* keyword argument combination indicates which attribute values should be checked.
		  • If a bit in the mask is 0, an attribute value of the link or that bit is irrelevant.
		  • If a bit in the mask is 1, the attribute value of that link and the required affinity of the tunnel for that bit must match.

Step 5	**list** **Example:** `Router(config-lsp-attr)# list`	(Optional) Displays the contents of the LSP Attribute list. • Use the **list** command to display the path option attributes configured in the attribute list.

Step 6	**affinity** *value* [**mask***value*]
	Example:
	`Router(config-lsp-attr)# affinity 0 mask 0`
		Specifies attribute flags for links comprising an LSP.
		• The *value* argument is a value required for links comprising an LSP. Values of bits are either 0 or 1.
		• The **mask***value* keyword argument combination indicates which attribute values should be checked.
		  • If a bit in the mask is 0, an attribute value of the link or that bit is irrelevant.
		  • If a bit in the mask is 1, the attribute value of that link and the required affinity of the tunnel for that bit must match.

Step 7	**list** **Example:** `Router(config-lsp-attr)# list`	(Optional) Displays the contents of the LSP attribute list. • Use the **list** command to verify that the path option attributes is modified in the attribute list.
Step 8	**exit** **Example:** `Router(config-lsp-attr)# exit`	(Optional) Exits LSP Attributes configuration mode.
Step 9	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.

Deleting an LSP Attribute List

Perform this task to delete an LSP attribute list. You would perform this task when you no longer require the LSP attribute path options specified in the LSP attribute list for an MPLS TE tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no mpls traffic-eng lsp attributes** *string*
4. **end**
5. **show mpls traffic-eng lsp attributes** [*string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no mpls traffic-eng lsp attributes** *string* **Example:** `Router(config)# no mpls traffic-eng lsp attributes 1`	Removes a specified LSP Attribute list from the device configuration. • The *string* argument identifies the specific LSP attribute list to remove.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.
Step 5	**show mpls traffic-eng lsp attributes** [*string*] **Example:** `Router# show mpls traffic-eng lsp attributes`	(Optional) Displays information about configured LSP attribute lists. • Use the **show mpls traffic-eng lsp attributes** command to verify that the LSP attribute list was deleted from the router.

Verifying Attributes Within an LSP Attribute List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng lsp attributes** *string* **list**
4. **exit**
5. **end**

DETAILED STEPS

Step 1 enable

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 configure terminal

Use this command to enter global configuration mode. For example:

Example:

```
Router# configure terminal
Router(config)#
```

Step 3 mpls traffic-eng lsp attributes *string* list

Use this command to enter LSP Attributes configuration mode for a specific LSP attribute list and to verify that the contents of the attributes list are as expected. For example:

Example:

```
Router(config)# mpls traffic-eng lsp attributes 1 list
LIST 1
 bandwidth 1000
 priority 1 1
```

Step 4 exit

Use this command to exit LSP Attributes configuration mode. For example:

```
Router(config-lsp-attr)# exit
```

Example:

```
Router(config)#
```

Step 5 end

Use this command to exit to privileged EXEC mode. For example:

Example:

```
Router(config)# exit
Router#
```

Verifying All LSP Attribute Lists

Perform this task to verify all configured LSP attribute lists. Use this task to display all LSP attribute lists to verify that the attributes lists that you configured are in operation.

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng lsp attributes** *string* [**details**]
3. **show running-config** | **begin** *text-string*
4. **exit**

DETAILED STEPS

Step 1 enable

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 show mpls traffic-eng lsp attributes *string* [details]

Use this command to verify that all configured LSP attribute lists are as expected. For example:

Example:

```
Router# show mpls traffic-eng lsp attributes
LIST 1
  affinity 1 mask 1
  bandwidth 1000
  priority 1 1
LIST 2
  bandwidth 5000
LIST hipriority
  priority 0 0
!
```

Step 3 show running-config | begin *text-string*

Use this command to verify that all configured LSP attribute lists are as expected. Use the **begin** command modifier with the**mpls traffic-eng lsp** *text-string* to locate the LSP attributes information in the configuration file. For example:

Example:

```
Router# show running-config | begin mpls traffic-eng lsp
```

```
mpls traffic-eng lsp attributes 1
  affinity 1 mask 1
  bandwidth 1000
  priority 1 1
!
mpls traffic-eng lsp attributes 2
  bandwidth 5000
!
mpls traffic-eng lsp attributes hipriority
  priority 0 0
.
.
.
Router#
```

Step 4 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Associating an LSP Attribute List with a Path Option for an MPLS TE Tunnel

Perform this task to associate an LSP attribute list with a path option for an MPLS TE tunnel. This task is required if you want to apply the LSP attribute list that you configured to path options for your MPLS TE tunnels.

Based on your requirements, you can configure LSP attributes lists with different sets of attributes for different path options. LSP attribute lists also provide an easy way to configure multiple TE tunnels to use the same LSP attributes. That is, you can reference the same LSP attribute list to configure LSP-specific parameters for one or more TE tunnels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel destination** {*hostname* | *ip-address*}
5. **tunnel mode mpls traffic-eng**
6. **tunnel mpls traffic-eng autoroute announce**
7. **tunnel mpls traffic-eng bandwidth** [**sub-pool**| **global**] *bandwidth*
8. **tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*]
9. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*} [**verbatim**]} [**attributes** *string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface that you want to configure. • The *number* argument is the number of the tunnel interface that you want to create or configure.
Step 4	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option. • The *hostname* argument is the name of the host destination. • The *ip-address* argument is the IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the encapsulation mode for the tunnel for MPLS TE.
Step 6	**tunnel mpls traffic-eng autoroute announce** **Example:** `Router(config-if)# tunnel mpls traffic-eng autoroute announce`	Specifies that the IGP should use the tunnel (if the tunnel is up) in its enhanced shortest path first (SPF) calculation.
Step 7	**tunnel mpls traffic-eng bandwidth** [**sub-pool**| **global**] *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 1000`	Configures the bandwidth required for an MPLS TE tunnel and assigns it either to the subpool or the global pool. • The **sub-pool** keyword indicates a subpool tunnel. • The **global** keyword indicates a global pool tunnel. Entering this keyword is not necessary, for all tunnels are in the global pool in the absence of the **sub-pool** keyword.

	Command or Action	Purpose
		• The *kbps* argument is the bandwidth, in kilobits per second, set aside for the MPLS TE tunnel. The range is from 1 to 4294967295.
Step 8	**tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*] **Example:** `Router(config-if)# tunnel mpls traffic-eng priority 1 1`	Sets the priority to be used when the system determines which existing tunnels are eligible to be preempted. • The *setup-priority* argument is the priority used when signaling an LSP for this tunnel to determine which existing tunnels can be preempted. Valid values are from 0 to 7. A lower number indicates a higher priority. An LSP with a setup priority of 0 can preempt any LSP with a non-0 priority. • The *hold-priority* argument is the priority associated with an LSP for this tunnel to determine if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.
Step 9	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*} [**verbatim**]} [**attributes** *string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 dynamic attributes 1` **Example:**	Adds an LSP attribute list to specify LSP-related parameters for a path options for an MPLS TE tunnel. • The *number* argument identifies the path option. • The **dynamic** keyword indicates that the path option is dynamically calculated (the router figures out the best path). • The **explicit** keyword indicates that the path option is specified. You specify the IP addresses of the path. • The **name** *path-name*keyword argument combination identifies the name of the explicit path option. • The *path-number* argument identifies the number of the explicit path option. • The **verbatim** keyword bypasses the topology database verification. **Note** You can use the **verbatim** keyword only with the explicit path option. • The **attributes** *string* keyword argument combination names an attribute list to specify path options for the LSP. • The **bandwidth** keyword specifies LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option.

	Command or Action	Purpose
		• The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all path options are from the global pool in the absence of the **sub-pool** keyword. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295. • The **lockdown** keyword disables reoptimization of the LSP.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Modifying a Path Option to Use a Different LSP Attribute List

Perform this task to modify the path option to use a different LSP Attribute list.

Based on your requirements, you can configure LSP attributes lists with different sets of attributes for different path options or change the set of attributes associated with a path option. The **tunnel mpls traffic-eng path-option** *number* **dynamic attributes** *string* command is used in interface configuration mode to modify the path option to use a different LSP attribute list. The **attributes** and *string* keyword and argument names the new LSP attribute list for the path option specified.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel destination** {*hostname* | *ip-address*}
5. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Configures the interface type and enters interface configuration mode. • The *type* argument is the type of interface that you want to configure. • The *number* argument is the number of the tunnel interface that you want to create or configure.
Step 4	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option. • The *hostname* argument is the name of the host destination. • The *ip-address* argument is the IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 dynamic attributes 1`	Adds an LSP Attribute list to specify LSP-related parameters for a path options for an MPLS TE tunnel. • The *number* argument identifies the path option. • The **dynamic** keyword indicates that the path option is dynamically calculated (the router figures out the best path). • The **explicit** keyword indicates that the path option is specified. You specify the IP addresses of the path. • The **name***path-name*keyword argument combination identifies the name of the explicit path option. • The *path-number* argument identifies the number of the explicit path option. • The **verbatim** keyword bypasses the topology database verification. **Note** You can use the **verbatim** keyword only with the explicit path option. • The **attributes***string* keyword argument combination names an attribute list to specify path options for the LSP. • The **bandwidth** keyword specifies LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option. • The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all

	Command or Action	Purpose
		path options are from the global pool in the absence of the **sub-pool** keyword. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295. • The **lockdown** keyword disables reoptimization of the LSP.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Removing a Path Option for an LSP for an MPLS TE Tunnel

Perform this task to remove a path option for an LSP for an MPLS TE tunnel. Use this task to remove a path option for an LSP when your MPLS TE tunnel traffic requirements change.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel destination** {*hostname* | *ip-address*}
5. **no tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Configures the interface type and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface tunnel 1`	• The *type* argument is the type of interface that you want to configure. • The *number* argument is the number of the tunnel interface that you want to create or configure.
Step 4	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option. • The *hostname* argument is the name of the host destination. • The *ip-address* argument is the IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	**no tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**] **Example:** `Router(config-if)# no tunnel mpls traffic-eng path-option 1 dynamic attributes 1`	Removes an LSP Attribute list that specifies LSP-related parameters for a path option for an MPLS TE tunnel. • The *number* argument identifies the path option. • The **dynamic** keyword indicates that the path option is dynamically calculated (the router figures out the best path). • The **explicit** keyword indicates that the path option is specified. You specify the IP addresses of the path. • The **name***path-name*keyword argument combination identifies the name of the explicit path option. • The *path-number* argument identifies the number of the explicit path option. • The **verbatim** keyword bypasses the topology database verification. **Note** You can use the **verbatim** keyword only with the explicit path option. • The **attributes***string* keyword argument combination names an attribute list to specify path options for the LSP. • The **bandwidth** keyword specifies LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option. • The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all path options are from the global pool in the absence of the **sub-pool** keyword.

	Command or Action	Purpose
		• The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295. • The **lockdown** keyword disables reoptimization of the LSP.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Verifying that LSP Is Signaled Using the Correct Attributes

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**]
3. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**]

Use this command to verify that the LSP is signaled using the correct attributes for the specified tunnel. For example:

Example:

```
Router# show mpls traffic-eng tunnels tunnel1
Name: Router-t1                       (Tunnel1) Destination: 10.10.10.12
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 2, type explicit path2 (Basis for Setup, path weight 65834)
  Config Parameters:
    Bandwidth: 1000      kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: IGP (global)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 1        bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 2 is active
    BandwidthOverride: enabled   LockDown: disabled  Verbatim: disabled
    Bandwidth Override:
```

```
Signalling: 1        kbps (Global)
Overriding: 1000        kbps (Global) configured on tunnel
```

The output shows that the following attributes are signaled for tunnel tunnel1: affinity 0 mask 0, auto-bw disabled, bandwidth 1000, lockdown disabled, and priority 1 1.

Step 3 **exit**

Use this command to return to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuring a Path Option for Bandwidth Override

This section contains the following tasks for configuring a path option for bandwidth override:

Note Once you configure bandwidth as a path-option parameter, you can no longer configure an LSP Attribute list as a path-option parameter.

Configuring Fallback Bandwidth Path Options for TE Tunnels

Perform this task to configure fallback bandwidth path options for a TE tunnel. Use this task to configure path options that reduce the bandwidth constraint each time the headend of a tunnel fails to establish an LSP.

Configuration of the Path Option for Bandwidth Override feature can reduce bandwidth constraints on path options temporarily and improve the chances that an LSP is set up for the TE tunnel. When a TE tunnel uses a path option with bandwidth override, the traffic engineering software attempts every 30 seconds to reoptimize the tunnel to use the preferred path option with the original configured bandwidth. The Path Option for Bandwidth Override feature is designed as a temporary reduction in bandwidth constraint. To force immediate reoptimization of all traffic engineering tunnels, you can use the **mplstraffic-engreoptimize** command. You can also configure the **lockdown** command with bandwidth override to prevent automatic reoptimization.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel destination** {*hostname* | *ip-address*}
5. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface that you want to configure. • The *number* argument is the number of the tunnel interface that you want to create or configure.
Step 4	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option. • The *hostname* argument is the name of the host destination. • The *ip-address* argument is the IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 dynamic bandwidth 500`	Adds a Path Option for Bandwidth Override to specify a bandwidth fallback for a path option for an MPLS TE tunnel. • The *number* argument identifies the path option. • The **dynamic** keyword indicates that the path option is dynamically calculated (the router figures out the best path). • The **explicit** keyword indicates that the path option is specified. You specify the IP addresses of the path. • The **name***path-name*keyword argument combination identifies the name of the explicit path option. • The *path-number* argument identifies the number of the explicit path option. • The **verbatim** keyword bypasses the topology database verification.

	Command or Action	Purpose
		Note You can use the **verbatim** keyword only with the explicit path option. • The **attributes***string* keyword argument combination names an attribute list to specify path options for the LSP. • The **bandwidth** keyword specifies LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option. • The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all path options are from the global pool in the absence of the **sub-pool** keyword. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295. • The **lockdown** keyword disables reoptimization of the LSP.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Modifying the Bandwidth on a Path Option for Bandwidth Override

Perform this task to modify the bandwidth on a Path Option for Bandwidth Override. You might need to further reduce or modify the bandwidth constraint for a path option to ensure that the headend of a tunnel establishes an LSP.

The Path Option for Bandwidth Override feature is designed as a temporary reduction in bandwidth constraint. To force immediate reoptimization of all traffic engineering tunnels, you can use the **mplstraffic-engreoptimize** command. You can also configure the **lockdown** command with bandwidth override to prevent automatic reoptimization.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel destination** {*hostname* | *ip-address*}
5. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**]
6. **end**
7. **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Configures the interface type and enters interface configuration mode. • The *type* argument is the type of interface that you want to configure. • The *number* argument is the number of the tunnel interface that you want to create or configure.
Step 4	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option. • The *hostname* argument is the name of the host destination. • The *ip-address* argument is the IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name* | *path-number*} [**verbatim**]} [**attributes***string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 2 dynamic bandwidth 500` **Example:**	Adds a Path Option for Bandwidth Override to specify a bandwidth fallback for a path option for an MPLS TE tunnel. • The *number* argument identifies the path option. • The **dynamic** keyword indicates that the path option is dynamically calculated (the router figures out the best path). • The **explicit** keyword indicates that the path option is specified. You specify the IP addresses of the path. • The **name***path-name*keyword argument combination identifies the name of the explicit path option. • The *path-number* argument identifies the number of the explicit path option. • The **verbatim** keyword bypasses the topology database verification.

	Command or Action	Purpose
		Note You can use the **verbatim** keyword only with the explicit path option. • The **attributes***string* keyword argument combination names an attribute list to specify path options for the LSP. • The **bandwidth** keyword specifies LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option. • The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all path options are from the global pool in the absence of the **sub-pool** keyword. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295. • The **lockdown** keyword disables reoptimization of the LSP.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.
Step 7	**show mpls traffic-eng tunnels** *tunnel-interface* [**brief**] **Example:** `Router# show mpls traffic-eng tunnels tunnel1`	(Optional) Displays information about tunnels. • Use the **showmplstraffic-engtunnels** command to verify which bandwidth path option is in use by the LSP.

Removing a Path Option for Bandwidth Override

Perform this task to remove the bandwidth on the path option for bandwidth override. The Path Option for Bandwidth Override feature is designed as a temporary reduction in bandwidth constraint. Use this task to remove the bandwidth override when it is not required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel destination** {*hostname* | *ip-address*}
5. **no tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*} [**verbatim**]} [**attributes** *string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**]
6. **end**
7. **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Configures a tunnel interface type and enters interface configuration mode. • The *number* argument is the number of the tunnel interface that you want to create or configure.
Step 4	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option. • The *hostname* argument is the name of the host destination. • The *ip-address* argument is the IP address of the host destination expressed in decimal in four-part, dotted notation.
Step 5	**no tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*} [**verbatim**]} [**attributes** *string*] [**bandwidth** [**sub-pool** | **global**] *kbps*] [**lockdown**] **Example:** `Router(config-if)# no tunnel mpls traffic-eng path-option 2 dynamic bandwidth 500`	Removes a path option for bandwidth override that specifies a bandwidth fallback for a path option for an MPLS TE tunnel. • The *number* argument identifies the path option. • The **dynamic** keyword indicates that the path option is dynamically calculated (the router figures out the best path). • The **explicit** keyword indicates that the path option is specified. You specify the IP addresses of the path. • The **name** *path-name*keyword argument combination identifies the name of the explicit path option. • The *path-number* argument identifies the number of the explicit path option. • The **verbatim** keyword bypasses the topology database verification. **Note** You can use the **verbatim** keyword only with the explicit path option.

	Command or Action	Purpose
		• The **attributes** *string* keyword argument combination names an attribute list to specify path options for the LSP. • The **bandwidth** keyword specifies LSP bandwidth. • The **sub-pool** keyword indicates a subpool path option. • The **global** keyword indicates a global pool path option. Entering this keyword is not necessary, for all path options are from the global pool in the absence of the **sub-pool** keyword. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295. • The **lockdown** keyword disables reoptimization of the LSP.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.
Step 7	**show mpls traffic-eng tunnels** *tunnel-interface* [**brief**] **Example:** `Router# show mpls traffic-eng tunnels tunnel1`	(Optional) Displays information about tunnels. • Use the **show mpls traffic-eng tunnels** command to verify which bandwidth path option is in use by the LSP.

Verifying that LSP Is Signaled Using the Correct Bandwidth

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**]
3. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**]

Use this command to verify that the LSP is signaled with the correct bandwidth and to verify that the bandwidth configured on the tunnel is overridden. For example:

Example:

```
Router# show mpls traffic-eng tunnels tunnel21
Name: Router-t21                       (Tunnel21) Destination: 10.10.10.12
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 2, type explicit path2 (Basis for Setup, path weight 65834)
    path option 1, type explicit path1
  Config Parameters:
    Bandwidth: 1000      kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: IGP (global)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 1        bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 2 is active
    BandwidthOverride: enabled   LockDown: disabled  Verbatim: disabled
    Bandwidth Override:
      Signalling: 500 kbps (Global)
      Overriding: 1000        kbps (Global) configured on tunnel
```

If bandwidth override is actively being signaled, the **show mpls traffic-eng tunnel** command displays the bandwidth override information under the Active Path Option Parameters heading. The example shows that BandwidthOverride is enabled and that the tunnel is signaled using path-option 2. The bandwidth signaled is 500. This is the value configured on the path option 2 and it overrides the 1000 kbps bandwidth configured on the tunnel interface.

Step 3 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Troubleshooting Tips

If the tunnel state is down and you configured a path-option with bandwidth override enabled, the **showmplstraffic-engtunnels** command indicates other reasons why a tunnel is not established. For example:

- The tunnel destination is not in the routing table.
- If the bandwidth override value is not zero, the bandwidth constraint may still be too large.
- Other attributes configured on the tunnel, such as affinity, might prevent the calculation of a path over the existing topology.
- TE might not be configured on all links necessary to reach tunnel destination.

Configuration Examples for MPLS Traffic Engineering--RSVP Hello State Timer

Configuring LSP Attribute List Examples

Configuring an LSP Attribute List: Example

This example shows the configuration of the affinity, bandwidth, and priority LSP-related attributes in an LSP attribute list identified with the numeral 1:

```
Router(config)# mpls traffic-eng lsp attributes 1
Router(config-lsp-attr)# affinity 7 7
Router(config-lsp-attr)# bandwidth 1000
Router(config-lsp-attr)# priority 1 1
Router(config-lsp-attr)# exit
```

Adding Attributes to an LSP Attribute List: Example

This example shows the addition of protection attributes to the LSP attribute list identified with the numeral 1:

```
Router(config)# mpls traffic-eng lsp attributes 1
Router(config-lsp-attr)# affinity 7 7
Router(config-lsp-attr)# bandwidth 1000
Router(config-lsp-attr)# priority 1 1
Router(config-lsp-attr)# protection fast-reroute
Router(config-lsp-attr)# exit
```

Removing an Attribute from an LSP Attribute List: Example

The following example shows removing the priority attribute from the LSP attribute list identified by the string simple:

```
Router(config)# mpls traffic-eng lsp attributes simple
Router(config-lsp-attr)# priority 1 1
Router(config-lsp-attr)# list
LIST simple
 priority 1 1
!
Router(config-lsp-attr)# no priority
Router(config-lsp-attr)# list
LIST simple
!
Router(config-lsp-attr)# exit
```

Modifying an Attribute in an LSP Attribute List: Example

The following example shows modifying the bandwidth in an LSP attribute list identified by the numeral 5:

```
Router(config)# mpls traffic-eng lsp attributes 5
Router(config-lsp-attr)# bandwidth 1000
```

```
Router(config-lsp-attr)# priority 1 1
Router(config-lsp-attr)# list
LIST 5
 bandwidth 1000
 priority 1 1
Router(config-lsp-attr)# bandwidth 500
Router(config-lsp-attr)# list
LIST 5
 bandwidth 500
 priority 1 1
Router(config-lsp-attr)# exit
```

Deleting an LSP Attribute List: Example

The following example shows the deletion of an LSP attribute list identified by numeral 1:

```
Router(config)# mpls traffic-eng lsp attributes 1
Router(config-lsp-attr)# affinity 7 7
Router(config-lsp-attr)# bandwidth 1000
Router(config-lsp-attr)# priority 1 1

Router(config-lsp-attr)# exit
!
Router(config)# no mpls traffic-eng lsp attributes 1
```

Associating an LSP Attribute List with a Path Option for a TE Tunnel: Example

The following example associates the LSP attribute list identified by the numeral 3 with path option 1:

```
Router(config)# mpls traffic-eng lsp attributes 3
Router(config-lsp-attr)# bandwidth 1000
Router(config-lsp-attr)# priority 2 2
Router(config-lsp-attr)# protection fast-reroute
Router(config-lsp-attr)# exit
!
!
Router(config)# interface Tunnel 1
Router(config-if)# ip unnumbered FastEthernet1/0/1
Router(config-if)# tunnel destination 10.112.0.12
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng affinity 1
Router(config-if)# tunnel mpls traffic-eng bandwidth 5000
Router(config-if)# tunnel mpls traffic-eng path-option 1 dynamic attributes 3
```

In this configuration, the LSP will have the following attributes:

```
{bandwidth = 1000
 priority = 2 2
 affinity 1
 reroute enabled.
 }
```

The LSP attribute list referenced by the path option will take precedence over the values configured on the tunnel interface.

Modifying a Path Option to Use a Different LSP Attribute List: Example

The following example modifies path option 1 to use an LSP attribute list identified by the numeral 1:

```
Router(config)# mpls traffic-eng lsp attributes 1
Router(config-lsp-attr)# affinity 7 7
Router(config-lsp-attr)# bandwidth 500
Router(config-lsp-attr)# priority 1 1
Router(config-lsp-attr)# exit
Router(config)# mpls traffic-eng lsp attributes 2
Router(config-lsp-attr)# bandwidth 1000
Router(config-lsp-attr)# priority 1 1
Router(config-lsp-attr)# exit
Router(config)# interface Tunnel 1
Router(config-if)# ip unnumbered FastEthernet1/0/1
Router(config-if)# tunnel destination 10.112.0.12
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng affinity 1
Router(config-if)# tunnel mpls traffic-eng bandwidth 5000
Router(config-if)# tunnel mpls traffic-eng path-option 1 dynamic attributes 1
```

In this configuration, the LSP has the following attributes:

```
{affinity = 7 7
 bandwidth = 500
 priority = 1 1
 }
```

Removing a Path Option for an LSP for an MPLS TE Tunnel: Example

The following example shows the removal of path option 1 for an LSP for a TE tunnel:

```
Router(config)# interface Tunnel 1
Router(config-if)# ip unnumbered FastEthernet1/0/1
Router(config-if)# tunnel destination 10.112.0.12
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng affinity 1
Router(config-if)# tunnel mpls traffic-eng bandwidth 5000
Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit path1 attributes 1
Router(config-if)# tunnel mpls traffic-eng path-option 2 explicit path2 attributes 2
!
!
Router(config-if)# no tunnel mpls traffic-eng path-option 1 explicit path1 attributes 1
```

Configuring a Path Option for Bandwidth Override Examples

Configuring a Path Option to Override the Bandwidth: Example

The following examples show how to configure a path option to override the bandwidth:

```
Router(config-if)# tunnel mpls traffic-eng path-option 3 explicit name path1 ?
       attributes Specify an LSP attribute list
       bandwidth  override the bandwidth configured on the tunnel
       lockdown   not a candidate for reoptimization
       <cr>
Router(config-if)# tunnel mpls traffic-eng path-option 3 explicit name path1 bandwidth ?
       <0-4294967295>  bandwidth requirement in kbps
       sub-pool        tunnel uses sub-pool bandwidth
Router(config-if)# tunnel mpls traffic-eng path-option 3 explicit name path1 bandwidth 500
 ?
```

```
lockdown   not a candidate for reoptimization
<cr>
```

Note Once you configure bandwidth as a path-option parameter, you can no longer configure an LSP attribute list as a path-option parameter.

Configuring Fallback Bandwidth Path Options for TE Tunnels: Example

The following example shows multiple path options configured with the **tunnel mpls traffic-eng path-option** command:

```
interface Tunnel 1
 ip unnumbered Loopback0
 tunnel destination 10.10.10.12
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name path1
 tunnel mpls traffic-eng path-option 2 explicit name path2 bandwidth 500
 tunnel mpls traffic-eng path-option 3 dynamic bandwidth 0
end
```

The device selects a path option for an LSP in order of preference, as follows:

- The device attempts to signal an LSP using path options starting with path-option 1.

The device attempts to signal an LSP with the 1000 kbps bandwidth configured on the tunnel interface because path-option 1 has no bandwidth configured.

- If 1000 kbps bandwidth is not available over the network, the device attempts to establish an LSP using path-option 2.

Path-option 2 has a bandwidth of 500 kbps configured. This reduces the bandwidth constraint from the original 1000 kbps configured on the tunnel interface.

- If 500 kbps is not available, the device attempts to establish an LSP using path-option 3.

Path-option 3 is configured as dynamic and has bandwidth 0. The device establishes the LSP if an IP path exists to the destination and all other tunnel constraints are met.

Modifying the Bandwidth on a Path Option for Bandwidth Override: Example

The following example shows modifying the bandwidth on a path option for bandwidth override. Path-option 3 is changed to an explicit path with a bandwidth of 100 kbps. Path-option 4 is configured with bandwidth 0.

```
interface Tunnel 1
 ip unnumbered Loopback0
 tunnel destination 10.10.10.12
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name path1
 tunnel mpls traffic-eng path-option 2 explicit name path2 bandwidth 500
```

```
 tunnel mpls traffic-eng path-option 3 dynamic bandwidth 0
!
!
Router(config)# tunnel mpls traffic-eng path-option 3 explicit name path3 bandwidth 100
Router(config)# tunnel mpls traffic-eng path-option 4 dynamic bandwidth 0
```

Removing the Path Option Bandwidth Value for an LSP for an MPLS TE Tunnel: Example

The following example shows the removal of the bandwidth for path option 3 for an LSP for an MPLS TE tunnel:

```
interface Tunnel 1
 ip unnumbered Loopback0
 tunnel destination 10.10.10.12
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 1 explicit name path1
 tunnel mpls traffic-eng path-option 2 explicit name path2 bandwidth 500
 tunnel mpls traffic-eng path-option 3 explicit name path3 bandwidth 100
 tunnel mpls traffic-eng path-option 4 dynamic bandwidth 0
!
Router(config)# no tunnel mpls traffic-eng path-option 3 explicit name path3 bandwidth 100
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS TE command descriptions	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering LSP Attributes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 166: Feature Information for MPLS Traffic Engineering LSP Attributes

Feature Name	Releases	Feature Information
MPLS Traffic Engineering LSP Attributes	Cisco IOS XE Release 2.3	This document describes how to configure label switched path (LSP) attributes for path options associated with Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels. The MPLS Traffic Engineering--LSP Attributes feature is an extension to MPLS TE that provides an LSP Attribute List feature and a Path Option for Bandwidth Override feature. These features provide flexibility in the configuration of LSP attributes for MPLS TE tunnel path options. Several LSP attributes can be applied to path options for TE tunnels using an LSP attribute list. If bandwidth is the only LSP attribute you require, then you can configure a Path Option for Bandwidth Override. The following commands were introduced or modified: **affinity** (LSP Attributes), **bandwidth**(LSP Attributes), **exit**(LSP Attributes), **list**(LSP Attributes), **lockdown**(LSP Attributes), **mpls traffic-eng lsp attributes**, **priority**(LSP Attributes), **protection**(LSP Attributes), **record-route**(LSP Attributes), **show mpls traffic-eng lsp attributes**, and **show mpls traffic-eng tunnels**.

Glossary

bandwidth --The difference between the highest and lowest frequencies available for network signals. The term also is used to describe the rated throughput capacity of a given network medium or protocol. The frequency range necessary to convey a signal measured in units of hertz (Hz). For example, voice signals typically require approximately 7 kHz of bandwidth and data traffic typically requires approximately 50 kHz of bandwidth.

bandwidth reservation --The process of assigning bandwidth to users and applications served by a network. This process involves assigning priority to different flows of traffic based on how critical and delay-sensitive they are. This makes the best use of available bandwidth, and if the network becomes congested, lower-priority traffic can be dropped. Sometimes called bandwidth allocation

global pool --The total bandwidth allocated to an Multiprotocol Label Switching (MPLS) traffic engineering link.

label switched path (LSP) tunnel --A configured connection between two routers, using label switching to carry the packets.

LSR --label switch router. A Multiprotocol Label Switching (MPLS) node that can forward native Layer 3 packets. The LSR forwards a packet based on the value of a label attached to the packet.

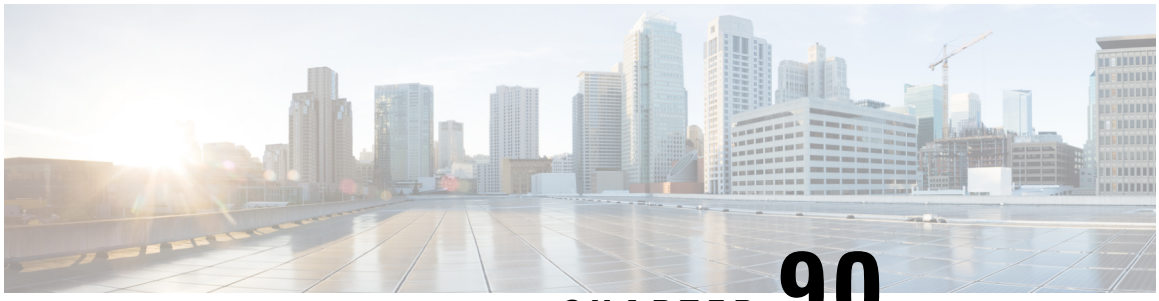
MPLS TE --Multiprotocol Label Switching (MPLS) traffic engineering (formerly known as “RRR” or Resource Reservation Routing). The use of label switching to improve traffic performance along with an efficient use of network resources.

subpool --The more restrictive bandwidth in an Multiprotocol Label Switching (MPLS) traffic engineering link. The subpool is a portion of the link's overall global pool bandwidth.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used. The application of scientific principles and technology to measure, model, and control internet traffic in order to simultaneously optimize traffic performance and network resource utilization.

traffic engineering tunnel --A label-switched tunnel used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.

tunnel --A secure communication path between two peers, such as two routers.



CHAPTER 90

MPLS Traffic Engineering AutoTunnel Mesh Groups

The MPLS Traffic Engineering Autotunnel Mesh Groups feature allows a network administrator to configure traffic engineering (TE) label switched paths (LSPs) by using a few command-line interface (CLI) commands.

In a network topology where edge TE label switch routers (LSRs) are connected by core LSRs, the MPLS Traffic Engineering--Autotunnel Mesh Groups feature automatically constructs a mesh of TE LSPs among the provider edge (PE) devices.

Prerequisites for MPLS Traffic Engineering--AutoTunnel Mesh Groups

- Be knowledgeable about MPLS TE.
- Decide how you will set up autotunnels (that is, identify the tunnel commands that you will include in the template interface).
- Identify a block of addresses that you will reserve for mesh tunnel interfaces.

Restrictions for MPLS Traffic Engineering--AutoTunnel Mesh Groups

- Mesh groups do not support interarea tunnels because the destinations of those tunnels do not exist in the local area TE database.

- You cannot configure a static route to route traffic over autotunnel mesh group TE tunnels. You should use only the autoroute for tunnel selection.
- Intermediate System-to-System (IS-IS) does not support Interior Gateway Protocol (IGP) distribution of mesh group information. For IS-IS, only Access Control Lists (ACLs) can be used.

Information About MPLS Traffic Engineering--AutoTunnel Mesh Groups

AutoTunnel Mesh Groups Description and Benefits

An autotunnel mesh group (referred to as a mesh group) is a set of connections between edge LSRs in a network. There are two types of mesh groups:

- Full--All the edge LSRs are connected. Each PE device has a tunnel to each of the other PE devices.
- Partial--Some of the edge LSRs are not connected to each other by tunnels.

In a network topology where edge TE LSRs are connected by core LSRs, the MPLS Traffic Engineering--Autotunnel Mesh Groups feature automatically constructs a mesh of TE LSPs among the PE devices.

Initially, you must configure each existing TE LSR to be a member of the mesh by using a minimal set of configuration commands. When the network grows (that is, when one or more TE LSRs are added to the network as PE devices), you do not need to reconfigure the existing TE LSR members of that mesh.

Mesh groups have the following benefits:

- Minimize the initial configuration of the network. You configure one template interface per mesh, and it propagates to all mesh tunnel interfaces, as needed.
- Minimize future configurations resulting from network growth. The feature eliminates the need to reconfigure each existing TE LSR to establish a full mesh of TE LSPs whenever a new PE device is added to the network.
- Enable existing devices to configure TE LSPs to new PE devices.
- Enable the construction of a mesh of TE LSPs among the PE devices automatically.

Access Lists for Mesh Tunnel Interfaces

The access list determines the destination addresses for the mesh tunnel interfaces. It is useful if you preallocate a block of related IP addresses. You can use that block of addresses to control the PE devices to which a full or partial mesh of TE tunnel LSPs is established. The access list allows matches for only the addresses that are learned and stored in the TE topology database.

For example, you can create an access list that matches all 10.1.1.1 IP addresses. You configure a template with the access list, then the template creates mesh tunnel interfaces to destinations within the TE topology database that match destinations in that access list.

Whenever the TE topology database is updated (for example, when a new TE LSR is inserted into the Interior Gateway Protocol (IGP), the destination address is stored in the TE topology database of each device in the IGP. At each update, the Mesh Group feature compares the destination address contained in the database to IP addresses in the access list associated with all template interfaces. If there is a match, the Mesh Group feature establishes a mesh tunnel interface to the tunnel destination IP address.

AutoTunnel Template Interfaces

An autotunnel template interface is a logical entity; that is, it is a configuration for a tunnel interface that is not tied to specific tunnel interfaces. It can be applied dynamically, when needed.

Mesh tunnel interfaces are tunnel interfaces that are created, configured dynamically (for example, by the applying [or cloning] of a template interface), used, and then freed when they are no longer needed.

A mesh tunnel interface obtains its configuration information from a template, except for the tunnel's destination address, which it obtains from the TE topology database that matches an access list or from the IGP mesh group advertisement.

The template interface allows you to enter commands once per mesh group. These commands specify how mesh tunnel interfaces are created. Each time a new device is added to the network, a new mesh tunnel interface is created. The configuration of the interface is duplicated from the template. Each mesh tunnel interface has the same path constraints and other parameters configured on the template interface. Only the tunnel destination address is different.

OSPF Flooding of Mesh Group Information

For OSPF to advertise or flood mesh group information, you need to configure a mesh group in OSPF and add that mesh group to an autotemplate interface. When the configuration is complete, OSPF advertises the mesh group IDs to all LSRs. MPLS TE LSPs automatically connect the edge LSRs in each mesh group.

OSPF can advertise mesh group IDs for an OSPF area. OSPF is the only IGP supported in some software releases of the MPLS Traffic Engineering--Autotunnel Mesh Groups feature.

How to Configure MPLS Traffic Engineering--AutoTunnel Mesh Groups

Configuring a Mesh of TE Tunnel LSPs

Perform the following tasks on each PE device in your network to configure a mesh of TE tunnel LSPs:

Note You can perform these tasks in any order.

Enabling Autotunnel Mesh Groups Globally

Perform this task on all PE devices in your network that you want to be part of an autotunnel mesh group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel mesh**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng auto-tunnel mesh** **Example:** `Device(config)# mpls traffic-eng auto-tunnel mesh`	Enables autotunnel mesh groups globally.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Creating an Access List Using a Name

The access list determines the destination addresses for the mesh tunnel interfaces. You can use an access list to control the PE devices to which a full or partial mesh of TE tunnel LSPs is established. The access list allows matches for only the addresses that are learned and stored in the TE topology database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} *access-list-name*
4. **permit** *source* [*source-wildcard*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Device(config)# ip access-list standard a1`	Defines an IP access list using a name and enters standard named access list configuration mode. • The **standard** keyword specifies a standard IP access list. • The **extended** keyword specifies an extended IP access list. • The *access-list-name* argument is the name of the access list. A name cannot contain a space or quotation mark and must begin with an alphabetic character. This prevents confusion with numbered access lists.

Step 4	**permit** *source* [*source-wildcard*]
	Example:
	`Device(config-std-nacl)# permit 10.0.0.0 0.255.255.255`
		Sets conditions to allow a packet to pass a named IP access list.
		• The *source* argument is the number of the network or host from which the packet is being sent. There are three alternative ways to specify the source:
		  • Use a 32-bit quantity in four-part dotted decimal format.
		  • Use the **any** keyword as an abbreviation for a source and source-wildcard of 0.0.0.0 255.255.255.255.
		  • Use host source as an abbreviation for a source and source-wildcard of source 0.0.0.0.
		• The *source-wildcard* argument is the wildcard bits to be applied to source. There are three alternative ways to specify the source wildcard:
		  • Use a 32-bit quantity in four-part dotted decimal format. Place 1s in the bit positions you want to ignore.
		  • Use the **any** keyword as an abbreviation for a source and source-wildcard of 0.0.0.0 255.255.255.255.
		  • Use host source as an abbreviation for a source and source-wildcard of source 0.0.0.0.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-std-nacl)# end`	Exits to privileged EXEC mode.

Creating an Autotunnel Template Interface

Creating an autotunnel template interface helps minimize the initial configuration of the network. You configure one template interface per mesh, which propagates to all mesh tunnel interfaces, as needed.

Note You can use the following commands to create a minimal configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface auto-template** *interface-num*
4. **ip unnumbered** *interface-type interface-number*
5. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **ipip** | **iptalk** | **mpls** | **nos**}
6. **tunnel mpls traffic-eng autoroute announce**
7. **tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*]
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}} [**lockdown**]
9. **tunnel destination access-list** *num*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface auto-template** *interface-num* **Example:** `Device(config)# interface auto-template 1`	Creates a template interface and enters interface configuration mode. • The *interface-num* argument is the interface number. Valid values are from 1 to 25.

	Command or Action	Purpose
Step 4	**ip unnumbered** *interface-type interface-number* **Example:** `Device(config-if)# ip unnumbered Loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface. • The *type* and *number* arguments name the type and number of another interface on which the device has an assigned IP address. It cannot be another unnumbered interface.
Step 5	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **ipip** | **iptalk** | **mpls** | **nos**} **Example:** `Device(config-if)# tunnel mode mpls`	Sets the encapsulation mode for the tunnel interface.
Step 6	**tunnel mpls traffic-eng autoroute announce** **Example:** `Device(config-if)# tunnel mpls traffic-eng autoroute announce`	Specifies that the IGP should use the tunnel (if the tunnel is up) in its enhanced shortest path first algorithm (SPF) calculation.
Step 7	**tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*] **Example:** `Device(config-if)# tunnel mpls traffic-eng priority 1 1`	Configures the setup and reservation priority for an MPLS TE tunnel. • The *setup-priority* argument is the priority used when an LSP is signaled for this tunnel and determines which existing tunnels can be preempted. Valid values are from 0 to 7, where a lower number indicates a higher priority. Therefore, an LSP with a setup priority of 0 can preempt any LSP with a non-0 priority. • The *hold-priority* argument is the priority associated with an LSP for this tunnel and determines if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}} [**lockdown**] **Example:** `Device(config-if)# tunnel mpls traffic-eng path-option 1 dynamic`	Configures a path option for an MPLS TE tunnel. • The *number* argument is the number of the path option. When multiple path options are configured, lower numbered options are preferred. • The **dynamic** keyword specifies that the path of the LSP is dynamically calculated. • The **explicit** keyword specifies that the path of the LSP is an IP explicit path. • The **name** *path-name* keyword-argument pair is the path name of the IP explicit path that the tunnel uses with this option.

	Command or Action	Purpose
		• The *path-number* argument is the path number of the IP explicit path that the tunnel uses with this option. • The **lockdown** keyword specifies that the LSP cannot be reoptimized.
Step 9	**tunnel destination access-list** *num* **Example:** `Device(config-if)# tunnel destination access-list 1`	Specifies the access list that the template interface uses for obtaining the mesh tunnel interface destination address. • The *num* argument is the number of the access list.
Step 10	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Specifying the Range of Mesh Tunnel Interface Numbers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel mesh tunnel-num min** *num* **max** *num*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng auto-tunnel mesh tunnel-num min** *num* **max** *num* **Example:** `Device(config)# mpls traffic-eng auto-tunnel mesh tunnel-num min 1000 max 2000`	Specifies the range of mesh tunnel interface numbers. • The **min** *num* keyword-argument pair specifies the beginning number of the range of mesh tunnel interface numbers. Valid values are from 1 to 65535.

	Command or Action	Purpose
		• The **max** *num* keyword-argument pair specifies the ending number of the range of mesh tunnel interface numbers. Valid values are from 1 to 65535.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Displaying Configuration Information About Tunnels

SUMMARY STEPS

1. **enable**
2. **show running interface auto-template** *num*
3. **show interface tunnel** *num* **configuration**
4. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Device> enable
Device#
```

Step 2 **show running interface auto-template** *num*

Use this command to display interface configuration information for a tunnel interface. For example:

Example:

```
Device# show running interface auto-template 1
interface auto-template1
 ip unnumbered Loopback0
 no ip directed-broadcast
 no keepalive
 tunnel destination access-list 1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
```

This output shows that autotunnel template interface auto-template1 uses an access list (access-list 1) to determine the destination addresses for the mesh tunnel interfaces.

Step 3 **show interface tunnel** *num* **configuration**

Use this command to display the configuration of the mesh tunnel interface. For example:

Example:

```
Device# show interface tunnel 5 configuration
interface tunnel 5
 ip unnumbered Loopback0
 no ip directed-broadcast
 no keepalive
 tunnel destination access-list 1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
```

Step 4 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Device# exit
Device>
```

Monitoring the Autotunnel Mesh Network

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng tunnels property auto-tunnel mesh** [**brief**]
3. **show mpls traffic-eng auto-tunnel mesh**
4. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Device> enable
Device#
```

Step 2 **show mpls traffic-eng tunnels property auto-tunnel mesh** [**brief**]

Use this command to monitor mesh tunnel interfaces. This command restricts the output of the **show mpls traffic-eng tunnels** command to display only mesh tunnel interfaces. For example:

Example:

```
Device# show mpls traffic-eng tunnels property auto-tunnel mesh brief
Signalling Summary:
 LSP Tunnels Process:            running
 RSVP Process:                   running
 Forwarding:                     enabled
```

```
 Periodic reoptimization:        every 3600 seconds, next in 491 seconds
 Periodic FRR Promotion:         Not Running
 Periodic auto-bw collection:    disabled
TUNNEL NAME                      DESTINATION    UP IF     DOWN IF
STATE/PROT
device_t64336                    10.2.2.2        -        Se2/0
up/up
device_t64337                    10.3.3.3        -        Se2/0
up/up
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Step 3 **show mpls traffic-eng auto-tunnel mesh**

Use this command to display the cloned mesh tunnel interfaces of each autotemplate interface and the current range of mesh tunnel interface numbers. For example:

Example:

```
Device# show mpls traffic-eng auto-tunnel mesh
Auto-Template1:
 Using access-list 1 to clone the following tunnel interfaces:
  Destination  Interface
  -----------  ---------
  10.2.2.2     Tunnel64336
  10.3.3.3     Tunnel64337
Mesh tunnel interface numbers: min 64336 max 65337
```

Step 4 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Device# exit
Device>
```

Troubleshooting Tips

You can configure mesh tunnel interfaces directly. However, you cannot delete them manually, and manual configuration is not permanent. The configuration is overwritten when the template changes or the mesh tunnel interface is deleted and re-created. If you attempt to manually delete a mesh tunnel interface, an error message appears.

You can enter the **show mpls traffic-eng tunnels destination** *address* command to display information about tunnels that are destined for a specified IP address.

Enter the **show mpls traffic-eng tunnels property auto-tunnel mesh** command to display information about mesh tunnel interfaces.

Configuring IGP Flooding for Autotunnel Mesh Groups

Perform the following task to configure IGP flooding for autotunnel mesh groups. Use this task to configure an OSPF-based discovery for identifying mesh group members and advertising the mesh group IDs to all LSRs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel mesh**
4. **router ospf** *process-id*
5. **mpls traffic-eng mesh-group** *mesh-group-id* *interface-type interface-number* **area** *area-id*
6. **exit**
7. Repeat steps 4 and 5 at other LSRs to advertise the mesh group numbers to which they belong.
8. **interface auto-template** *interface-num*
9. **tunnel destination mesh-group** *mesh-group-id*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng auto-tunnel mesh** **Example:** `Device(config)# mpls traffic-eng auto-tunnel mesh`	Enables autotunnel mesh groups globally.
Step 4	**router ospf** *process-id* **Example:** `Device(config)# router ospf 100`	Enters router configuration mode and configures an OSPF routing process. • The *process-id*argument is an internally used identification parameter for an OSPF routing process. It is locally assigned and can be any positive integer. A unique value is assigned for each OSPF routing process.
Step 5	**mpls traffic-eng mesh-group** *mesh-group-id* *interface-type interface-number* **area** *area-id* **Example:** `Device(config-router)# mpls traffic-eng mesh-group 10 loopback 0 area 100`	Advertises the autotunnel mesh group number of an LSR. • The *mesh-group-id* is a number that identifies a specific mesh group. • The *interface-type* and *interface-number* arguments specify a type of interface and an interface number. • The **area** *area-id* keyword-argument pair identifies the area.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits to global configuration mode.
Step 7	Repeat steps 4 and 5 at other LSRs to advertise the mesh group numbers to which they belong.	--
Step 8	**interface auto-template** *interface-num* **Example:** `Device(config)# interface auto-template 1`	Creates a template interface and enters interface configuration mode. • The *interface-num* argument identifies the interface number. Valid values are from 1 to 25.
Step 9	**tunnel destination mesh-group** *mesh-group-id* **Example:** `Device(config-if)# tunnel destination mesh-group 10`	Specifies a mesh group that a template interface uses to signal tunnels for all mesh group members. • The *mesh-group-id* is a number that identifies a specific mesh group.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuration Examples for MPLS Traffic Engineering--Autotunnel Mesh Groups

Examples: Configuring a Mesh of TE Tunnel LSPs

This section contains the following configuration examples for configuring a mesh of TE tunnel LSP:

Example: Enabling Autotunnel Mesh Groups Globally

The following example shows how to enable autotunnel mesh groups globally:

```
configure terminal
!
mpls traffic-eng auto-tunnel mesh
end
```

Example: Creating an Access List Using a Name

The following examples shows how to create an access list using a name to determine the destination addresses for the mesh tunnel interfaces:

```
configure terminal
!
ip access-list standard a1
 permit 10.0.0.0 0.255.255.255
end
```

In this example, any IP address in the TE topology database that matches access list a1 causes the creation of a mesh tunnel interface with that destination address.

Example: Creating an AutoTunnel Template Interface

This example shows how to create an AutoTunnel template interface. In the following example, an AutoTunnel template is created and configured with a typical set of TE commands. The mesh group created from the template consists of mesh tunnel interfaces with destination addresses that match access list a1.

Note The following example shows a typical configuration.

```
configure terminal
!
interface auto-template 1
 ip unnumbered Loopback0
 tunnel mode mpls
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1

 tunnel mpls traffic-eng path-option 1 dynamic
 tunnel destination access-list a1
 end
```

Example: Specifying the Range of Mesh Tunnel Interface Numbers

In the following example, the lowest mesh tunnel interface number can be 1000, and the highest mesh tunnel interface number can be 2000:

```
configure terminal
!
mpls traffic-eng auto-tunnel mesh tunnel-num min 1000 max 2000
end
```

Example: Configuring IGP Flooding for Autotunnel Mesh Groups

In the following example, OSPF is configured to advertise the device membership in mesh group 10:

```
configure terminal
!
mpls traffic-eng auto-tunnel mesh
router ospf 100
 mpls traffic-eng mesh-group 10 loopback 0 area 100
 exit
!
interface auto-template 1
 tunnel destination mesh-group 10
 end
```

Additional References

Related Documents

Related Topic	Document Title
MPLS traffic engineering command descriptions	*Multiprotocol Label Switching Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering--Autotunnel Mesh Groups

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 167: Feature Information for MPLS Traffic Engineering--Autotunnel Mesh Groups

Feature Name	Releases	Feature Information
MPLS Traffic Engineering--Autotunnel Mesh Groups	12.0(27)S 12.0(29)S 12.2(33)SRA 12.2(33)SXH 12.4(20)T 12.2(33)SRE Cisco IOS XE Release 3.6S	The MPLS Traffic Engineering--AutoTunnel Mesh Groups feature allows a network administrator to configure TE LSPs. In Cisco IOS Release 12.2(27)S, this feature was introduced. In Cisco IOS Release 12.0(29)S, this feature was updated to include Interior Gateway Protocol (IGP) flooding of autotunnel mesh groups. In Cisco IOS Release 12.2(33)SRA, this feature was integrated. In Cisco IOS Release 12.2(33)SXH, support was added. In Cisco IOS Release 12.4(20)T, this feature was integrated. In Cisco IOS Release 12.2(33)SRE, this feature was integrated. A device with autotunnel mesh groups can be configured with stateful switchover (SSO) redundancy. In Cisco IOS XE Release 3.6S, this feature was integrated. These commands were introduced or modified: **mpls traffic-eng auto-tunnel mesh**, **mpls traffic-eng auto-tunnel mesh tunnel-num**, **mpls traffic-eng mesh-group**, **show mpls traffic-eng auto-tunnel mesh**.
MPLS TE--Autotunnel/Auotmesh SSO Coexistence	Cisco IOS XE Release 3.5S 15.0(1)S	In Cisco IOS XE Release 3.5S, this feature was integrated. In Cisco IOS Release 15.0(1)S, this feature was integrated. **Note** Starting with Cisco IOS Release 15.2(2)S and Cisco IOS XE Release 3.6S, the SSO Support for MPLS TE Autotunnel and Automesh feature replaces the MPLS TE - Autotunnel/Automesh SSO Coexistence feature. For more information, see the *MPLS High Availability Configuration Guide* for the new implementation.

Glossary

CE device --customer edge device. A device that is part of a customer's network and interfaces to a provider edge (PE) device.

customer network --A network that is under the control of an end customer. Private addresses can be used in a customer network. Customer networks are logically isolated from each other and from the service provider's network.

edge device --A device at the edge of the network that receives and transmits packets. It can define the boundaries of the Multiprotocol Label Switching (MPLS) network.

headend --The label switch router (LSR) where a tunnel originates. The tunnel's “head” or tunnel interface resides at this LSR as well.

label --A short, fixed-length data construct that tells switching nodes how to forward data (packets).

label switched path (LSP) tunnel --A configured connection between two devices in which label switching is used to carry the packets.

LSP --label switched path. A path that a labeled packet follows over several hops, starting at an ingress LSR and ending at an egress LSR.

LSR --label switch router. A Layer 3 device that forwards a packet based on the value of a label encapsulated in the packet.

mesh group --A set of label switch devices (LSRs) that are members of a full or partial network of traffic engineering (TE) label switched paths (LSPs).

P device --provider core device.

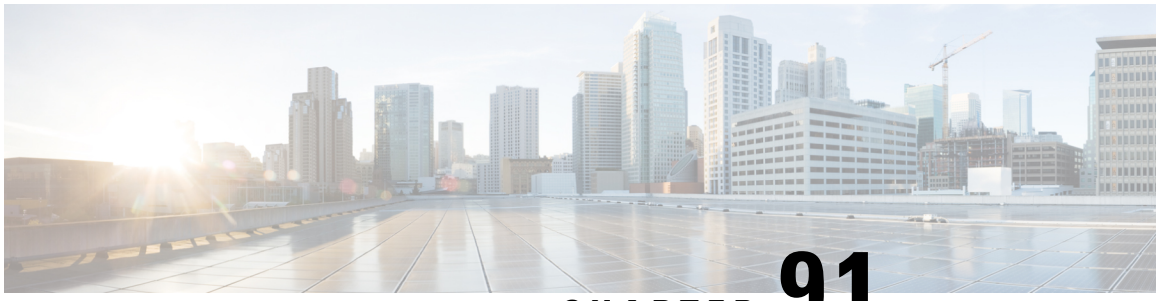
PE device --provider edge device. A device at the edge of the service provider’s network that interfaces to customer edge (CE) devices.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

tailend --The downstream, receive end of a tunnel.

traffic engineering --The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel --A secure communication path between two peers, such as two devices. A traffic engineering tunnel is a label switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing.



CHAPTER 91

MPLS Traffic Engineering Verbatim Path Support

The MPLS Traffic Engineering--Verbatim Path Support feature allows network nodes to support Resource Reservation Protocol (RSVP) extensions without supporting Interior Gateway Protocol (IGP) extensions for traffic engineering (TE), thereby bypassing the topology database verification process.

Prerequisites for MPLS Traffic Engineering--Verbatim Path Support

- A Multiprotocol Label Switching (MPLS) TE tunnel must be configured globally.
- MPLS TE must be enabled on all links.

Restrictions for MPLS Traffic Engineering Verbatim Path Support

- The **verbatim** keyword can be used only on a label-switched path (LSP) that is configured with the explicit path option.
- This release does not support reoptimization on the verbatim LSP.

Information About MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

MPLS TE Verbatim Path Support Overview

MPLS TE LSPs usually require that all the nodes in the network are TE aware, meaning they have IGP extensions to TE in place. However, some network administrators want the ability to build TE LSPs to traverse nodes that do not support IGP extensions to TE, but that do support RSVP extensions to TE.

Verbatim LSPs are helpful when all or some of the intermediate nodes in a network do not support IGP extensions for TE.

When this feature is enabled, the IP explicit path is not checked against the TE topology database. Because the TE topology database is not verified, a Path message with IP explicit path information is routed using the shortest path first (SPF) algorithm for IP routing.

How to Configure MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

Configuring MPLS Traffic Engineering--Verbatim Path Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered loopback** *number*
5. **tunnel destination** {*host-name*| *ip-address*}
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** {**sub-pool** *kbps* | *kbps}*
8. **tunnel mpls traffic-eng autoroute announce**
9. **tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*]
10. **tunnel mpls traffic-eng path-option** *preference-number* {**dynamic** [**attributes** *string* | **bandwidth** {**sub-pool** *kbps* | *kbps*} **| lockdown | verbatim**] | **explicit**{**name** *path-name* | **identifier** *path-number* **}}**
11. **exit**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode. • The *number* argument identifies the tunnel number to be configured.

Step 4	**ip unnumbered loopback** *number*
	Example:
	`Router(config-if)`
	`# ip unnumbered loopback 1`
		Configures an unnumbered IP interface, which enables IP processing without an explicit address. A loopback interface is usually configured with the router ID.
		Note An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.

Step 5	**tunnel destination** {*host-name*| *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.100.100.100`	Specifies the destination for a tunnel. • The *host-name* argument is the name of the host destination. • The *ip-address* argument is the IP Version 4 address of the host destination expressed in decimal in four-part, dotted notation.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** {**sub-pool** *kbps* | *kbps}* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 1000`	Configures the bandwidth required for an MPLS TE tunnel and assigns it either to the sub-pool or the global pool. • The **sub-pool** keyword indicates a subpool tunnel. • The *kbps* argument is the bandwidth, in kilobits per second, set aside for the MPLS TE tunnel. The range is from 1 to 4294967295.
Step 8	**tunnel mpls traffic-eng autoroute announce** **Example:** `Router(config-if)# tunnel mpls traffic-eng autoroute announce`	Specifies that IGP should use the tunnel (if the tunnel is up) in its enhanced SPF calculation.

	Command or Action	Purpose
Step 9	**tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*] **Example:** `Router(config-if)# tunnel mpls traffic-eng priority 1 1`	Configures setup and reservation priority for a tunnel. • The *setup-priority* argument is the priority used when signaling an LSP for this tunnel to determine which existing tunnels can be preempted. Valid values are from 0 to 7. A lower number indicates a higher priority. An LSP with a setup priority of 0 can preempt any LSP with a non-0 priority. • The *hold-priority* argument is the priority associated with an LSP for this tunnel to determine if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.
Step 10	**tunnel mpls traffic-eng path-option** *preference-number* {**dynamic** [**attributes** *string* | **bandwidth** {**sub-pool** *kbps* | *kbps*} | **lockdown** | **verbatim**] | **explicit** {**name** *path-name* | **identifier** *path-number* **}}** **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit name test verbatim` **Example:**	Specifies LSP-related parameters, including the **verbatim** keyword used with an explicit path option, for an MPLS TE tunnel. • The *preference-number* argument identifies the path option. • The **protect** keyword and *preference-number argument identify the path option with protection.* • The **dynamic** keyword indicates that the path option is dynamically calculated. (The router figures out the best path.) • The **explicit** keyword indicates that the path option is specified. The IP addresses are specified for the path. • The **name** *path-name*keyword argument combination identifies the name of the explicit path option. • The *path-number* argument identifies the number of the explicit path option. • The **verbatim** keyword bypasses the topology database verification. **Note** You can use the **verbatim** keyword only with the explicit path option. • The **attributes** *string* keyword argument combination names an attribute list to specify path options for the LSP. • The **bandwidth** keyword specifies the LSP bandwidth.

	Command or Action	Purpose
		• The **sub-pool** keyword indicates a subpool path option. • The *kbps* argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295. • The **lockdown** keyword disables reoptimization of the LSP.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying Verbatim LSPs for MPLS TE Tunnels

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls traffic-eng tunnels** *tunnel-interface* [**brief**] **Example:** `Router# show mpls traffic-eng tunnels tunnel1`	Displays information about tunnels including those configured with an explicit path option using verbatim.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Configuration Examples for MPLS Traffic Engineering Verbatim Path Support

Configuring MPLS Traffic Engineering Verbatim Path Support Example

The following example shows a tunnel that has been configured with an explicit path option using verbatim:

```
interface tunnel 1
 ip unnumbered loopback 1
 tunnel destination 10.10.100.100
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng path-option 1 explicit name path1 verbatim
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Interface commands	*Cisco IOS Interface and Hardware Component Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this release.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering Verbatim Path Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 168: Feature Information for MPLS Traffic Engineering Verbatim Path Support

Feature Name	Releases	Feature Information
MPLS Traffic Engineering: Verbatim Path Support	Cisco IOS XE Release 2.3	The MPLS Traffic Engineering Verbatim Path Support feature allows network nodes to support Resource Reservation Protocol (RSVP) extensions without supporting Interior Gateway Protocol (IGP) extensions for traffic engineering (TE), thereby bypassing the topology database verification process. The following commands were introduced or modified: **show mpls traffic-eng tunnels**, **tunnel mpls traffic-eng path option**.

Glossary

Fast Reroute --Procedures that enable temporary routing around a failed link or node while a new label-switched path (LSP) is being established at the head end.

headend --The router that originates and maintains a given label-switched path (LSP) . This is the first router in the LSP's path.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information protocol (RIP).

LSP --label-switched path. A configured connection between two routers, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

LSR --label switching router. A device that forwards Multiprotocol Label Switching (MPLS) packets based on the value of a fixed-length label encapsulated in each packet.

merge point --The backup tunnel's tail.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

PLR --point of local repair. The head-end of the backup tunnel.

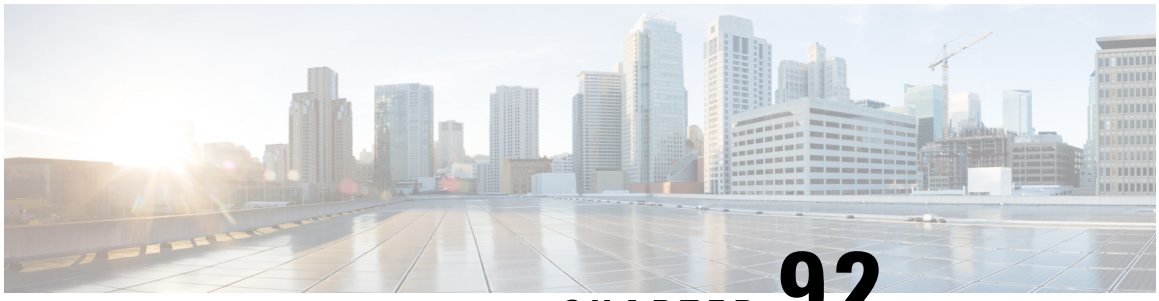
RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

SPF --shortest path first. Routing algorithm that iterates on length of path to determine a shortest-path spanning tree. Commonly used in link-state routing algorithms. Sometimes called Dijkstra's algorithm.

tailend --The router upon which an label-switched path (LSP) is terminated. This is the last router in the LSP's path.

traffic engineering --The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel --A secure communications path between two peers, such as routers.



CHAPTER 92

MPLS Traffic Engineering--RSVP Hello State Timer

The MPLS Traffic Engineering--RSVP Hello State Timer feature detects when a neighbor is down and quickly triggers a state timeout, which frees resources such as bandwidth that can be reused by other label switched paths (LSPs).

Resource Reservation Protocol (RSVP) hellos can be used to detect when a neighboring node is down. The hello state timer then triggers a state timeout. As a result, network convergence time is reduced, and nodes can forward traffic on alternate paths or assist in stateful switchover (SSO) operation.

Prerequisites for MPLS Traffic Engineering--RSVP Hello State Timer

Perform the following tasks on routers before configuring the MPLS Traffic Engineering--RSVP Hello State Timer feature:

- Configure Resource Reservation Protocol (RSVP).
- Enable Multiprotocol Label Switching (MPLS).
- Configure traffic engineering (TE).
- Enable hellos for state timeout.

Restrictions for MPLS Traffic Engineering--RSVP Hello State Timer

- Hellos for state timeout are dependent on graceful restart, if it is configured; however, graceful restart is independent of hellos for state timeout.
- Unnumbered interfaces are not supported.
- Hellos for state timeout are configured on a per-interface basis.

Information About MPLS Traffic Engineering - Configurable Path Calculation Metric for Tunnels

Hellos for State Timeout

When RSVP signals a TE LSP and there is a failure somewhere along the path, the failure can remain undetected for as long as two minutes. During this time, bandwidth is held by the nonfunctioning LSP on the nodes downstream from the point of failure along the path with the state intact. If this bandwidth is needed by headend tunnels to signal or resignal LSPs, tunnels may fail to come up for several minutes thereby negatively affecting convergence time.

Hellos enable RSVP nodes to detect when a neighboring node is not reachable. After a certain number of intervals, hellos notice that a neighbor is not responding and delete its state. This action frees the node's resources to be reused by other LSPs.

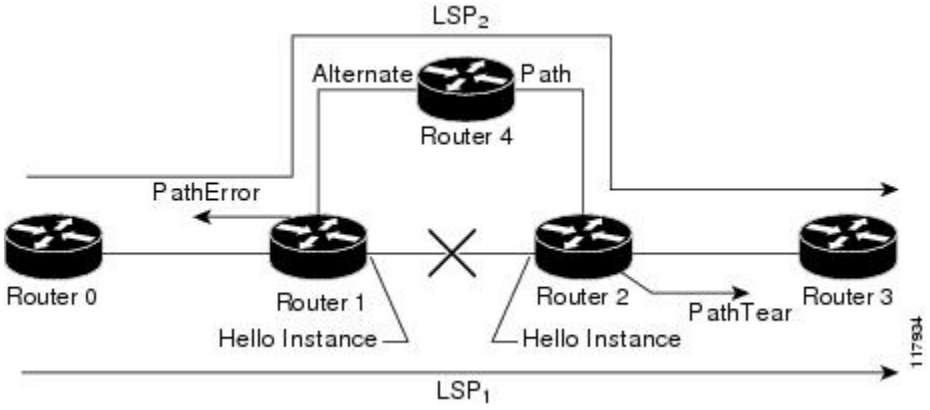
Hellos must be configured both globally on the router and on the specific interface to be operational.

Hello Instance

A hello instance implements RSVP hellos for a given router interface address and a remote IP address. A hello instance is expensive because of the large number of hello requests that are sent and the strains they put on the router resources. Therefore, you should create a hello instance only when it is needed to time out state and delete the hello instance when it is no longer necessary.

Hellos for Nonfast-Reroutable TE LSP

The figure below shows a nonfast-reroutable TE LSP from Router 1 to Router 3 via Router 2.



Assume that the link between Router 1 and Router 2 fails. This type of problem can be detected by various means including interface failure, Interior Gateway Protocol (IGP) (Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS)), and RSVP hellos. However, sometimes interface failure cannot be detected; for example, when Router 1 and Router 2 are interconnected through a Layer 2 switch. The IGP may be slow detecting the failure. Or there may be no IGP running between Router 1 and Router 2; for example, between two Autonomous System Boundary Routers (ASBRs) interconnecting two autonomous systems.

If hellos were running between Router 1 and Router 2, each router would notice that communication was lost and time out the state immediately.

Router 2 sends a delayed PathTear message to Router 3 so that the state can be deleted on all nodes thereby speeding up the convergence time.

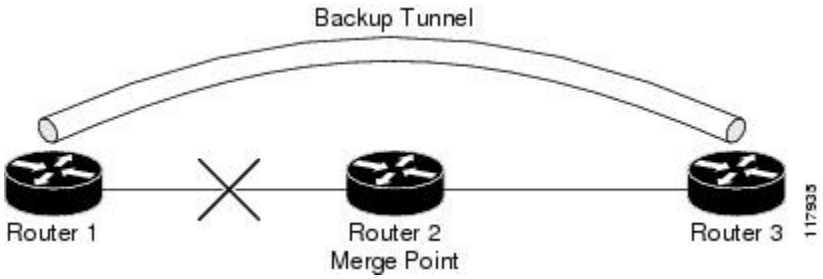
Note The PathTear message is delayed one second because on some platforms data is being forwarded even after the control plane is down.

Router 1 sends a destructive PathError message upstream to Router 0 with error code ROUTING_PROBLEM and error value NO_ROUTE.

LSP1 goes from Router 0 to Router 1 to Router 2 to Router 3; LSP 2 goes from Router 0 to Router 1 to Router 4 to Router 2 to Router 3.

Hellos for Fast-Reroutable TE LSP with Backup Tunnel

The figure below shows a fast reroutable TE LSP with a backup tunnel from Router1 to Router 2 to Router 3.



This TE LSP has a backup tunnel from Router 1 to Router 3 protecting the fast reroutable TE LSP against a failure in the Router 1 to Router 2 link and node Router 2. However, assume that a failure occurs in the link connecting Router 1 to Router 2. If hellos were running between Router 1 and Router 2, the routers would notice that the link is down, but would not time out the state. Router 2 notices the failure, but cannot time out the TE LSP because Router 2 may be a merge point, or another downstream node may be a merge point. Router 1 notices the failure and switches to the backup LSP; however, Router 1 cannot time out the state either.

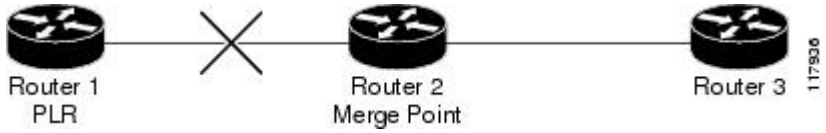
Note A hello instance is not created in the preceding scenario because the neighbor is down and the hello instance cannot take action.

Hellos for Fast-Reroutable TE LSP Without Backup Tunnel

On a fast-reroutable TE LSP with no backup tunnel, a hello instance can be created with the neighbor downstream (next hop (NHOP)). On a nonfast-reroutable TE LSP, a hello instance can be created with the neighbor downstream (NHOP) and the neighbor upstream (previous hop (PHOP)). This is in addition to the existing hellos for Fast Reroute.

Note If both Fast Reroute and hellos for state timeout hello instances are needed on the same link, only one hello instance is created. It will have the Fast Reroute configuration including interval, missed refreshes, and differentiated services code point (DSCP). When a neighbor is down, Fast Reroute and the hello state timer take action.

The figure below shows a fast-reroutable TE LSP. without a backup tunnel, from Router 1 (the point of local repair (PLR)), to Router 2 to Router 3.



Assume that a failure occurs in the link connecting Router 1 to Router 3. Router 1 can time out the state for the TE LSP because Router 1 knows there is no backup tunnel. However, Router 2 cannot time out the state because Router 2 does not know whether a backup tunnel exists. Also, Router 2 may be a merge point, and therefore cannot time out the state.

Note A hello instance is not created in the preceding scenario because the neighbor is down and the hello instance cannot take action.

How to Configure MPLS Traffic Engineering--RSVP Hello State Timer

Note The following tasks also enable Fast Reroute; however, this section focuses on the RSVP hello state timer.

Enabling the Hello State Timer Globally

Perform this task to enable the RSVP hello state timer globally to reduce network convergence, allow nodes to forward traffic on alternate paths, or assist in stateful switchover (SSO) operation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello** **Example:** `Router(config)# ip rsvp signalling hello`	Enables hellos for state timeout globally on a router.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Enabling the Hello State Timer on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*]
4. **ip rsvp signalling hello**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*] **Example:** `Router(config)# interface FastEthernet 0/0/0`	Enters interface configuration mode. • The *type slot subslot* / *port*[**.** *subinterface-number*] arguments identify the interface to be configured.
Step 4	**ip rsvp signalling hello** **Example:** `Router(config-if)# ip rsvp signalling hello`	Enables hellos for state timeout on an interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Setting a DSCP Value on an Interface

Perform this task to set a differentiated services code point DSCP value for hello messages on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*]
4. **ip rsvp signalling hello reroute dscp** *num*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*] **Example:** `Router(config)# interface FastEthernet 0/0/0`	Enters interface configuration mode. • The *type slot* / *subslot* / *port*[**.** *subinterface-number*] arguments identify the interface to be configured.
Step 4	**ip rsvp signalling hello reroute dscp** *num* **Example:** `Router(config-if)# ip rsvp signalling hello reroute dscp 30`	Sets a DSCP value for RSVP hello messages on an interface of a router from 0 to 63 with hellos for state timeout enabled.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Setting a Hello Request Interval on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*]
4. **ip rsvp signalling hello reroute refresh interval** *interval-value*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*] **Example:** `Router(config)# interface FastEthernet 0/0/0`	Enters interface configuration mode. • The *type slot subslot* / *port*[**.** *subinterface-number*] argument identifies the interface to be configured.
Step 4	**ip rsvp signalling hello reroute refresh interval** *interval-value* **Example:** `Router(config-if)# ip rsvp signalling hello reroute refresh interval 5000`	Sets a hello request interval on an interface of a router with hellos for state timer enabled.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Setting the Number of Hello Messages that can be Missed on an Interface

Perform this task to set the number of consecutive hello messages that are lost (missed) before hello declares the neighbor down.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*]
4. **ip rsvp signalling hello reroute refresh misses** *msg-count*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* [**.** *subinterface-number*] **Example:** `Router(config)# interface FastEthernet 0/0/0`	Enters interface configuration mode. • The *type slot subslot* / *port*[**.** *subinterface-number*] arguments identify the interface to be configured.
Step 4	**ip rsvp signalling hello reroute refresh misses** *msg-count* **Example:** `Router(config-if)# ip rsvp signalling hello reroute refresh misses 5`	Configures the number of consecutive hello messages that are lost before hello declares the neighbor down.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Verifying Hello for State Timer Configuration

SUMMARY STEPS

1. **enable**
2. **show ip rsvp hello**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp hello** **Example:** `Router# show ip rsvp hello`	Displays the status of RSVP TE hellos and statistics including hello state timer (reroute).

Configuration Examples for MPLS Traffic Engineering--RSVP Hello State Timer

Example

In the following example, the hello state timer is enabled globally and on an interface. Related parameters, including a DSCP value, a refresh interval, and a missed refresh limit, are set on an interface.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp signalling hello
Router(config)# interface FastEthernet 0/0/0
Router(config-if)# ip rsvp signalling hello
Router(config-if)# ip rsvp signalling hello reroute dscp 30
Router(config-if)# ip rsvp signalling hello reroute refresh interval 5000
Router(config-if)# ip rsvp signalling hello reroute refresh misses 5
Router(config-if)# end
```

The following example verifies the status of the hello state timer (reroute):

```
Router# show ip rsvp hello
Hello:
  Fast-Reroute/Reroute:Enabled
    Statistics:Enabled
  Graceful Restart:Enabled (help-neighbor only)
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	• *Cisco IOS Quality of Service Solutions Command Reference* • *Cisco IOS Multiprotocol Label Switching Command Reference*
Stateful Switchover	Stateful Switchover
MPLS Label Distribution Protocol	MPLS Label Distribution Protocol (LDP) Overview
Cisco nonstop forwarding	Cisco Nonstop Forwarding
Information on backup tunnels, link and node failures, RSVP hellos	MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection)

Related Topic	Document Title
Graceful restart	NSF/SSO - MPLS TE and RSVP Graceful Restart

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBS are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3209	RSVP-TE: Extensions to RSVP for LSP Tunnels

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering--RSVP Hello State Timer

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 169: Feature Information for MPLS Traffic Engineering--RSVP Hello State Timer

Feature Name	Releases	Feature Information
MPLS Traffic Engineering--RSVP Hello State Timer	Cisco IOS XE Release 2.3	The MPLS Traffic Engineering--RSVP Hello State Timer feature detects when a neighbor is down and quickly triggers a state timeout, which frees resources such as bandwidth that can be reused by other label switched paths (LSPs). This feature was integrated into Cisco IOS XE Release 2.3. The following commands were introduced or modified: **ip rsvp signalling hello dscp**, **ip rsvp signalling hello refresh interval**, **ip rsvp signalling hello refresh misses**, **ip rsvp signalling hello reroute dscp**, **ip rsvp signalling hello reroute refresh interval**, **ip rsvp signalling hello reroute refresh misses**, **show ip rsvp hello**.

Glossary

autonomous system --A collection of networks that share the same routing protocol and that are under the same system administration.

ASBR --autonomous system boundary router. A router that connects and exchanges information between two or more autonomous systems.

backup tunnel --A Multiprotocol Label Switching (MPLS) traffic engineering tunnel used to protect other (primary) tunnel traffic when a link or node failure occurs.

DSCP --differentiated services code point. Six bits in the IP header, as defined by the Internet Engineering Task Force (IETF). These bits determine the class of service provided to the IP packet.

FRR --Fast Reroute. A mechanism for protecting Multiprotocol Label Switching (MPLS) traffic engineering (TE) label switched paths (LSPs) from link and node failure by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

graceful restart --A process for helping a neighboring Route Processor (RP) restart after a node failure has occurred.

headend --The router that originates and maintains a given label switched paths (LSP). This is the first router in the LSP's path.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Internal Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

IS-IS --Intermediate System-to-Intermediate System. Open systems Interconnection (OSI) link-state hierarchical routing protocol whereby Intermediate System (IS) routers exchange routing information based on a single metric to determine network topology.

instance --A mechanism that implements the RSVP hello extensions for a given router interface address and remote IP address. Active hello instances periodically send Hello Request messages, expecting Hello ACK messages in response. If the expected ACK message is not received, the active hello instance declares that

the neighbor (remote IP address) is unreachable (that is, it is lost). This can cause LSPs crossing this neighbor to be fast rerouted.

label --A short, fixed-length data identifier that tells switching nodes how to forward data (packets or cells).

LDP --Label Distribution Protocol. The protocol that supports Multiprotocol Label Switching (MPLS) hop-by-hop forwarding by distributing bindings between labels and network prefixes. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LSP --label switched path is a configured connection between two routers, in which Multiprotocol Label Switching (MPLS) is used to carry packets. The LSP is created by the concatenation of one or more label-switched hops, allowing a packet to be forwarded by swapping labels from one MPLS node to another MPLS node.

merge point --The backup tunnel's tail.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. MPLS enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels.

OSPF --Open Shortest Path First. A link-state routing protocol used for routing.

PLR --point of local repair. The headend of the backup tunnel.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

state --Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

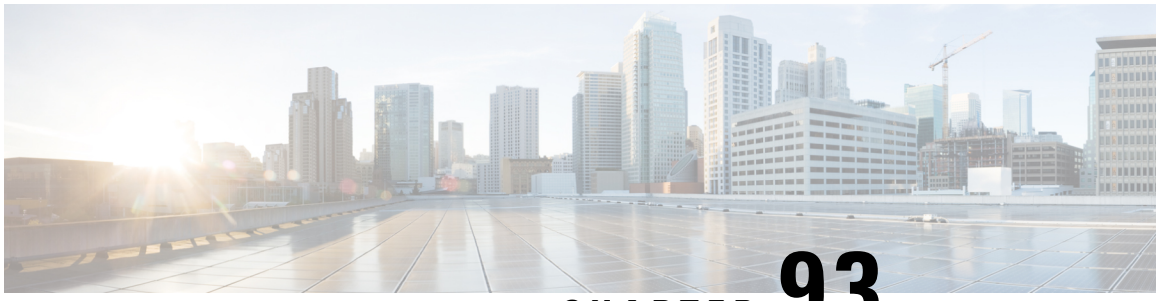
tailend --The router upon which an LSP is terminated. This is the last router in the LSP's path.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

topology --The physical arrangement of network nodes and media within an enterprise networking structure.

tunnel --Secure communications path between two peers, such as two routers.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.



CHAPTER 93

MPLS Traffic Engineering Forwarding Adjacency

The MPLS Traffic Engineering Forwarding Adjacency feature allows a network administrator to handle a traffic engineering (TE) label switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network based on the Shortest Path First (SPF) algorithm.

Both Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) are supported.

Prerequisites for MPLS Traffic Engineering Forwarding Adjacency

Your network must support the following Cisco IOS XE features:

- Multiprotocol Label Switching (MPLS)
- IP Cisco Express Forwarding
- IS-IS

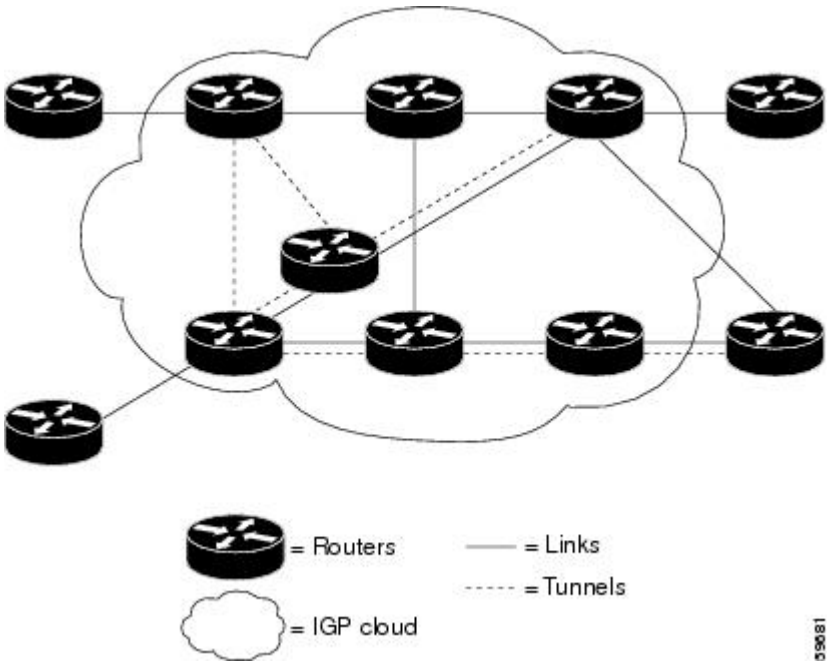
Restrictions for MPLS Traffic Engineering Forwarding Adjacency

- Using the MPLS Traffic Engineering Forwarding Adjacency feature increases the size of the IGP database by advertising a TE tunnel as a link.
- When the MPLS Traffic Engineering Forwarding Adjacency feature is enabled on a TE tunnel, the link is advertised in the IGP network as a type, length, value (TLV) 22 object without any TE sub-TLV.
- You must configure MPLS TE forwarding adjacency tunnels bidirectionally.

Information About MPLS Traffic Engineering Forwarding Adjacency

MPLS Traffic Engineering Forwarding Adjacency Functionality

The MPLS Traffic Engineering Forwarding Adjacency feature allows a network administrator to handle a TE LSP tunnel as a link in an IGP network based on the SPF algorithm. A forwarding adjacency can be created between routers regardless of their location in the network. The routers can be located multiple hops from each other, as shown in the figure below.



As a result, a TE tunnel is advertised as a link in an IGP network with the link's cost associated with it.

Routers outside of the TE domain see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network.

MPLS Traffic Engineering Forwarding Adjacency Benefits

TE tunnel interfaces advertised for SPF--TE tunnel interfaces are advertised in the IGP network just like any other links. Routers can then use these advertisements in their IGPs to compute the SPF even if they are not the headend of any TE tunnels.

How to Configure MPLS Traffic Engineering Forwarding Adjacency

Configuring a Tunnel Interface for MPLS TE Forwarding Adjacency

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **exit**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 0`	Designates a tunnel interface for the forwarding adjacency, and enters interface configuration mode.
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring MPLS TE Forwarding Adjacency on Tunnels

Note You must configure a forwarding adjacency on two LSP tunnels bidirectionally, from A to B and B to A. Otherwise, the forwarding adjacency is advertised, but not used in the IGP network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng forwarding-adjacency** [**holdtime** *value*]
5. **isis metric** {*metric-value*| **maximum**} {**level-1**| **level-2**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 0`	Designates a tunnel interface for the forwarding adjacency, and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng forwarding-adjacency** [**holdtime** *value*] **Example:** `Router(config-if)# tunnel mpls traffic-eng forwarding-adjacency`	Advertises a TE tunnel as a link in an IGP network.
Step 5	**isis metric** {*metric-value*| **maximum**} {**level-1**| **level-2**} **Example:** `Router(config-if)# isis metric 2 level-1`	Configures the IS-IS metric for a tunnel interface to be used as a forwarding adjacency. • You should specify the **isis metric** command with level-1 or level-2 to be consistent with the IGP level at which you are performing traffic engineering. Otherwise, the metric has the default value of 10.

Verifying MPLS TE Forwarding Adjacency

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng forwarding-adjacency** [*ip-address*]
3. **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
4. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show mpls traffic-eng forwarding-adjacency** [*ip-address*]

Use this command to see the current tunnels. For example:

Example:

```
Router# show mpls traffic-eng forwarding-adjacency

 destination 0168.0001.0007.00 has 1 tunnels
  Tunnel7     (traffic share 100000, nexthop 192.168.1.7)
                (flags:Announce Forward-Adjacency, holdtime 0)
Router# show mpls traffic-eng forwarding-adjacency 192.168.1.7
 destination 0168.0001.0007.00 has 1 tunnels
  Tunnel7     (traffic share 100000, nexthop 192.168.1.7)
                (flags:Announce Forward-Adjacency, holdtime 0)
```

Step 3 **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]

Use this command to display information about the IS-IS link-state database. For example:

Example:

```
Router# show isis database
IS-IS Level-1 Link State Database

LSPID                 LSP Seq Num    LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C00.0C35.00-00  0x0000000C     0x5696        792           0/0/0
0000.0C00.40AF.00-00  0x00000009     0x8452        1077          1/0/0
0000.0C00.62E6.00-00  0x0000000A     0x38E7        383           0/0/0
0000.0C00.62E6.03-00  0x00000006     0x82BC        384           0/0/0
0800.2B16.24EA.00-00  0x00001D9F     0x8864        1188          1/0/0
0800.2B16.24EA.01-00  0x00001E36     0x0935        1198          1/0/0

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num    LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C00.0C35.03-00  0x00000005     0x04C8        792           0/0/0
0000.0C00.3E51.00-00  0x00000007     0xAF96        758           0/0/0
```

```
0000.0C00.40AF.00-00  0x0000000A     0x3AA9        1077          0/0/0
```

Step 4 **exit**

Use this command to exit to user EXEC. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS Traffic Engineering Forwarding Adjacency

This section provides a configuration example for the MPLS Traffic Engineering Forwarding Adjacency feature using an IS-IS metric.

Example MPLS TE Forwarding Adjacency

The following output shows the configuration of a tunnel interface, a forwarding adjacency, and an IS-IS metric:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tunnel 7
Router(config-if)# tunnel mpls traffic-eng forwarding-adjacency
Router(config-if)# isis metric 2 level-1
```

Following is sample command output when a forwarding adjacency has been configured:

```
Router# show running-config
Building configuration...
Current configuration :364 bytes
!
interface Tunnel7
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 192.168.1.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng forwarding-adjacency
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng path-option 10 explicit name short
 isis metric 2 level 1
```

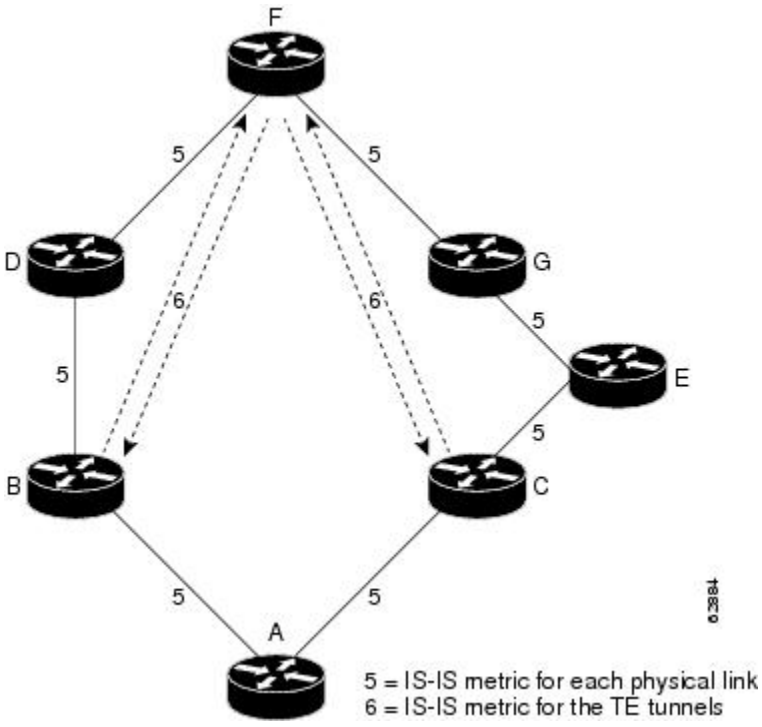
Note Do not specify the **tunnel mpls traffic-eng autoroute announce** command in your configuration when you are using forwarding adjacency.

Following is an example where forwarding adjacency is configured with OFPF:

```
Router# configure terminal

Router# show running-config

Building configuration...
Current configuration : 310 bytes
interface tunnel 1
!
interface Tunnel1
 ip unnumbered Loopback0
 ip ospf cost 6
 tunnel destination 172.16.255.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng forwarding-adjacency  tunnel mpls
 traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 10 dynamic
 end
Router# show mpls traffic-eng forwarding-adjacency

 destination 172.16.255.5, area ospf 172  area 0, has 1 tunnels
  Tunnel1      (load balancing metric 2000000, nexthop 172.16.255.5)
               (flags:  Forward-Adjacency, holdtime 0)
Router#
```

Usage Tips

In the figure below, if you have no forwarding adjacencies configured for the TE tunnels between Band F and C and F, all the traffic that A must forward to F goes through B because B is the shortest path from A to F. (The cost from A to F is 15 through B and 20 through C.)



If you have forwarding adjacencies configured on the TE tunnels between B and F and C and F and also on the TE tunnels between F and B and F and C, then when A computes the SPF algorithm, A sees two equal cost paths of 11 to F. As a result, traffic across the A-B and A-C links is shared.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS traffic engineering commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
IP switching commands	*Cisco IOS IP Switching Command Reference*
IS-IS TLVs	Intermediate System-to-Intermediate System (IS-IS) TLVs (white paper)

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Glossary

Cisco Express Forwarding --A scalable, distributed, Layer 3 switching solution designed to meet the future performance requirements of the Internet and enterprise networks.

forwarding adjacency --A traffic engineering link (or LSP) into an IS-IS/OSPF network.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

IS-IS --Intermediate System-to-Intermediate System. Open System Interconnection (OSI) link-state hierarchical routing protocol whereby Intermediate System (IS) routers exchange routing information based on a single metric to determine network topology.

label switched path (LSP) --A sequence of hops (R0...R*n*) in which a packet travels from R0 to R*n* through label switching mechanisms. A switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

label switched path (LSP) tunnel --A configured connection between two routers, using label switching to carry the packets.

MPLS-- Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

OSPF --Open Shortest Path First. A link-state, hierarchical IGP routing algorithm proposed as a successor to RIP in the Internet community. OSPF features include least-cost routing, multipath routing, and load balancing. OSPF was derived from an early version of the IS-IS protocol. *See also* IS-IS.

SPF --Shortest Path First. A routing algorithm used as the basis for OSPF operations. When an SPF router is powered up, it initializes its routing-protocol data structures and then waits for indications from lower-layer protocols that its interfaces are functional.

TLV --type, length, value. A block of information embedded in Cisco Discovery Protocol advertisements.

traffic engineering --The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been applied.

traffic engineering tunnel --A label switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing would cause the tunnel to take.

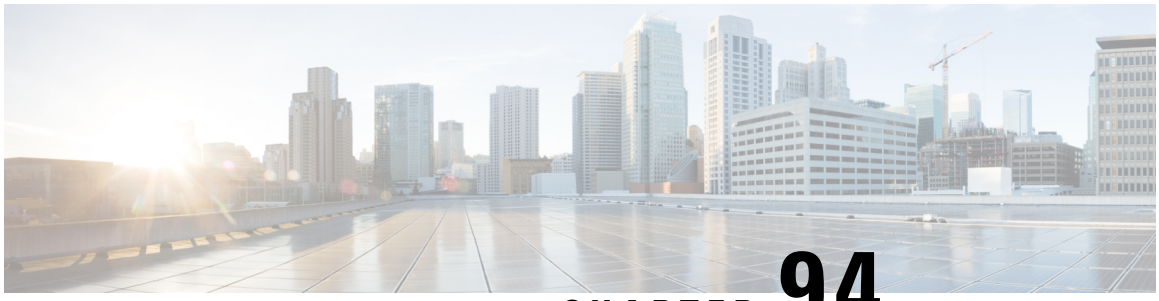
Feature Information for MPLS Traffic Engineering Forwarding Adjacency

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 170: Feature Information for MPLS Traffic Engineering Forwarding Adjacency

Feature Name	Releases	Feature Information
MPLS Traffic Engineering Forwarding Adjacency	12.0(15)S 12.0(16)ST 12.2(18)S 12.2(18)SXD 12.2(27)SBC 12.2(28)SB 12.4(20)T Cisco IOS XE Release 2.3	The MPLS Traffic Engineering Forwarding Adjacency feature allows a network administrator to handle a TE LSP tunnel as a link in an IGP network based on the SPF algorithm. In 12.0(15)S, this feature was introduced. In 12.0(16)ST, this feature was integrated. In 12.2(18)S, this feature was integrated. In 12.2(18)SXD, this feature was integrated. In 12.2(27)SBC, this feature was integrated. In 12.2(28)SB, this feature was integrated. In 12.4(20)T, this feature was integrated. In Cisco IOS XE Release 2.3, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified: **debug mpls traffic-eng forwarding-adjacency**, **show mpls traffic-eng forwarding-adjacency**, and **tunnel mpls traffic-eng forwarding-adjacency**.



CHAPTER 94

MPLS Traffic Engineering Class-based Tunnel Selection

The MPLS Traffic Engineering (TE): Class-based Tunnel Selection feature enables you to dynamically route and forward traffic with different class of service (CoS) values onto different TE tunnels between the same tunnel headend and the same tailend. The TE tunnels can be regular TE or DiffServ-aware TE (DS-TE) tunnels.

The set of TE (or DS-TE) tunnels from the same headend to the same tailend that you configure to carry different CoS values is referred to as a "tunnel bundle." After configuration, Class-Based Tunnel Selection (CBTS) dynamically routes and forwards each packet into the tunnel that:

- Is configured to carry the CoS of the packet
- Has the right headend for the destination of the packet

Because CBTS offers dynamic routing over DS-TE tunnels and requires minimum configuration, it greatly eases deployment of DS-TE in large-scale networks.

CBTS can distribute all CoS values on eight different tunnels.

CBTS also allows the TE tunnels of a tunnel bundle to exit headend routers through different interfaces.

Prerequisites for MPLS Traffic Engineering Class-based Tunnel Selection

- Multiprotocol Label Switching (MPLS) must be enabled on all tunnel interfaces.
- Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled in global configuration mode.

Restrictions for MPLS Traffic Engineering Class-based Tunnel Selection

- For a given destination, all CoS values are carried in tunnels terminating at the same tailend. Either all CoS values are carried in tunnels or no values are carried in tunnels. In other words, for a given destination, you cannot map some CoS values in a DS-TE tunnel and other CoS values in a Shortest Path First (SPF) Label Distribution Protocol (LDP) or SPF IP path.
- CBTS does not allow load-balancing of a given experimental (EXP) value in multiple tunnels. If two or more tunnels are configured to carry a given EXP value, CBTS picks one of those tunnels to carry this EXP value.
- The operation of CBTS is not supported MPLS TE Automesh or label-controlled (LC)-ATM.
- For Any Transport over MPLS (AToM), the operation of CBTS is supported only with Ethernet over MPLS (EoMPLS).
- With Cisco IOS XE Release 3.6S and later releases, you must configure a primary tunnel to make CBTS work. For configuration information, see the “Configuring a Primary Tunnel” section.

Information About MPLS Traffic Engineering Class-based Tunnel Selection

Incoming Traffic Supported by MPLS TE Class-based Tunnel Selection

The CBTS feature supports the following kinds of incoming packets:

- At a provider edge (PE) device—Unlabeled packets that enter a Virtual Private Network (VPN) routing and forwarding (VRF) instance interface
- At a provider core (P) device—Unlabeled and MPLS-labeled packets that enter a non-VRF interface
- At a PE device in a Carrier Supporting Carrier (CSC) or interautonomous system (Inter-AS)—MPLS-labeled packets that enter a VRF interface

CoS Attributes for MPLS TE Class-based Tunnel Selection

CBTS supports tunnel selection based on the value of the EXP field that the headend device imposes on the packet. Before imposing this value, the device considers the input modular quality of service (QoS) command-line interface (CLI) (MQC). If the input MQC modifies the EXP field value, CBTS uses the modified value for its tunnel selection.

Packets may enter the headend from multiple incoming interfaces. These interfaces can come from different customers that have different DiffServ policies. In such cases, service providers generally use input MQC to apply their own DiffServ policies and mark imposed EXP values accordingly. Thus, CBTS can operate consistently for all customers by considering the EXP values marked by the service provider.

Note If the output MQC modifies the EXP field, CBTS ignores the change in the EXP value.

CBTS allows up to eight different tunnels on which it can distribute all classes of service.

Routing Protocols and MPLS TE Class-based Tunnel Selection

CBTS routes and forwards packets to MPLS TE tunnels for specified destinations through use of the following routing protocols:

- Intermediate System-to-Intermediate System (IS-IS) with Autoroute configured
- Open Shortest Path First (OSPF) with Autoroute configured
- Static routing
- Border Gateway Protocol (BGP) with recursion configured on the BGP next hop with packets forwarded on the tunnel through the use of IS-IS, OSPF, or static routing

Tunnel Selection with MPLS TE Class-based Tunnel Selection

This section contains the following topics related to tunnel selection:

EXP Mapping Configuration

With CBTS, you can configure each tunnel with any of the following:

- The same EXP information configured as it was before the CBTS feature was introduced, that is, with no EXP-related information
- One or more EXP values for the tunnel to carry
- A property that allows the carrying of all EXP values not currently allocated to any up-tunnel (default)
- One or more EXP values for the tunnel to carry, and the default property that allows the carrying of all EXP values not currently allocated to any up-tunnel

The default property (the carrying of all EXP values not currently allocated to any up-tunnel) effectively provides a way for the operator to avoid explicitly listing all possible EXP values. Even more important, the default property allows the operator to indicate tunnel preferences onto which to “bump” certain EXP values, should the tunnel carrying those EXP values go down. (See the **tunnel mpls traffic-eng exp** command for the command syntax.)

The configuration of each tunnel is independent of the configuration of any other tunnel. CBTS does not attempt to perform any consistency check for EXP configuration.

This feature allows configurations where:

- Not all EXP values are explicitly allocated to tunnels.
- Multiple tunnels have the default property.
- Some tunnels have EXP values configured and others do not have any values configured.

- A given EXP value is configured on multiple tunnels.

Tunnel Selection for EXP Values

Tunnel selection with this feature is a two-step process:

1. For a given prefix, routing (autoroute, static routes) occurs exactly as it did without the CBTS feature. The device selects the set of operating tunnels that have the best metrics, regardless of the EXP-related information configured on the tunnel.
2. CBTS maps all of the EXP values to the selected set of tunnels.
3. If a given EXP value is configured:
   - On only one of the tunnels in the selected set, CBTS maps the EXP value onto that tunnel.
   - On two or more of the tunnels in the selected set, CBTS arbitrarily maps the EXP value onto one of these tunnels.
4. If a given EXP value is not configured on any of the tunnels in the selected set:
   - And only one of the tunnels in the selected set is configured as a default, CBTS maps the EXP value onto that tunnel.
   - And two or more of the tunnels in the selected set are configured as defaults, CBTS arbitrarily maps the EXP value onto one of these tunnels.
   - And no tunnel in the selected set of tunnels is configured as a default, CBTS arbitrarily maps the EXP value onto one of these tunnels.

CBTS relies on autoroute to select the tunnel bundle. Autoroute selects only tunnels that are on the SPF to the destination. Therefore, similar to Autoroute, CBTS does not introduce any risk of routing loops.

Tunnel Selection Examples

The following examples show various tunnel configurations that are set up by an operator and indicate how CBTS maps packets carrying EXP values onto these tunnels. Each example describes a different configuration: a default tunnel configured, more than one tunnel configured with the same EXP value, and so on.

Example 1—Default Tunnel Configured

An operator configures the following parameters on tunnels T1 and T2:

- T1: exp = 5
- T2: exp = default

If T1 and T2 are next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1
- Packets with <Dest = P, exp = anything-other-than-5> onto T2

Example 2—EXP Values Configured on Two Tunnels; One Default Tunnel

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5

- T2: exp = 3 and 4
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1
- Packets with <Dest = P, exp = 3 or 4> onto T2
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3

Example 3—More than One Tunnel with the Same EXP

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 5
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (arbitrary selection)
- Packets with <Dest = P, exp = anything-other-than-5> onto T3
- No packets onto T2

Example 4—Static Route Configured

An operator configures the following parameters on tunnels T1 and T2:

- T1: exp = 5
- T2: exp = 3
- Static route to P on T2

If prefix P is behind the T1 and T2 tailend device, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = anything> onto T2
- No packets onto T1

Static routes are preferred over dynamic routes; therefore, the device chooses only T2 as the "selected set" of tunnels.

Example 5—No Default or Metric Configuration

An operator configures the following parameters on tunnels T1 and T2:

- T1: exp = 5
- T2: exp = 3

If T1 and T2 are the next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1
- Packets with <Dest = P, exp = 3> onto T2
- Packets with <Dest = P, exp = anything-other-than-3-or-5> onto T2

If a packet arrives with an EXP value that is different from any value configured for a tunnel, the packet goes in to the default tunnel. If no default tunnel is configured, the packet goes in to the tunnel that is arbitrarily selected by CBTS.

Multipath with Non-TE Paths and MPLS TE Class-Based Tunnel Selection

For a given prefix in the routing process, the device might select a set of paths that includes both TE tunnels and non-TE-tunnel paths (SPF paths). For example, internal Border Gateway Protocol (iBGP) Multipath might be activated and result in multiple BGP next hops for that prefix, where one BGP next hop is reachable through TE tunnels and other BGP next hops are reachable through non-TE-tunnel paths.

An equal cost IGP path might also exist over TE tunnels and over a non-TE tunnel path. For example, a TE tunnel metric might be modified to be equal to the SPF path.

In these situations, CBTS maps traffic in the following manner:

- If a given EXP value is configured on one or more of the tunnels in the selected set, CBTS maps the EXP value onto that tunnel or one of those tunnels.
- If a given EXP value is not configured on any of the tunnels in the selected set but one or more of the tunnels is configured as a default in the selected set, then CBTS maps the EXP value onto that tunnel or one of those tunnels.
- If a given EXP value is not configured on any of the tunnels from the selected set and no tunnel in the selected set is configured as a default, CBTS arbitrarily maps the EXP value onto one of the tunnels in the selected set, and performs CoS-unaware load-balancing with other non-TE paths.
- If the routing process allocates all EXP values to tunnels or if a default is used, then routing does not use the non-TE paths unless all TE tunnels are down.

MPLS TE Class-Based Tunnel Selection and Policy-Based Routing

If you configure both policy-based routing (PBR) over TE tunnels (in non-VRF environments) and CBTS, the PBR decision overrides the CBTS decision. PBR is an input process that the device performs ahead of regular forwarding.

Tunnel Failure Handling

For CBTS operation, the important question is whether the tunnel interface is up or down, not whether the current TE label switched path (LSP) is up or down. For example, a TE LSP might go down but is reestablished by the headend because another path option exists. The tunnel interface does not go down during the transient period while the TE LSP is reestablished. Because the tunnel interface does not go down, the corresponding EXP does not get rerouted onto another tunnel during the transient period.

When a tunnel used by CBTS for forwarding goes down, the feature adjusts its tunnel selection for the affected EXP values. It reapplies the tunnel selection algorithm to define the behavior of packets for all EXP values, as shown in the examples that follow.

Example 1—Tunnel Other than the Default Tunnel Goes Down

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 3 and 4
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P and Tunnel T1 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 3, 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3 (as before)
- Packets with <Dest = P, exp = 5> onto T3

Example 2—Default Tunnel Goes Down

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 3 and 4
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P and Tunnel T3 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (as before)
- Packets with <Dest = P, exp = 3, 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T1

Example 3—Two Default Tunnels Are Configured

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 3, 4, and default
- T3: exp = 0, 1, 2, 6, 7, and default

If T1, T2, and T3 are next-hop interfaces for prefix P and Tunnel T3 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (as before)
- Packets with <Dest = P, exp = 3, 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T2

If tunnel T2 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3 (as before)
- Packets with <Dest = P, exp = 3, or 4> onto T3

If tunnel T1 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 3, or 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3 (as before)
- Packets with <Dest = P, exp = 5> onto either T2 or T3, but not both

In Example 3, the operator configures the EXP default option on two tunnels to ensure that nonvoice traffic is never redirected onto the voice tunnel (T1).

Misordering of Packets

In DiffServ, packets from a given flow might get marked with EXP values that are different from each other but belong to the same CoS value because of in-contract and out-of-contract marking of packets. We can refer to these values of EXP bits as EXP-in and EXP-out.

If packets for EXP-in are sent on a different tunnel than packets for EXP-out, then misordering of packets within the same flows could occur. For that reason, CBTS allows operators to ensure that EXP-in and EXP-out never get mapped onto different tunnels.

The CBTS feature allows the operator to configure EXP-in and EXP-out to be transported on the same tunnel when that tunnel is up. This ensures that the feature does not introduce misordering of packets. In case of tunnel failure, the tunnel selection algorithm ensures that if EXP-in and EXP-out were carried on the same tunnel before the failure, they are still carried on a single tunnel after the failure. Thus, CBTS protects against nontransient misordering even in the event of tunnel failure.

Note CBTS does not attempt to force EXP-in and EXP-out to be carried on the same tunnel. The operator must configure CBTS so that EXP-in and EXP-out are carried on the same tunnel. This is comparable to the regular DiffServ situation, where the operator must ensure that EXP-in and EXP-out are configured to go in the same queue.

Fast Reroute and MPLS TE Class-based Tunnel Selection

CBTS allows Fast Reroute (FRR) protection on tunnels for which you configure CoS-based selection.

Note You cannot configure FRR on a primary tunnel.

CBTS operation with FRR does not change the number of or the way in which FRR backup tunnels might be used. The operation of FRR is the same as when CBTS is not activated. After you configure primary tunnels from a given headend to a given tailend, you can use FRR in the same way whether you activate CoS-based tunnel selection or not. This includes the following possibilities:

- None of the tunnels use FRR.

- All of the *x* tunnels are FRR-protected and share the same backup tunnel, if the traffic goes out the same interface.
- Some of the *x* tunnels are not FRR-protected; the remaining tunnels are FRR-protected and share the same backup tunnel, if the traffic goes out the same interface.
- Some of the *x* tunnels are not FRR-protected; the remaining tunnels are FRR-protected and are protected by different backup tunnels (for example, if the traffic goes out different interfaces, or if the traffic goes out the same interface). Bandwidth guarantees exist on the backup tunnels.

The important question for CBTS operation is only whether a tunnel interface goes down or stays up. FRR protects a given tunnel in exactly the same way as if CBTS were not configured on the tunnel.

DS-TE Tunnels and MPLS TE Class-based Tunnel Selection

CBTS operates over tunnels using DS-TE. Therefore, the tunnels on which CoS-based selection is performed can each arbitrarily and independently use a bandwidth from the global pool or the subpool.

Reoptimization and MPLS TE Class-based Tunnel Selection

CBTS allows tunnels on which CoS-based selection is performed to be reoptimized. Reoptimization does not affect CBTS operation.

Interarea and Inter-AS and MPLS TE Class-based Tunnel Selection

The CBTS operates over tunnels that are interarea when the interarea tunnels use static routes on destination prefixes or on the BGP next hops.

ATM PVCs and MPLS TE Class-based Tunnel Selection

CBTS operates over ATM permanent virtual circuits (PVCs). This means that TE or DS-TE tunnels handled by CBTS can span links that are ATM PVCs. ATM PVCs might be used on the headend device that is running CBTS and on transit label switch routers (LSRs).

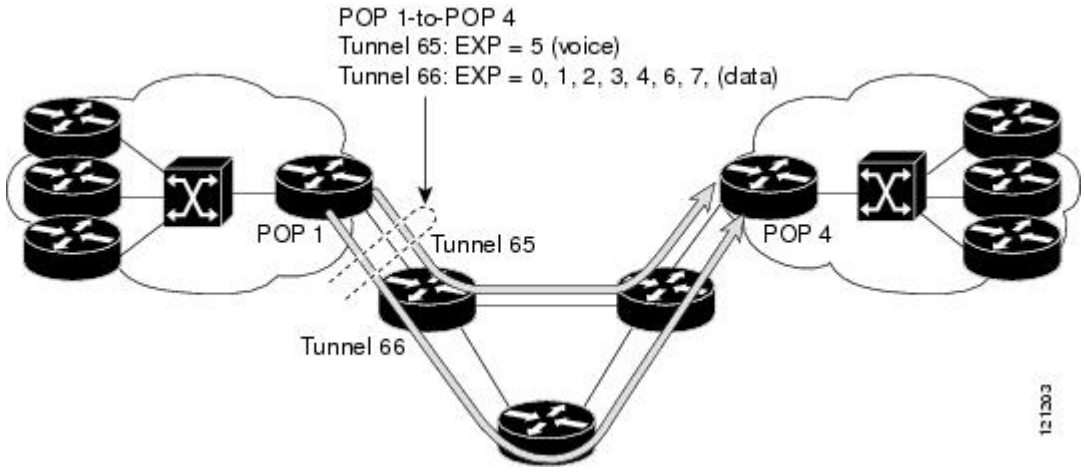
How to Configure MPLS Traffic Engineering Class-based Tunnel Selection

You need to configure the CBTS feature only on the tunnel headend. No CBTS configuration is required on the tailend or transit LSR.

Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend

The figure below shows an example of two tunnels, Tunnel 65 and Tunnel 66, transporting different classes of traffic between the same headend and the same tailend.

Figure 150: Tunnels Transporting Different Classes of Service Between the Same Headend and Tailend



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** {*hostname* | *ip-address*}
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** [**sub-pool** | **global**] *bandwidth*
8. **exit**
9. Repeat steps 3 through 8 on the same headend device to create additional tunnels from this headend to the same tailend.
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 65`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 5	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Device(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Device(config-if)# tunnel mode mpls traffic-eng`	Sets the mode of a tunnel to MPLS for TE.
Step 7	**tunnel mpls traffic-eng bandwidth** [**sub-pool** | **global**] *bandwidth* **Example:** `Device(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 3000`	Configures the bandwidth for the MPLS TE tunnel. If automatic bandwidth is configured for the tunnel, use the **tunnel mpls traffic-eng bandwidth** command to configure the initial tunnel bandwidth, which is adjusted by the autobandwidth mechanism. **Note** You can configure any existing MPLS TE command on these TE or DS-TE tunnels.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 9	Repeat steps 3 through 8 on the same headend device to create additional tunnels from this headend to the same tailend.	--
Step 10	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring EXP Values to Be Carried by Each MPLS TE or DS-TE Tunnel

For each tunnel that you create, you must indicate which EXP values the tunnel carries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **tunnel mpls traffic-eng eng** [*list-of-exp-values*] [**default**]
5. **exit**
6. Repeat steps 3 through 5 for all MPLS TE tunnels that you created in the Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend, on page 1995.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface tunnel 65`	Configures an interface type and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng eng** [*list-of-exp-values*] [**default**] **Example:** `Device(config-if)# tunnel mpls traffic-eng exp 5`	Specifies the EXP bits that will be forwarded over a member tunnel that is part of the CBTS bundle.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 6	Repeat steps 3 through 5 for all MPLS TE tunnels that you created in the Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend, on page 1995.	--
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying That the MPLS TE or DS-TE Tunnels Are Operating and Announced to the IGP

SUMMARY STEPS

1. **show mpls traffic-eng topology** {*ip-address* | **igp-id** {**isis** *nsap-address* | **ospf** *ip-address*} [**brief**]
2. **show mpls traffic-eng tunnels** *number* [**brief**] [**protection**]
3. **show ip cef summary**
4. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface*| **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
5. **show mpls traffic-eng autoroute**

DETAILED STEPS

Step 1 **show mpls traffic-eng topology** {*ip-address* | **igp-id** {**isis** *nsap-address* | **ospf** *ip-address*} [**brief**]

Use this command to display the MPLS TE global topology currently known at this node:

Example:

```
Device# show mpls traffic-eng topology
My_System_id: 0000.0025.0003.00

IGP Id: 0000.0024.0004.00, MPLS TE Id:172.16.4.4 Router Node
      link[0 ]:Intf Address: 10.1.1.4
                  Nbr IGP Id: 0000.0024.0004.02,
                admin_weight:10, affinity_bits:0x0
                max_link_bw:10000 max_link_reservable: 10000
   globalpool subpool
                total allocated reservable   reservable
                ---------------  ----------  ----------
 bw[0]:  0  1000 500
 bw[1]: 10  990 490
 bw[2]:  600  390 390
 bw[3]:  0  390 390
 bw[4]:  0  390 390
 bw[5]:  0  390 390
```

Step 2 **show mpls traffic-eng tunnels** *number* [**brief**] [**protection**]

Use this command to display information for a specified tunneling interface:

Example:

```
Device# show mpls traffic-eng tunnels 500 brief protection

Device#_t500
  LSP Head, Tunnel500, Admin: up, Oper: up
  Src 172.16.0.5, Dest 172.16.0.8, Instance 17
  Fast Reroute Protection: None
  Path Protection: 1 Common Link(s) , 1 Common Node(s)
    Primary lsp path:192.168.6.6 192.168.7.7
                     192.168.8.8 192.168.0.8
    Protect lsp path:172.16.7.7 192.168.8.8
                     10.0.0.8
    Path Protect Parameters:
```

```
  Bandwidth: 50       kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
InLabel  :  -
OutLabel : Serial5/3, 46
RSVP Signalling Info:
     Src 172.16.0.5, Dst 172.16.0.8, Tun_Id 500, Tun_Instance 18
  RSVP Path Info:
    My Address: 172.16.0.5
    Explicit Route: 192.168.7.7 192.168.8.8
    Record   Route:   NONE
    Tspec: ave rate=50 kbits, burst=1000 bytes, peak rate=50 kbits
  RSVP Resv Info:
    Record   Route:   NONE
    Fspec: ave rate=50 kbits, burst=1000 bytes, peak rate=50 kbits
```

Step 3 **show ip cef summary**

Use this command to display a summary of the IP CEF table:

Example:

```
Device# show ip cef summary
IP Distributed CEF with switching (Table Version 25), flags=0x0
 21 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 1
 21 leaves, 16 nodes, 19496 bytes, 36 inserts, 15 invalidations
 0 load sharing elements, 0 bytes, 0 references
 universal per-destination load sharing algorithm, id 5163EC15
 3(0) CEF resets, 0 revisions of existing leaves
 Resolution Timer: Exponential (currently 1s, peak 1s)
 0 in-place/0 aborted modifications
 refcounts:  4377 leaf, 4352 node
 Table epoch: 0 (21 entries at this epoch)
 Adjacency Table has 9 adjacencies
```

Step 4 **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface*| **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]

Use this command to display the contents of the MPLS Label Forwarding Information Base (LFIB):

Example:

```
Device# show mpls forwarding-table
Local Outgoing      Prefix            Bytes tag Outgoing       Next Hop
Label Label or VC   or Tunnel Id      switched  interface
26    No Label      10.253.0.0/16     0         Et4/0/0       10.27.32.4
28    1/33          10.15.0.0/16      0         AT0/0.1       point2point
29    Pop Label     10.91.0.0/16      0         Hs5/0         point2point
      1/36          10.91.0.0/16      0         AT0/0.1       point2point
30    32            10.250.0.97/32    0         Et4/0/2       10.92.0.7
      32            10.250.0.97/32    0         Hs5/0         point2point
34    26            10.77.0.0/24      0         Et4/0/2       10.92.0.7
      26            10.77.0.0/24      0         Hs5/0         point2point
35    No Label[T]   10.100.100.101/32 0         Tu301         point2point
36    Pop Label     10.1.0.0/16      0         Hs5/0         point2point
      1/37          10.1.0.0/16      0         AT0/0.1       point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

Step 5 **show mpls traffic-eng autoroute**

Use this command to display tunnels that are announced to the IGP, including interface, destination, and bandwidth:

Example:

```
Device# show mpls traffic-eng autoroute
MPLS TE autorouting enabled
destination 10.0.0.9, area ospf 10  area 0, has 4 tunnels
    Tunnel1     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel2     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel3     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel4     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
```

Configuring a Primary Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** {*hostname* | *ip-address*}
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng autoroute announce**
8. **tunnel mpls traffic-eng exp-bundle master**
9. **tunnel mpls traffic-eng exp-bundle member** *tunnel-number*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 65`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 5	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Device(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Device(config-if)# tunnel mode mpls traffic-eng`	Sets the mode of a tunnel to MPLS for TE.
Step 7	**tunnel mpls traffic-eng autoroute announce** **Example:** `Device(config-if)# tunnel mpls traffic-eng autoroute announce`	Specifies that the IGP should use the tunnel in its enhanced SPF calculation if the tunnel is up
Step 8	**tunnel mpls traffic-eng exp-bundle master** **Example:** `Device(config-if)# tunnel mpls traffic-eng exp-bundle master`	Configures a primary tunnel.
Step 9	**tunnel mpls traffic-eng exp-bundle member** *tunnel-number* **Example:** `Device(config-if)# tunnel mpls traffic-eng exp-bundle member tunnel1`	Identifies which tunnel is a member of a primary tunnel.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.

Configuration Examples for MPLS Traffic Engineering Class-based Tunnel Selection

Example: Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend

The following example shows how to create multiple MPLS TE or DS-TE tunnels from the same headend to the same tailend:

```
Device(config)# interface Tunnel 65

Device(config-if)# ip numbered loopback 0
Device(config-if)# tunnel destination 10.1.1.1

Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000

Device(config-if)# ^Z
Device(config)# interface Tunnel 66

Device(config-if)# ip numbered loopback 0
Device(config-if)# tunnel destination 10.1.1.1

Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel mpls traffic-eng bandwidth 50000
Device(config-if)# end
Device#
```

Example: Configuring EXP Values to Be Carried by Each MPLS TE or DS-TE Tunnel

The following example shows how to configure EXP values to be carried by each MPLS TE or DS-TE tunnel that you created:

```
Device(config)# interface Tunnel 65

Device(config-if)# tunnel mpls traffic-eng exp 5
Device(config-if)# ^Z
Device(config)#
Device(config)# interface Tunnel 66

Device(config-if)# tunnel mpls traffic-eng exp 0 1 2 3 4 6 7
Device(config-if)# end
Device#
```

Example: Verifying That the MPLS TE or DS-TE Tunnels Are Operating and Announced to the IGP

The output for each of the following examples helps verify that the MPLS TE or DS-TE tunnels are operating and visible.

The **show mpls traffic-eng topology** command output displays the MPLS TE global topology:

```
Device# show mpls traffic-eng topology 10.0.0.1
IGP Id: 10.0.0.1, MPLS TE Id:10.0.0.1 Router Node  (ospf 10  area 0) id 1
      link[0]: Broadcast, DR: 10.0.1.2, nbr_node_id:6, gen:18
   frag_id 0, Intf Address:10.1.1.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                          Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:             0             1000                0
 bw[1]:             0             1000                0
 bw[2]:             0             1000                0
 bw[3]:             0             1000                0
 bw[4]:             0             1000                0
 bw[5]:             0             1000                0
 bw[6]:             0             1000                0
 bw[7]:           100              900                0
      link[1]: Broadcast, DR: 10.0.2.2, nbr_node_id:7, gen:19
   frag_id 1, Intf Address:10.0.2.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                          Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:             0             1000                0
 bw[1]:             0             1000                0
 bw[2]:             0             1000                0
 bw[3]:             0             1000                0
 bw[4]:             0             1000                0
 bw[5]:             0             1000                0
 bw[6]:             0             1000                0
 bw[7]:           300              700                0
Device#
Device# show mpls traffic-eng topology 10.0.0.9
IGP Id: 10.0.0.9, MPLS TE Id:10.0.0.9 Router Node  (ospf 10  area 0) id 3
      link[0]: Point-to-Point, Nbr IGP Id: 10.0.0.5, nbr_node_id:5, gen:9
   frag_id 1, Intf Address:10.0.5.2, Nbr Intf Address:10.0.5.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 155000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                          Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:             0             1000                0
 bw[1]:             0             1000                0
```

```
 bw[2]:            0             1000               0
 bw[3]:            0             1000               0
 bw[4]:            0             1000               0
 bw[5]:            0             1000               0
 bw[6]:            0             1000               0
 bw[7]:            0             1000               0
      link[1]: Point-to-Point, Nbr IGP Id: 10.0.0.7, nbr_node_id:4, gen:9
   frag_id 0, Intf Address:10.0.6.2, Nbr Intf Address:10.0.6.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 155000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                       Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:            0             1000               0
 bw[1]:            0             1000               0
 bw[2]:            0             1000               0
 bw[3]:            0             1000               0
 bw[4]:            0             1000               0
 bw[5]:            0             1000               0
 bw[6]:            0             1000               0
 bw[7]:            0             1000               0
Device#
```

The **show mpls traffic-eng tunnels** command output displays information about a tunnel:

```
Device# show mpls traffic-eng tunnels tunnel1
Name: Router_t1                           (Tunnel1) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path1 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/0, 12304
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 1, Tun_Instance 10
    RSVP Path Info:
      My Address: 10.0.1.1
      Explicit Route: 10.0.1.2 10.0.3.2 10.0.5.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 180.0.2.2 10.0.3.2 180.0.5.2
                    10.0.0.9
  History:
    Tunnel:
      Time since created: 15 minutes, 18 seconds
      Time since path change: 15 minutes, 5 seconds
    Current LSP:
      Uptime: 15 minutes, 5 seconds
Device# show mpls traffic-eng tunnel tunnel2
```

```
Name: Router_t2                           (Tunnel2) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path2 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/1, 12305
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 2, Tun_Instance 10
    RSVP Path Info:
      My Address: 10.0.2.1
      Explicit Route: 10.0.2.2  10.0.4.2  10.0.6.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 10.0.2.2 10.0.3.2 10.0.5.2
                    10.0.0.9
  History:
    Tunnel:
      Time since created: 15 minutes, 19 seconds
      Time since path change: 15 minutes, 6 seconds
    Current LSP:
      Uptime: 15 minutes, 6 seconds
Device# show mpls traffic-eng tunnels tunnel3
Name: Router_t3                           (Tunnel3) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path2 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/1, 12306
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 3, Tun_Instance 8
    RSVP Path Info:
      My Address: 10.0.2.1
'     Explicit Route: 10.0.2.2 10.0.4.2 10.0.6.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 10.0.2.2 10.0.3.2 10.0.5.2
                    10.0.0.9
  History:
```

```
    Tunnel:
      Time since created: 15 minutes, 19 seconds
      Time since path change: 15 minutes, 7 seconds
    Current LSP:
      Uptime: 15 minutes, 7 seconds
Device# show mpls traffic-eng tunnels tunnel4
Name: Router_t4                           (Tunnel4) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path2 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/1, 12307
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 4, Tun_Instance 6
    RSVP Path Info:
      My Address: 10.0.2.1
      Explicit Route: 10.0.2.2 10.0.4.2 10.0.6.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 10.0.2.2 10.0.3.2 10.0.5.2
                    10.0.0.9
  History:
    Tunnel:
      Time since created: 15 minutes, 20 seconds
      Time since path change: 15 minutes, 8 seconds
    Current LSP:
      Uptime: 15 minutes, 8 seconds
```

The **show ip cef detail** command output displays detailed FIB entry information for a tunnel:

```
Device# show ip cef tunnel1 detail
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel1, 0 dependencies
```

```
    traffic share 1
    next hop 0.0.0.0, Tunnel1
    valid adjacency
    tag rewrite with Tu1, point2point, tags imposed {12304}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
Device# show ip cef tunnel2 detail
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel2, 0 dependencies
    traffic share 1
    next hop 0.0.0.0, Tunnel2
    valid adjacency
    tag rewrite with Tu2, point2point, tags imposed {12305}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
Device# show ip cef tunnel3 detail
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel3, 0 dependencies
    traffic share 1
    next hop 0.0.0.0, Tunnel3
    valid adjacency
    tag rewrite with Tu3, point2point, tags imposed {12306}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
Device# show ip cef tunnel4 detail
```

```
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel4, 0 dependencies
    traffic share 1
    next hop 0.0.0.0, Tunnel4
    valid adjacency
    tag rewrite with Tu4, point2point, tags imposed {12307}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
```

The **show mpls forwarding-table detail** command output displays detailed information from the MPLS LFIB:

```
Device# show mpls forwarding-table detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
Device#
Device# show mpls forwarding-table 10.0.0.9 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
Tun hd Untagged    10.0.0.9/32       0          Tu1        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12304}, via Fa6/0
 00027D884000000ED70178A88847 03010000
 No output feature configured
        Per-exp selection: 1
       Untagged    10.0.0.9/32       0          Tu2        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12305}, via Fa6/1
 00027D884001000ED70178A98847 03011000
 No output feature configured
        Per-exp selection: 2  3
       Untagged    10.0.0.9/32       0          Tu3        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12306}, via Fa6/1
 00027D884001000ED70178A98847 03012000
 No output feature configured
        Per-exp selection: 4  5
       Untagged    10.0.0.9/32       0          Tu4        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12307}, via Fa6/1
 00027D884001000ED70178A98847 03013000
 No output feature configured
        Per-exp selection: 0  6  7
Device#
```

The **show mpls traffic-eng autoroute** command output displays tunnels that are announced to the IGP:

```
Device# show mpls traffic-eng autoroute
```

```
MPLS TE autorouting enabled
  destination 10.0.0.9, area ospf 10  area 0, has 4 tunnels
    Tunnel1     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel2     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel3     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel4     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
Device#
```

Example: Configuring a Primary Tunnel

The following example specifies that there is a primary tunnel that includes tunnels Tunnel20000 through Tunnel20005:

```
interface Tunnel 200
 ip unnumbered Loopback 0
 tunnel destination 10.10.10.10
 tunnel mode mpls traffic-eng
 tunnel mode mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng exp-bundle master
 tunnel mpls traffic-eng exp-bundle member Tunnel20000
 tunnel mpls traffic-eng exp-bundle member Tunnel20001
 tunnel mpls traffic-eng exp-bundle member Tunnel20002
 tunnel mpls traffic-eng exp-bundle member Tunnel20003
 tunnel mpls traffic-eng exp-bundle member Tunnel20004
 tunnel mpls traffic-eng exp-bundle member Tunnel20005
```

Additional References

Related Documents

Related Topic	Document Title
MPLS traffic engineering commands	*Multiprotocol Label Switching Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering Class-based Tunnel Selection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 171: Feature Information for MPLS Traffic Engineering Class-based Tunnel Selection

Feature Name	Releases	Feature Configuration Information
MPLS Traffic Engineering : Class-based Tunnel Selection	12.0(29)S 12.2(33)SRA 12.2(32)SY 12.2(33)SXH 12.4(20)T Cisco IOS XE Release 3.6S	The MPLS Traffic Engineering (TE): Class-based Tunnel Selection feature enables you to dynamically route and forward traffic with different class of service (CoS) values onto different TE tunnels between the same tunnel headend and the same tailend. The TE tunnels can be regular TE or DiffServ-aware TE (DS-TE) tunnels. In 12.0(29)S, this feature was introduced. In 12.2(33)SRA, this feature was integrated and the following commands were added: • **tunnel mpls traffic-eng exp-bundle master** • **tunnel mpls traffic-eng exp-bundle member** • **show mpls traffic-eng exp** 12.0(32)SY, support for this feature was added on the Cisco 12000 family of routers. In 12.2(33)SXH, this feature was integrated. In 12.4(20)T, this feature was integrated. In Cisco IOS XE Release 3.6S, this feature was integrated.

Glossary

BGP --Border Gateway Protocol. Interdomain routing protocol that replaces External Gateway Protocol (EGP). BGP exchanges reachability information with other BGP systems. It is defined by RFC 116.3

bundled tunnels--Members of a primary tunnel. You define the EXP bits that will be forwarded over each bundled tunnel.

Cisco Express Forwarding--An advanced Layer 3 IP switching technology. Cisco Express Forwarding optimizes network performance and scalability for networks with large and dynamic traffic patterns, such as the Internet and networks characterized by intensive web-based applications or interactive sessions.

CoS --class of service. An indication of how an upper-layer protocol requires a lower-layer protocol to treat its messages. In Systems Network Architecture (SNA) subarea routing, CoS definitions are used by subarea nodes to determine the optimal route for establishing a given session. A CoS definition comprises a virtual route number and a transmission priority field. Also called type of service (ToS).

DS-TE --DiffServ-aware traffic engineering. The configuring of two bandwidth pools on each link, a global pool and a subpool. Multiprotocol Label Switching (MPLS) traffic engineering tunnels using the subpool bandwidth can be configured with quality of service (QoS) mechanisms to deliver guaranteed bandwidth services end-to-end across the network. Simultaneously, tunnels using the global pool can convey DiffServ traffic.

EXP --experimental field or bits. A 3-bit field in the Multiprotocol Label Switching (MPLS) header widely known as the EXP field or EXP bits because, according to RFC 3032, that field is reserved for experimental use. However, the most common use of those bits is for quality of service (QoS) purposes.

headend --The upstream, transmitting end of a tunnel. This is the first device in the label switched path (LSP).

LSP --label switched path. A sequence of hops (R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A label switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

master tunnel--A set of tunnels that have the same destination.

MPLS traffic engineering--Multiprotocol Label Switching traffic engineering. A constraint-based routing algorithm for routing label switched path (LSP) tunnels.

MQC --modular quality of service (QoS) command-line interface (CLI). A CLI structure that allows users to create traffic polices and attach those polices to interfaces.

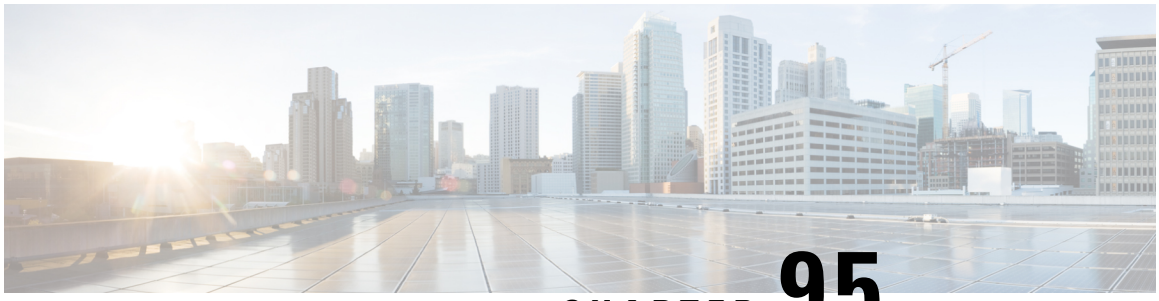
PBR --policy-based routing. A routing scheme in which packets are forwarded to specific interfaces based on user-configured policies. A policy might specify, for example, that traffic sent from a particular network should be forwarded out one interface, and all other traffic should be forwarded out another interface.

tailend --The downstream, receiving end of a tunnel. The device that terminates the traffic engineering label switched path (LSP).

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

ToS --type of service. See CoS.

tunnel --A secure communication path between two peers. A traffic engineering tunnel is a label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.



CHAPTER 95

MPLS Traffic Engineering Interarea Tunnels

The MPLS Traffic Engineering: Interarea Tunnels feature allows you to establish Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels that span multiple Interior Gateway Protocol (IGP) areas and levels, removing the restriction that had required the tunnel headend and tailend routers both be in the same area. The IGP can be either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).

Prerequisites for MPLS Traffic Engineering Interarea Tunnels

Your network must support the following software features:

- MPLS
- IP Cisco Express Forwarding
- IS-IS or OSPF
- TE tunnels

Restrictions for MPLS Traffic Engineering Interarea Tunnels

- The dynamic path option feature for TE tunnels (which is specified in the **tunnel mpls traffic-eng path-option** *number* **dynamic** command) is not supported for interarea tunnels. An explicit path identifying the Area Border Routers (ABRs) is required. When there are choices for the ABRs to be used, multiple explicit paths are recommended, each of which identifies a different sequence of ABRs.
- The MPLS TE AutoRoute feature (which is specified in the **tunnel mpls traffic-eng autoroute announce** command) is not supported for interarea tunnels because you would need to know the network topology behind the tailend router.

- Tunnel affinity (the **tunnel mpls traffic-eng affinity** command) is not supported for interarea tunnels.
- The reoptimization of tunnel paths is not supported for interarea tunnels.
- MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

Information About MPLS Traffic Engineering Interarea Tunnels

Interarea Tunnels Functionality

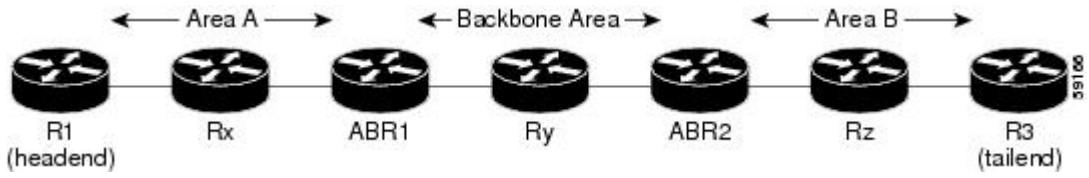
To configure an interarea tunnel, you specify on the headend router a loosely routed explicit path for the tunnel label switched path (LSP) that identifies each ABR the LSP should traverse using the **next-address loose** command. The headend router and the ABRs along the specified explicit path expand the loose hops, each computing the path segment to the next ABR or tunnel destination.

For example, to configure a TE tunnel from router R1 to router R3 in the simple multiarea network shown in the figure below, you would specify ABR1 and ABR2 as loose hops in the explicit path for the tunnel.

Note Rx can be configured as a loose hop as well. In that case, the headend router R1 computes the path to Rx and router Rx computes the path to ABR1.

To signal the tunnel LSP, the headend router (R1) computes the path to ABR1 and sends a Resource Reservation Protocol (RSVP) Path message specifying the path from itself to ABR1 as a sequence of strict hops followed by the path from ABR1 to the tailend as a sequence of loose hops (ABR2, R3). When ABR1 receives the Path message, it expands the path across the backbone area to ABR2 and forwards the Path message specifying the path from itself to ABR2 as a sequence of strict hops followed by the path from ABR2 to the tunnel tailend (R3) as a loose hop. When ABR2 receives the Path message, it expands the path across the tailend area to R3 and propagates the Path message specifying the path from itself to R2 as a sequence of strict hops.

Figure 151: Multiarea Network



Note Strictly speaking, IS-IS does not have the notion of an ABR. For the purpose of discussing the MPLS Traffic Engineering: Interarea Tunnels feature, an IS-IS level-1-2 router is considered to be an ABR.

Note The explicit path for a TE interarea tunnel may contain any number of non-ABR LSPs. Within an area, a combination of loose and strict next IP addresses is allowed. To specify the next IP address in the explicit path, use the **next-address** command.

Note With OSPF, if an area is connected to the backbone through a virtual link, there may be more than two ABRs in the path.

The following MPLS TE features are supported on interarea traffic engineering LSPs:

- Automatic bandwidth adjustment
- Diff-Serve-aware traffic engineering
- Fast reroute link protection
- Policy-based routing
- Static routing

Autoroute Destination Functionality

The autoroute destination feature allows you to automatically route traffic through a TE tunnel instead of manually configuring static routes.

You enable this feature on a per-tunnel basis by using the **tunnel mpls traffic-eng autoroute destination** command.

The following sections describe how the autoroute destination feature interacts with other features:

CBTS Interaction with Autoroute Destination

TE tunnels that have the autoroute destination feature enabled can also be configured as class-based traffic shaping (CBTS) tunnel bundle primarys or members. Within a CBTS bundle, only the primary tunnel with autoroute destination enabled is installed into the Routing Information Base (RIB); that is, the member tunnels are not installed into the RIB.

If member tunnels that have autoroute destination enabled are unconfigured from the bundle, they become regular TE tunnels and TE requests that the static process installs static routes over those tunnels in the RIB. Conversely, when regular TE tunnels with autoroute destination enabled are added to a CBTS bundle as members, TE requests that the static process removes the automatic static routes over those tunnels from the RIB.

Manually Configured Static Routes Interaction with Autoroute Destination

If there is a manually configured static route to the same destination as a tunnel with autoroute destination enabled via the **tunnel mpls traffic-eng autoroute destination** command, traffic for that destination is load-shared between the static route and the tunnel with autoroute destination enabled.

Autoroute Announce Interaction with Autoroute Destination

For intra-area tunnels, if a tunnel is configured with both autoroute announce and autoroute destination, the tunnel is announced to the RIB by both the IGP and the static process. RIBs prefer static routes, not IGP routes, so the autoroute destination features takes precedence over autoroute announce.

Forwarding Adjacency Interaction with Autoroute Destination

If a tunnel is configured with both forwarding adjacency and autoroute destination, the tunnel is announced to the RIB by both the IGP and the static process. The RIB prefers the static route. However, because the IGP was notified about the tunnel via the **forwarding adjacency** command and the tunnel information was flooded, forwarding adjacency continues to function.

MPLS Traffic Engineering Interarea Tunnels Benefits

- When it is desirable for the traffic from one router to another router in a different IGP area to travel over TE LSPs, the MPLS Traffic Engineering: Interarea Tunnels feature allows you to configure a tunnel that runs from the source router to the destination router. The alternative would be to configure a sequence of tunnels, each crossing one of the areas between source and destination routers such that the traffic arriving on one such tunnel is forwarded into the next such tunnel.
- The autoroute destination feature prevents you from having to manually configure static routes to route traffic over certain interarea tunnels such as ASBRs.

How to Configure MPLS Traffic Engineering Interarea Tunnels

Note You must configure either OSPF or IS-IS.

Configuring OSPF for Interarea Tunnels

Configuring OSPF for ABR Routers

For each ABR that is running OSPF, perform the following steps to configure traffic engineering on each area you want tunnels in or across. By having multiple areas and configuring traffic engineering in and across each area, the router can contain changes within the network within an area.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **router ospf** *process-id*
4. **network** *ip-address wildcard-mask* **area** *area-id*
5. **mpls traffic-eng router-id** *interface-name*
6. **mpls traffic-eng area 0**
7. **mpls traffic-eng area** *number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Enables OSPF and enters router configuration mode. The *process-id* argument is an internally used identification parameter for the OSPF routing process. It is logically assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(config-router)# network 192.168.45.0 0.0.255.255 area 1`	Specifies the interfaces on which OSPF is to run and specifies the area to which the interface is connected.
Step 5	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with a given interface. The router identifier is displayed in the **show mpls traffic-eng topology path** command output. **Note** The *interface-name* value must be Loopback0.
Step 6	**mpls traffic-eng area 0** **Example:** `Router(config-router)# mpls traffic-eng area 0`	Turns on MPLS traffic engineering for OSPF in area 0. **Note** To display the MPLS TE global topology currently known at this node, use the **show mpls traffic-eng topology** command.
Step 7	**mpls traffic-eng area** *number* **Example:** `Router(config-router)# mpls traffic-eng area 2`	Configures a router running OSPF MPLS to flood traffic engineering for the indicated OSPF area.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.

Configuring OSPF for Non-ABR Routers

For each non-ABR that is running OSPF, perform the following steps to configure OSPF.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **network** *ip-address wildcard-mask* **area** *area-id*
5. **mpls traffic-eng router-id** *interface-name*
6. **mpls traffic-eng area** *number*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Enables OSPF and enters router configuration mode. The *process-id* argument is an internally used identification parameter for the OSPF routing process. It is locally assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:**	Specifies the interfaces on which OSPF is to run and specifies the area to which the interface is connected.

	Command or Action	Purpose
	`Router(config-router)# network 192.168.10.10 255.255.255.0 area 1`	
Step 5	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with a given interface. The router identifier is displayed in the **show mpls traffic-eng topology path** command output. **Note** The *interface-name* value must be Loopback0.
Step 6	**mpls traffic-eng area** *number* **Example:** `Router(config-router)# mpls traffic-eng area 1`	Specifies the area that the router is in. **Note** To display the MPLS TE global topology currently known at this node, use the **show mpls traffic-eng topology** command.
Step 7	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.

Configuring IS-IS for Interarea Tunnels

Configuring IS-IS for Backbone Routers

To configure IS-IS for background (level-1-2) routers, perform the following steps.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis**
4. **metric-style wide**
5. **net** *nn.nnnn.nnnn.nnnn.nnnn*
6. **mpls traffic-eng router-id** *interface-name*
7. **mpls traffic-eng level-1**
8. **mpls traffic-eng level-2**
9. **interface** *typeslot* / *port*
10. **ip router isis**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** **Example:** `Router(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process for IP, and places the router in router configuration mode.
Step 4	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router to generate and accept only new-style type, length, value objects (TLVs).
Step 5	**net** *nn.nnnn.nnnn.nnnn.nnnn* **Example:** `Router(config-router)# net 10.0000.0100.0000.0010`	Configures the area ID (area address) and the system ID.
Step 6	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface Loopback0.
Step 7	**mpls traffic-eng level-1** **Example:** `Router(config-router)# mpls traffic-eng level-1`	Turns on MPLS traffic engineering for IS-IS at level 1. **Note** To display the MPLS TE global topology currently known at this node, use the **show mpls traffic-eng topology** command.
Step 8	**mpls traffic-eng level-2** **Example:** `Router(config-router)# mpls traffic-eng level-2`	Turns on MPLS traffic engineering for IS-IS at level 2. **Note** To display the MPLS TE global topology currently known at this node, use the **show mpls traffic-eng topology** command.
Step 9	**interface** *typeslot* / *port* **Example:**	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config-router)# interface POS1/1/0`	
Step 10	**ip router isis** **Example:** `Router(config-if)# ip router isis`	Enables IS-IS routing. Specify this command on each interface on which you want to run IS-IS.
Step 11	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring IS-IS for Nonbackbone Routers

To configure IS-IS for nonbackbone routers, perform the following steps.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis**
4. **metric-style wide**
5. **net** *nn.nnnn.nnnn.nnnn.nnnn*
6. **mpls traffic-eng router-id** *interface-name*
7. **mpls traffic-eng** {**level-1** | **level-2**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router isis** **Example:** `Router(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process for IP, and places the router in router configuration mode.
Step 4	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router to generate and accept only new-style TLVs.
Step 5	**net** *nn.nnnn.nnnn.nnnn.nnnn* **Example:** `Router(config-router)# net 10.0000.2000.0100.0001`	Configures the area ID (area address) and the system ID.
Step 6	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface Loopback0.
Step 7	**mpls traffic-eng** {**level-1** | **level-2**} **Example:** `Router(config-router)# mpls traffic-eng level-1`	Turns on MPLS traffic engineering for IS-IS at level 1. **Note** To display the MPLS TE global topology currently known at this node, use the **show mpls traffic-eng topology** command.
Step 8	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.

Configuring IS-IS for Interfaces

To configure IS-IS for interfaces, perform the following steps.

Note MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis**
4. **metric-style wide**

5. **net** *nn.nnnn.nnnn.nnnn.nnnn*
6. **mpls traffic-eng router-id** *interface-name*
7. **interface** *typeslot /port*
8. **ip router isis**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** **Example:** `Router(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process for IP. This command places the router in router configuration mode.
Step 4	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router to generate and accept only new-style TLVs.
Step 5	**net** *nn.nnnn.nnnn.nnnn.nnnn* **Example:** `Router(config-router)# net 10.0000.0100.0000.0010`	Configures the area ID (area address) and the system ID.
Step 6	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface Loopback0.
Step 7	**interface** *typeslot /port* **Example:** `Router(config-router)# interface POS1/1/0`	Specifies the interface and enters interface configuration mode.
Step 8	**ip router isis** **Example:**	Enables IS-IS routing.

	Command or Action	Purpose
	`Router(config-if)# ip router isis`	Specify this command on each interface on which you want to run IS-IS.
Step 9	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring MPLS and RSVP to Support Traffic Engineering

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **mpls traffic-eng tunnels**
5. **interface** *typeslot* / *port*
6. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
7. **ip rsvp bandwidth**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Router(config)# ip cef`	Enables Cisco Express Forwarding on the Route Processor card.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config)# mpls traffic-eng tunnels`	Enables MPLS traffic engineering tunnel signaling on a device.

	Command or Action	Purpose
Step 5	**interface** *typeslot* / *port* **Example:** `Router(config)# interface Loopback0`	Specifies the interface and enters interface configuration mode.
Step 6	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Router(config-if)# ip address 192.168.10.10 255.255.255.255`	Assigns an IP network address and network mask to the interface.
Step 7	**ip rsvp bandwidth** **Example:** `Router(config-if)# ip rsvp bandwidth`	Enables RSVP for IP on an interface.
Step 8	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Interarea Tunnel

Configuring an MPLS Traffic Engineering Interarea Tunnel to Use Explicit Paths

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *tunnel-interface*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** *number* **explicit** {**name** *path-name* | **identifier** *path-number*} [**lockdown**]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *tunnel-interface* **Example:** `Router(config)# interface Tunel1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered Loopback 0`	Gives the tunnel interface an IP address. An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.20.20`	Specifies the destination for a tunnel. You must enter the MPLS traffic engineering router ID of the destination device.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 300`	Configures the bandwidth required for the MPLS traffic engineering tunnel.
Step 8	**tunnel mpls traffic-eng path-option** *number* **explicit** {**name** *path-name* | **identifier** *path-number*} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit name path-Tunnel1`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. The **name**keyword must specify the ABRs the tunnel LSP must traverse as loose hops via the **next-address loose** command.
Step 9	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring Explicit Paths

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip explicit-path name** *pathname*
4. **next-address** [**loose** | **strict**] *ip-address*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip explicit-path name** *pathname* **Example:** `Router(config)# ip explicit-path name path-tunnel1`	Enters IP explicit path configuration mode and creates or modifies the specified path.
Step 4	**next-address** [**loose**	**strict**] *ip-address* **Example:** `Router(config-ip-expl-path)# next-address loose 192.168.40.40`	Specifies the next IP address in the explicit path. In a **next-address loose** command you must specify each ABR the path must traverse.
Step 5	**end** **Example:** `Router(config-ip-expl-path)# end`	Returns to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel with Autoroute Destination

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *tunnel-interface*

4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** *number* **explicit** {**name** *path-name* | **identifier** *path-number*} [**lockdown**]
9. **tunnel mpls traffic-eng autoroute destination**
10. **end**

DETAILED STEPS

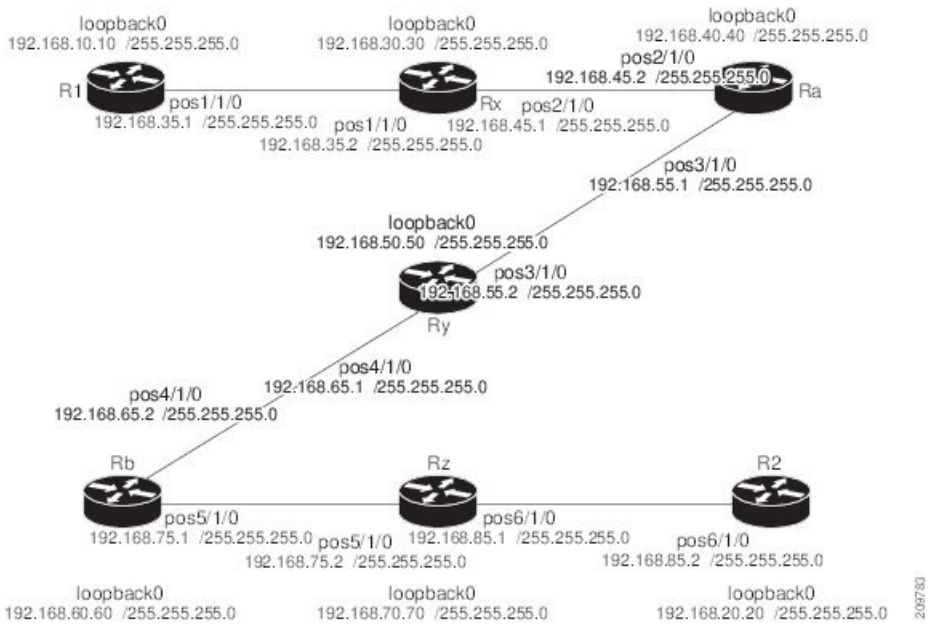
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *tunnel-interface* **Example:** `Router(config)# interface Tunnel1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered Loopback 0`	Gives the tunnel interface an IP address. An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.20.20`	Specifies the destination for a tunnel. You must enter the MPLS traffic engineering router ID of the destination device.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 300`	Configures the bandwidth required for the MPLS traffic engineering tunnel.

	Command or Action	Purpose
Step 8	**tunnel mpls traffic-eng path-option** *number* **explicit** {**name** *path-name* | **identifier** *path-number*} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit name path-Tunnel1`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. The **name**keyword must specify the ABRs the tunnel LSP must traverse as loose hops via the **next-address loose** command.
Step 9	**tunnel mpls traffic-eng autoroute destination** **Example:** `Router(config-if)# tunnel mpls traffic-eng autoroute destination`	Automatically routes traffic through a TE tunnel.
Step 10	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for MPLS Traffic Engineering Interarea Tunnels

This section shows how to configure MPLS traffic engineering interarea tunnels for the simple router topology illustrated in the figure below. It includes configuration fragments that illustrate the configurations shown in the following sections:

Figure 152: Router Topology



Configuring OSPF for Interarea Tunnels Example

The following configuration fragments show how to configure OSPF for interarea tunnels assuming that:

- Routers R1, Rx, and Ra are in OSPF Area 1
- Routers Ra, Ry, and Rb are in OSPF Area 0
- Routers Rb, Rz, and R2 are in OSPF Area 2
- Router Ra is an ABR for Area 0 and Area 1
- Router Rb is an ABR for Area 0 and Area 2

Router R1 OSPF Configuration

```
router ospf 1
 network 192.168.10.10 0.0.0.0 area 1
 network 192.168.35.0 0.0.0.255 area 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 1
```

Router Rx OSPF Configuration

```
router ospf 1
 network 192.168.30.30 0.0.0.0 area 1
 network 192.168.35.0 0.0.0.255 area 1
 network 192.168.45.0 0.0.0.255 area 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 1
```

Router Ra OSPF Configuration

Ra is an ABR for Area 0 and Area 1. Interface POS2/1/0 is in Area 1 and interface POS3/1/0 is in Area 0. The **mpls traffic-eng area** commands configure Ra for IGP TE updates for both areas.

```
router ospf 1
 network 192.168.40.40 0.0.0.0 area 0
 network 192.168.45.0 0.0.0.255 area 1
 network 192.168.55.0 0.0.0.255 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng area 1
```

Router Rb OSPF Configuration

Rb is an ABR for Area 0 and Area 2. Interface POS4/1/0 is in Area 0 and interface POS5/1/0 is in Area 2. The **mpls traffic-eng area** commands configure Rb for IGP TE updates for both areas.

```
router ospf 1
 network 192.168.60.60 0.0.0.0 area 0
 network 192.168.65.0 0.0.0.255 area 0
 network 192.168.75.0 0.0.0.255 area 2
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng area 2
```

Router Rz OSPF Configuration

```
router ospf 1
 network 192.168.70.70 0.0.0.0 area 2
 network 192.168.75.0 0.0.0.255 area 2
 network 192.168.85.0 0.0.0.255 area 2
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 2
```

Router R2 OSPF Configuration

```
router ospf 1
 network 192.168.20.20 0.0.0.0 area 2
 network 192.168.85.0 0.0.0.255 area 2
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 2
```

Configuring IS-IS for Interarea Tunnels Example

The following configuration fragments illustrate how to configure IS-IS for interarea tunnels assuming that:

- R1 and Rx are level-1 routers
- Ra, Ry, and Rb are level-1-2 routers
- Rz and R2 are level-1 routers

Router R1 IS-IS Configuration

```
interface POS1/1/0
 ip router isis
router isis
 metric-style wide
 net 10.0000.0100.0000.0010
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Router Rx IS-IS Configuration

```
clns routing
interface POS1/1/0
 ip router isis
interface POS2/1/0
 ip router isis
router isis
 metric-style wide
 net 10.0000.2000.0100.0001
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Router Ra IS-IS Configuration

```
clns routing
interface POS2/1/0
 ip router isis
interface POS3/1/0
```

```
 ip router isis
router isis
 metric-style wide
 net 10.0000.2000.0200.0002
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 mpls traffic-eng level-2
```

Router Ry IS-IS Configuration

```
clns routing
interface POS3/1/0
 ip router isis
interface POS4/1/0
 ip router isis
router isis
 metric-style wide
 net 10.0000.2000.0300.0003
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-2
```

Router Rb IS-IS Configuration

```
clns routing
interface POS4/1/0
 ip router isis
interface POS5/1/0
 ip router isis
router isis
 metric-style wide
 net 10.0000.2000.0400.0004
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 mpls traffic-eng level-2
```

Router Rz IS-IS Configuration

```
clns routing
interface POS5/1/0
 ip router isis
interface POS6/1/0
 ip router isis
router isis
 metric-style wide
 net 10.0000.2000.0500.0005
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Router R2 IS-IS Configuration

```
clns routing
interface POS6/1/0
 ip router isis
router isis
 metric-style wide
 net 10.0000.0200.0000.0020
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Configuring MPLS and RSVP to Support Traffic Engineering Example

The following configuration fragments show how to configure MPLS and RSVP to support traffic engineering on the routers.

Router R1 Traffic Engineering Configuration

```
ip cef
mpls traffic-eng tunnels
interface Loopback0
 ip address 192.168.10.10 255.255.255.255
interface POS1/1/0
!Each interface supporting MPLS TE must include the following:
 mpls traffic-eng tunnels
 ip rsvp bandwidth
```

The configuration of routers Rx, Ra, Ry, Rb, Rz, and R2 for traffic engineering operation is similar to that for R1.

Configuring an MPLS Traffic Engineering Interarea Tunnel Example

The following configuration fragments show how to configure an MPLS traffic engineering interarea tunnel. Tunnel1 is configured with a path option that is loosely routed through Ra and Rb.

R1 Interarea Tunnel Configuration

The following commands configure an MPLS TE tunnel to use explicit paths:

```
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 192.168.20.20
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 300
 tunnel mpls traffic-eng path-option 1 explicit name path-tunnel1
```

The following commands configure an explicit path:

```
ip explicit-path name path-tunnel1
 next-address loose 192.168.40.40
 next-address loose 192.168.60.60
 next-address loose 192.168.20.20  !Specifying the tunnel tailend in the loosely routed
!path is optional.
```

Note Generally for an interarea tunnel you should configure multiple loosely routed path options that specify different combinations of ABRs (for OSPF) or level-1-2 boundary routers (for IS-IS) to increase the likelihood that the tunnel will be successfully signaled. In this simple topology there are no other loosely routed paths.

Configuring an MPLS Traffic Engineering Tunnel with Autoroute Destination Example

The following example shows how to configure an MPLS TE tunnel with autoroute destination:

```
interface Tunnel103
 ip unnumbered Loopback0
 tunnel destination 10.1.0.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-end path-option 1 explicit name 111-103
 tunnel mpls traffic-eng autoroute destination
```

Additional References

Related Documents

Related Topic	Document Title
IS-IS	• Integrated IS-IS Routing Protocol Overview • *Cisco IOS IP Routing Protocols Command Reference*
Link protection	MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection)
MPLS traffic engineering commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
OSPF	• *Cisco IOS IP Routing Protocols Command Reference* • Configuring OSPF

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering Interarea Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 172: Feature Information for MPLS Traffic Engineering Interarea Tunnels

Feature Name	Releases	Feature Information
MPLS Traffic Engineering: Interarea Tunnels	12.0(19)ST1 12.0(21)ST 12.2(18)S 12.2(18)SXD 12.2(27)SBC 12.2(28)SB 12.2(33)SRB 12.4(20)T 12.2(33)SRE 15.2(1)S Cisco IOS-XE Release 3.5	The MPLS Traffic Engineering: Interarea Tunnels feature allows you to establish MPLS TE tunnels that span multiple IGP areas and levels, removing the restriction that had required the tunnel headend and tailend routers both to be in the same area. In 12.2(33)SRB, support was added for stateful switchover (SSO) recovery of LSPs that include loose hops. In 12.4(20)T, support was eliminated for SSO recovery of LSPs that include loose hops. In 12.2(33)SRE, the MPLS-TE Autoroute Destinations feature was added. In 15.2(1)S the MPLS-TE Autoroute Destinations feature was added. In Cisco IOS-XE Release 3.5, the MPLS-TE Autoroute Destinations feature was added. The following commands were introduced or modified: **show ip static route**, **show mpls traffic-eng autoroute**, **show mpls traffic-eng tunnels**, **tunnel mpls traffic-eng autoroute destination**.

Glossary

ABR --Area Border Router.A router connecting two areas. In OSPF, ABRs belong to both areas and must maintain separate topological databases for each. When an OSPF router has interfaces in more than one area, it is an Area Border Router.

area --A logical set of network segments (for example, one that is OSPF-based) and their attached devices. Areas usually are connected to other areas by routers, making up a single autonomous system. OSPF and IS-IS define their areas differently. OSPF area borders are marked by routers. Some interfaces are in one area, and other interfaces are in another area. With IS-IS, all the routers are completely within an area, and the area borders are on links, not on routers. The routers that connect the areas are level-2 routers, and routers that have no direct connectivity to another area are level-1 routers.

area ID --In an IS-IS router, this area address is associated with the entire router rather than an interface. A router can have up to three area addresses. Both the area ID and the system ID are defined on an IS-IS router by a single address, the Network Entry Title (NET).

autonomous system --A collection of networks under a common administration sharing a common routing strategy. Autonomous systems are subdivided by areas.

Cisco Express Forwarding --An advanced Layer 3 IP switching technology. Cisco Express Forwarding optimizes network performance and scalability for networks that have large and dynamic traffic patterns, such as the Internet, and for networks characterized by intensive Web-based applications or interactive sessions. Cisco Express Forwarding uses a Forwarding Information Base (FIB) to make IP destination prefix-based switching decisions. The FIB is conceptually similar to a routing table or information base. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next-hop address information based on the information in the IP routing table.

headend --The upstream, transmit end of a tunnel. The router that originates and maintains the traffic engineering LSP.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common IGPs include OSPF and Routing Information Protocol (RIP).

interarea TE --Ability for a traffic engineering LSP to span multiple areas.

IS-IS --Intermediate System-to-Intermediate System. IS-IS is an OSI link-state hierarchical routing protocol based on DECnet Phase V routing, where intermediate system (IS) routers exchange routing information based on a single metric to determine the network topology.

label switched path (LSP) tunnel --A configured connection between two routers in which label switching is used to carry the packets.

level-1 routers --Routers that are directly connected to other areas. The routers are not in the backbone. MPLS does not run in the background. These routers are also called internal routers.

level-2 routers --Routers that connect two areas. These routers let you run MPLS in the background.

load balancing --The distribution of traffic among multiple paths to the same destination so that the router uses bandwidth efficiently. Load balancing increases the use of network segments, thus increasing effective network bandwidth.

LSP --label switched path. A sequence of hops such as R0...Rn in which a packet travels from R0 to Rn through label switching mechanisms. A label switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

mask --A bit combination used to describe which part of an address refers to the network or the subnet and which part refers to the host.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets. ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

OSPF --Open Shortest Path First. Link-state, hierarchical IGP routing algorithm proposed as a successor to Routing Information Protocol (RIP) in the Internet community. OSPF features include least-cost routing, multipath routing, and load balancing.

process ID --Distinguishes one process from another within the device. An OSPF process ID can be any positive integer, and it has no significance outside the router on which it is configured.

router ID --Something by which a router originating a packet can be uniquely distinguished from all other routers. For example, an IP address from one of the router’s interfaces.

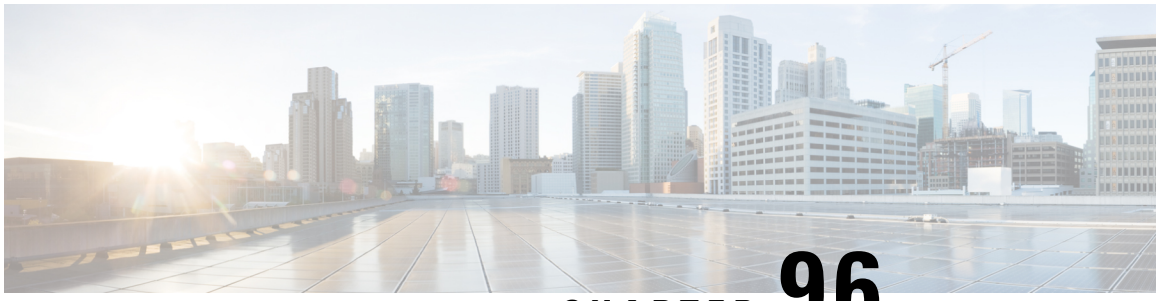
static routing --A static route is a fixed path preprogrammed by a network administrator. Static routes cannot make use of routing protocols and don't self-update after receipt of routing update messages; they must be updated by hand.

tailend --The downstream, receive end of a tunnel. The router that terminates the traffic engineering LSP.

traffic engineering --The techniques and processes that cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

tunnel --A secure communication path between two peers, such as two routers. A traffic engineering tunnel is a label switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.

virtual link --Ordinarily, each area is directly connected to area 0. A virtual link is used for a connection when an area is connected to an area that is one area away from area 0.



CHAPTER 96

MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

The Static IPv6 Routes over MPLS TE IPv4 Tunnels feature helps to statically enable IPv6 tunneling over Multiprotocol Label Switching (MPLS) traffic engineering (TE) IPv4 tunnels on edge devices. This feature provides a simple and cost-effective method to leverage an existing MPLS IPv4 backbone to integrate IPv6 services over service provider core backbones.

Prerequisites for MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

- The MPLS TE feature must be enabled by using the **mpls traffic-eng** command. This command is disabled by default.
- A TE tunnel must be configured.

Restrictions for MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

- Native TE IPv6 tunnels are not supported.
- TE IPv4 tunnel exposure to IPv6 Interior Gateway Protocol (IGP) through IPv6 forwarding adjacency or through autoroute announcement is not supported.
- Static IPv6 routes over TE IPv4 primary autotunnels or autotunnel meshes are not supported.
- Nonstandard Facilities (NSF), stateful switchover (SSO), and Cisco In-Service Software Upgrade (ISSU) high availability requirements are applicable only for dual Route Processor (RP) platforms.
- The TE IPv4 tunnel destination cannot be announced to IPv6 routing.
- TE IPv4 tunnels cannot be announced to IPv6 topologies.
- The tunnel interface needs both IPv4 and IPv6 addresses to forward IPv6 traffic under the tunnel interface. This is because tunnel interface adjacencies are sourced by the adjacency point-to-point manager, which only expects IPv4 to be enabled on the interface before the adjacency point-to-point manager sources the adjacencies.
- If the Static IPv6 Routes over MPLS TE IPv4 Tunnels feature is enabled, TE tunnel statistics will show both MPLS and IPv6 statistics because both IPv6 and MPLS adjacencies are created and used.
- Both the provider-edge-to-customer-edge (PE-to-CE) interface and the CE core-facing interface need IPv6 addresses.
- MPLS and interface statistics on the tunnel egress interface are not supported.
- IPv6 policy-based routing on MPLS TE IPv4 tunnels is not supported.
- Unequal load balancing of IPv6 static routes over multiple TE IPv4 tunnels is not supported.
- TE IPv4 tunnel autobandwidth is not supported.
- IPv6 multicast traffic over TE IPv4 point-to-multipoint tunnel is not supported.
- Generalized MPLS (GMPLS) is not supported.

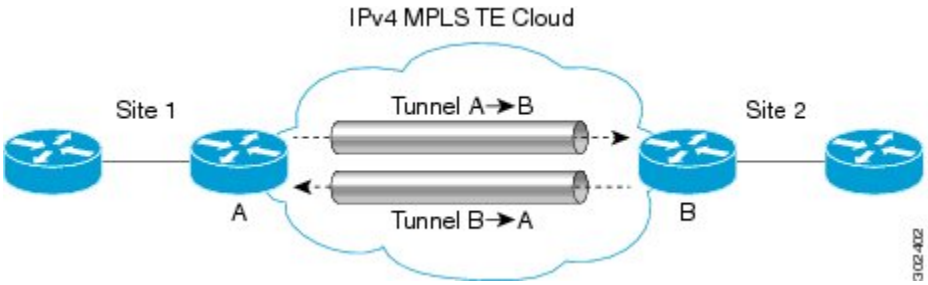
Information About MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

Overview of Static IPv6 Routes over MPLS TE IPv4 Tunnels

The Static IPv6 Routes over MPLS TE IPv4 Tunnels feature manually specifies an MPLS TE IPv4 tunnel as an egress interface for IPv6 routes. Communication is established between remote IPv6 domains by using standard IPv6 tunneling mechanism.

The figure below shows two IPv4-aware and IPv6-aware sites, Site 1 and Site 2, which are connected over an MPLS TE IPv4 core. MPLS TE tunnels are set up across the core between endpoints A and B. IPv6 prefixes from Site 1 are routed onto MPLS TE tunnels through edge device A and vice versa, and IPv6 prefixes from Site 2 are routed onto MPLS TE tunnels through edge device B.

Figure 153: Static IPv6 Route over MPLS TE IPv4 Tunnels



To carry IPv4 and IPv6 traffic on a single MPLS TE IPv4 tunnel, the MPLS Forwarding Infrastructure (MFI) is enhanced at the tunnel ingress and egress endpoints to differentiate between the two types of traffic.

How to Configure MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

Assigning an IPv6 Address to an MPLS TE IPv4 Tunnel

To enable a static IPv6 route over an MPLS TE IPv4 tunnel, first configure a TE IPv4 tunnel, and then assign an IPv6 address or IPv6 unnumbered loopback interface to the TE IPv4 tunnel. The steps for these tasks are listed below.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *interface-number*
4. **ip unnumbered loopback** *interface-number*
5. **ipv6 address** *ipv6-address/prefix-length*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface tunnel** *interface-number* **Example:** `Device(config)# interface tunnel 2`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**ip unnumbered loopback** *interface-number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 5	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:2222:7272::72/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

After assigning an IPv6 address to a TE IPv4 tunnel, configure the IPv6 route by using the IPv4 tunnel as the egress interface.

Configuring a Static IPv6 Route by Specifying an MPLS TE IPv4 Tunnel as the Egress Interface

To route IPv6 traffic over a TE IPv4 tunnel, specify the IPv4 tunnel as the egress interface.

Before you begin

Before configuring an IPv6 route by using a TE IPv4 tunnel as the egress interface, assign an IPv6 address to the TE IPv4 tunnel. For more information, see the "Assigning an IPv6 Address to an MPLS TE IPv4 Tunnel" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-address/prefix-length interface-type interface-number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** *ipv6-address/prefix-length interface-type interface-number* **Example:** `Device(config)# ipv6 route 2001:DB8:2222:7272::72/64 tunnel 2`	Implements static IPv6 routes. **Note** Using the **ipv6 route** command, specify the same tunnel *interface-number* on which the TE IPv4 tunnel is configured using the steps described in the "Assigning an IPv6 Address to an MPLS TE IPv4 Tunnel" section.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying IPv6 Routing over a TE IPv4 Tunnel

The IPv6 routing component is responsible for processing the static IPv6 route configuration and updating the IPv6 Routing Information Base (RIB). You can use the commands listed below in any order to verify the IPv6 routing configuration.

SUMMARY STEPS

1. **enable**
2. **show ipv6 route**
3. **show ipv6 cef** *interface-type interface-number*
4. **exit**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ipv6 route**

Example:

```
Device# show ipv6 route
```

Displays contents of the IPv6 routing table.

Step 3 **show ipv6 cef** *interface-type interface-number*

Example:

```
Device# show ipv6 cef tunnel 2
```

Display entries in the IPv6 Forwarding Information Base (FIB).

Step 4 **exit**

Example:

```
Device# exit
```

Exits privileged EXEC mode.

Displaying IPv6 Statistics over a TE IPv4 Tunnel

When the Static IPv6 Routes over MPLS TE IPv4 Tunnels feature is enabled, the TE IPv4 tunnel can carry both IPv4 and IPv6 traffic. You can display the statistics for IPv6 traffic going over the TE tunnel by using the commands described in this task. These commands can be used in any order. The statistics are displayed on a per-interface, per-protocol basis.

Note MPLS and interface statistics will be counted twice due to the presence of two midchain adjacencies in the tunnel. You can subtract IPv6 link adjacency statistics (obtained from the **show adjacency link ipv6** command) from the interface IPv6 statistics (obtained from the **show interface accounting** command) to arrive at accurate statistics.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [*ipv6-address*/*prefix-length*]
3. **show interfaces accounting**
4. **show interface** [*interface-type interface-number*] **stats**
5. **show adjacency**
6. **exit**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show mpls forwarding-table** [*ipv6-address*/*prefix-length*]

Example:

```
Device# show mpls forwarding-table
```

Displays the contents of MPLS Label FIB (LFIB).

Step 3 **show interfaces accounting**

Example:

```
Device# show interfaces accounting
```

Displays the number of packets of each protocol type that have been sent through all configured interfaces.

Step 4 **show interface** [*interface-type interface-number*] **stats**

Example:

```
Device# show interface stats
```

Displays numbers of packets that were process switched, fast switched, and distributed switched.

Step 5 **show adjacency**

Example:

```
Device# show adjacency
```

Displays information about the Cisco Express Forwarding adjacency table or the hardware Layer 3-switching adjacency table.

Step 6 **exit**

Example:

```
Device# exit
```

Exits privileged EXEC mode.

Troubleshooting IPv6 Routing over a TE IPv4 Tunnel

You can use the following commands for troubleshooting:

- **debug ipv6 cef**—Dsplays debug messages for Cisco Express Forwarding for IPv6.
- **debug ipv6 routing**—Displays debug messages for IPv6 routing table updates and route cache updates.
- **debug mpls traffic-eng**—Displays debug messages for MPLS traffic engineering activities.

Configuration Examples for MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

Example: Assigning an IPv6 Address to an MPLS TE IPv4 Tunnel

```
Device> enable
Device# configure terminal
Device(config)# interface tunnel 1
Device(config-if)# ip unnumbered loopback 0
Device(config-if)# ipv6 address 2001:DB8::/32
Device(config-if)# end
```

Example: Configuring a Static IPv6 Route by Specifying an MPLS TE IPv4 Tunnel as an Egress Interface

```
Device> enable
Device# configure terminal
Device(config)# ipv6 route 2001:DB8::/32 tunnel 1
Device(config)# end
```

Additional References for MPLS TE - Bundled Interface Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS traffic engineering commands	Cisco IOS Multiprotocol Label Switching Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

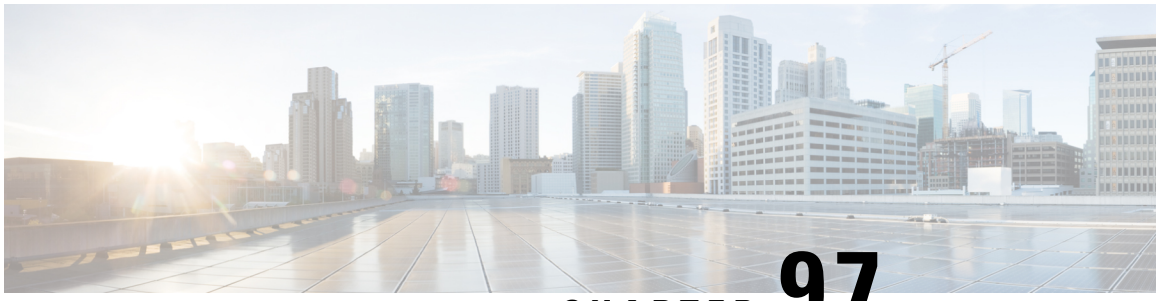
Feature Information for MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 173: Feature Information for MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels

Feature Name	Releases	Feature Information
MPLS Traffic Engineering Static IPv6 Routes over MPLS TE IPv4 Tunnels	15.2(4)S	The Static IPv6 Routes over MPLS TE IPv4 Tunnels feature helps to statically enable IPv6 tunneling over Multiprotocol Label Switching (MPLS) traffic engineering (TE) IPv4 tunnels through edge devices. This feature provides a simple and cost-effective method to leverage an existing MPLS IPv4 backbone to integrate IPv6 services over service provider core backbones.



CHAPTER 97

MPLS Traffic Engineering Automatic Bandwidth Adjustment for TE Tunnels

The MPLS Traffic Engineering (TE) Automatic Bandwidth Adjustment for TE Tunnels feature provides the means to automatically adjust the bandwidth allocation for traffic engineering tunnels based on their measured traffic load. The configured bandwidth in the running configuration is changed due to the automatic bandwidth behavior.

Prerequisites for MPLS TE Automatic Bandwidth Adjustment for TE Tunnels

Your network must support the following:

- Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels
- Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)

MPLS TE must be configured on the interface and on the tunnels.

Restrictions for MPLS TE Automatic Bandwidth Adjustment for TE Tunnels

- The automatic bandwidth adjustment feature treats each tunnel for which it has been enabled independently. That is, it adjusts the bandwidth for each such tunnel according to the adjustment frequency configured

for the tunnel and the sampled output rate for the tunnel since the last adjustment without regard for any adjustments previously made or pending for other tunnels.

- If a tunnel is brought down to calculate a new label switched path (LSP) because the LSP is not operational, the configured bandwidth is not saved. If the router is reloaded, the last saved automatic bandwidth value is used.
- You cannot configure MPLS TE over the logical generic routing encapsulation (GRE) tunnel interface.
- MPLS traffic engineering supports only a single IGP process/instance. Multiple IGP processes/instances are not supported and MPLS traffic engineering should not be configured in more than one IGP process/instance.

Information About MPLS TE Automatic Bandwidth Adjustment for TE Tunnels

MPLS TE Automatic Bandwidth Adjustment for TE Tunnels Overview

Traffic engineering autobandwidth samples the average output rate for each tunnel marked for automatic bandwidth adjustment. For each marked tunnel, the feature periodically (for example, once per day) adjusts the tunnel's allocated bandwidth to be the largest sample for the tunnel since the last adjustment.

The frequency with which tunnel bandwidth is adjusted and the allowable range of adjustments is configurable on a per-tunnel basis. In addition, the sampling interval and the interval over which to average tunnel traffic to obtain the average output rate is user-configurable on a per-tunnel basis.

MPLS TE Automatic Bandwidth Adjustment for TE Tunnels Benefits

The automatic bandwidth feature allows you to configure and monitor the bandwidth for MPLS TE tunnels. If automatic bandwidth is configured for a tunnel, TE automatically adjusts the tunnel's bandwidth.

How to Configure MPLS TE Automatic Bandwidth Adjustment for TE Tunnels

Configuring a Device to Support Traffic Engineering Tunnels

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **mpls traffic-eng tunnels**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Router(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding operation.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config)# mpls traffic-eng tunnels`	Enables the MPLS traffic engineering tunnel feature on a device.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring IS-IS or OSPF for MPLS Traffic Engineering

Perform one of the follow tasks to configure IS-IS or OSPF for MPLS TE:

Configuring IS-IS for MPLS Traffic Engineering

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis**
4. **mpls traffic-eng level-1**
5. **mpls traffic-eng router-id loopback0**
6. **metric-style wide**
7. **exit**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** **Example:** `Router(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process for IP, and enters router configuration mode.
Step 4	**mpls traffic-eng level-1** **Example:** `Router(config-router)# mpls traffic-eng level-1`	Turns on MPLS TE for IS-IS level 1.
Step 5	**mpls traffic-eng router-id loopback0** **Example:** `Router(config-router)# mpls traffic-eng router-id loopback0`	Specifies that the TE router identifier for the node is the IP address associated with interface loopback0.
Step 6	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router to generate and accept only new-style type, length, value objects (TLVs).
Step 7	**exit** **Example:** `Router(config-router)# exit`	Exits to global configuration mode.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring OSPF for MPLS Traffic Engineering

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router ospf** *process-id*
4. **mpls traffic-eng area** *number*
5. **mpls traffic-eng router-id loopback0**
6. **exit**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 200`	Configures an OSPF routing process for IP and enters router configuration mode. • The value for the *process-id* argument is an internally used identification parameter for an OSPF routing process. It is locally assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**mpls traffic-eng area** *number* **Example:** `Router(config-router)# mpls traffic-eng area 0`	Turns on MPLS TE for the indicated OSPF area.
Step 5	**mpls traffic-eng router-id loopback0** **Example:** `Router(config-router)# mpls traffic-eng router-id loopback0`	Specifies that the TE router identifier for the node is the IP address associated with interface loopback0.
Step 6	**exit** **Example:** `Router(config-router)# exit`	Exits to global configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring Bandwidth on Each Link That a Tunnel Crosses

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **mpls traffic-eng tunnels**
5. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** *kbps*]
6. **exit**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config-if)# mpls traffic-eng tunnels`	Enables MPLS TE tunnels on an interface.
Step 5	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** *kbps*] **Example:** `Router(config-if)# ip rsvp bandwidth 1000 100`	Enables Resource Reservation Protocol (RSVP) for IP on an interface. • The *interface-kbps* argument specifies the maximum amount of bandwidth (in kbps) that may be allocated by RSVP flows. The range is from 1 to 10000000. • The *single-flow-kbps*argument is the maximum amount of bandwidth, in kbps, that may be allocated to a single flow. The range is from 1 to 10000000.
Step 6	**exit** **Example:**	Exits to global configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	
Step 7	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel

To configure an MPLS TE tunnel, perform the following task. The MPLS TE tunnel has two path setup options: a preferred explicit path and a backup dynamic path.

Note The configuration applies only to the TE head-end node. The configuration applies to all nodes and interfaces in the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *interface-type interface-number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** [**protect**] *preference-number*{**dynamic** | **explicit** | {**name** *path-name* | *path-number*}} [**lockdown**]
9. **exit**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**ip unnumbered** *interface-type interface-number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Gives the tunnel interface an IP address that is the same as that of interface Loopback0. • An MPLS TE tunnel interface should be unnumbered because it represents a unidirectional link. **Note** This command is not effective until Lookback0 has been configured with an IP address.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 10.3.3.3`	Specifies the destination for a tunnel. • The destination must be the MPLS TE router ID of the destination device.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the encapsulation mode of the tunnel to MPLS TE.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 250`	Configures the bandwidth for the MPLS TE tunnel. • The *bandwidth* argument is the bandwidth, in kilobits per second, set for the MPLS TE tunnel. The range is from 1 to 4294967295. The default is 0. • If automatic bandwidth is configured for the tunnel, the **tunnel mpls traffic-eng bandwidth** command configures the initial tunnel bandwidth, which will be adjusted by the autobandwidth mechanism. **Note** If you configure a tunnel's bandwidth with the **tunnel mpls traffic-eng bandwidth** command and the minimum amount of automatic bandwidth with the **tunnel mpls traffic-eng auto-bw**command, the minimum amount of automatic bandwidth adjustment is the lower of those two configured values.
Step 8	**tunnel mpls traffic-eng path-option** [**protect**] *preference-number*{**dynamic** | **explicit** | {**name** *path-name* | *path-number*}} [**lockdown**] **Example:**	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the TE topology database. • A dynamic path is used if an explicit path is currently unavailable.

	Command or Action	Purpose
	`Router(config-if)# tunnel mpls traffic-eng path-option 10 explicit avoid-protected-link`	
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Troubleshooting Tips

Each **tunnel mpls traffic-eng auto-bw** command supersedes the previous one. Therefore, if you want to specify multiple options for a tunnel, you must specify them all in a single **tunnel mpls traffic-eng auto-bw** command.

Enabling Automatic Bandwidth Adjustment on a Platform

To enable automatic bandwidth adjustment on a platform and initiate sampling the output rate for tunnels configured for bandwidth adjustment, perform the following task.

Note This task is applicable only to the TE head-end router. The configuration applies to all locally-configured TE head-end interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-bw timers** [**frequency** *seconds*]
4. **no mpls traffic-eng auto-bw timers**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng auto-bw timers** [**frequency** *seconds*] **Example:** `Router(config)# mpls traffic-eng auto-bw timers frequency 300`	Enables automatic bandwidth adjustment on a platform and begins sampling the output rate for tunnels that have been configured for automatic bandwidth adjustment. • The **frequency** keyword specifies the interval, in seconds, for sampling the output rate of each tunnel configured for automatic bandwidth. The range is 1 through 604800. The recommended value is 300.

Step 4	**no mpls traffic-eng auto-bw timers**
	Example:
	`Router(config)# no mpls traffic-eng auto-bw timers`
		(Optional) Disables automatic bandwidth adjustment on a platform.
		• Use the **no** version of the command, whichterminates output rate sampling and bandwidth adjustment for tunnels. In addition, the **no** form of the command restores the configured bandwidth for each tunnel where the configured bandwidth is determined as follows:
		  • If the tunnel bandwidth was explicitly configured via the **tunnel mpls traffic-eng bandwidth** command after the running configuration was written to the startup configuration, the configured bandwidth is the bandwidth specified by that command.
		  • Otherwise, the configured bandwidth is the bandwidth specified for the tunnel in the startup configuration.

Step 5	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Enabling Automatic Bandwidth Adjustment for a Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng auto-bw [collect-bw] [frequency** *seconds*] [**adjustment-threshold** *percent*] [**overflow-limit** *number* **overflow-threshold** *percent*] [**max-bw** *kbps*] [**min-bw** *kbps*]

5. **exit**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng auto-bw [collect-bw] [frequency** *seconds*] [**adjustment-threshold** *percent*] [**overflow-limit** *number* **overflow-threshold** *percent*] [**max-bw** *kbps*] [**min-bw** *kbps*] **Example:** `Router(config-if)# tunnel mpls traffic-eng auto-bw max-bw 2000 min-bw 1000`	Enables automatic bandwidth adjustment for the tunnel and controls the manner in which the bandwidth for a tunnel is adjusted.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuring the Interval for Computing the Tunnel Average Output Rate

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*

4. **load-interval** *seconds*
5. **exit**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**load-interval** *seconds* **Example:** `Router(config-if)# load-interval 90`	Configures the interval over which the input and output rates for the interface are averaged. • The *seconds* argument is the length of time for which data is used to compute load statistics. The value is a multiple of 30, from 30 to 600 (30, 60, 90, 120, and so on). The default is 300.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Verifying Automatic Bandwidth Configuration

SUMMARY STEPS

1. **show mpls traffic-eng tunnels**
2. **show running-config**

DETAILED STEPS

Step 1 show mpls traffic-eng tunnels

Use this command to display information about tunnels, including automatic bandwidth information for tunnels that have the feature enabled. For example:

Example:

```
Router# show mpls traffic-eng tunnels
Name:tagsw4500-9_t1 (Tunnel1) Destination:10.0.0.4
Status:
Admin:up Oper:up Path:valid Signalling:connected
path option 1, type explicit pbr_south (Basis for Setup, path weight 30)
path option 2, type dynamic
Config Parameters:
Bandwidth:13 kbps (Global) Priority:7 7 Affinity:0x0/0xFFFF
AutoRoute: disabled LockDown:disabled Loadshare:13 bw-based
auto-bw:(300/265) 53 Bandwidth Requested: 13
   Adjustment threshold: 5%
   Overflow Limit: 4  Overflow Threshold: 25%
  Overflow Threshold Crossed: 1
  Sample Missed: 1 Samples Collected: 1
Active Path Option Parameters:
  State: dynamic path option 1 is active
  BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
 InLabel  :  -
 OutLabel : Serial3/0, 18
 RSVP Signalling Info:
   Src 10.0.0.1, Dst 10.0.0.4, Tun_Id 2, Tun_Instance 2
  RSVP Path Info:
   My Address: 10.105.0.1
   Explicit Route: 10.105.0.2 104.105.0.1 10.0.0.4
   Record   Route:   NONE
   Tspec: ave rate=13 kbits, burst=1000 bytes, peak rate=13 kbits
 Record   Route:   NONE
   Tspec: ave rate=13 kbits, burst=1000 bytes, peak rate=13 kbits
  RSVP Resv Info:
   Record   Route:   NONE
   Fspec: ave rate=13 kbits, burst=1000 bytes, peak rate=13 kbits
  Shortest Unconstrained Path Info:
   Path Weight: 128 (TE)
   Explicit Route: 10.105.0.2 104.105.0.1 10.0.0.4
  History:
   Tunnel:
    Time since created: 7 minutes, 56 seconds
     Time since path change: 7 minutes, 18 seconds
     Number of LSP IDs (Tun_Instances) used: 2
     Number of Auto-bw Adjustment resize requests: 1
     Time since last Auto-bw Adjustment resize request: 1 minutes, 7 seconds
     Number of Auto-bw Overflow resize requests: 1
     Time since last Auto-bw Overflow resize request: 52 seconds
     Current LSP:
      Uptime: 52 seconds
      Selection: reoptimization
    Prior LSP:
ID: path option 1 [1]
  Removal Trigger: configuration changed
```

In the command output:

- The auto-bw line indicates that automatic bandwidth adjustment is enabled for the tunnel.

- 300 is the time, in seconds, between bandwidth adjustments.
- 265 is the time, in seconds, remaining until the next bandwidth adjustment.
- 53 is the largest bandwidth sample since the last bandwidth adjustment.
- 13 is the last bandwidth adjustment and the bandwidth currently requested for the tunnel.
- The adjustment threshold is 5 percent.
- The overflow limit is 4.
- The overflow threshold is 25 percent.
- The overflow crossed is 1.

Example:

Step 2 **show running-config**

Use this command to verify that the **tunnel mpls traffic-eng auto bw** command is as you expected. For example:

Example:

```
Router# show running-config
.
.
.
interface tunnel1
  ip unnumbered loopback 0
  tunnel destination 192.168.17.17 255.255.255.0
  tunnel mode mpls traffic-eng
  tunnel mpls traffic-eng bandwidth 1500
  tunnel mpls traffic-eng priority 1 1
  tunnel mpls traffic-eng path-option 1 dynamic
```

tunnel mpls traffic-eng auto bw max-bw 2000 min-bw 1000 !Enable automatic bandwidth

Example:

```
.
.
.
```

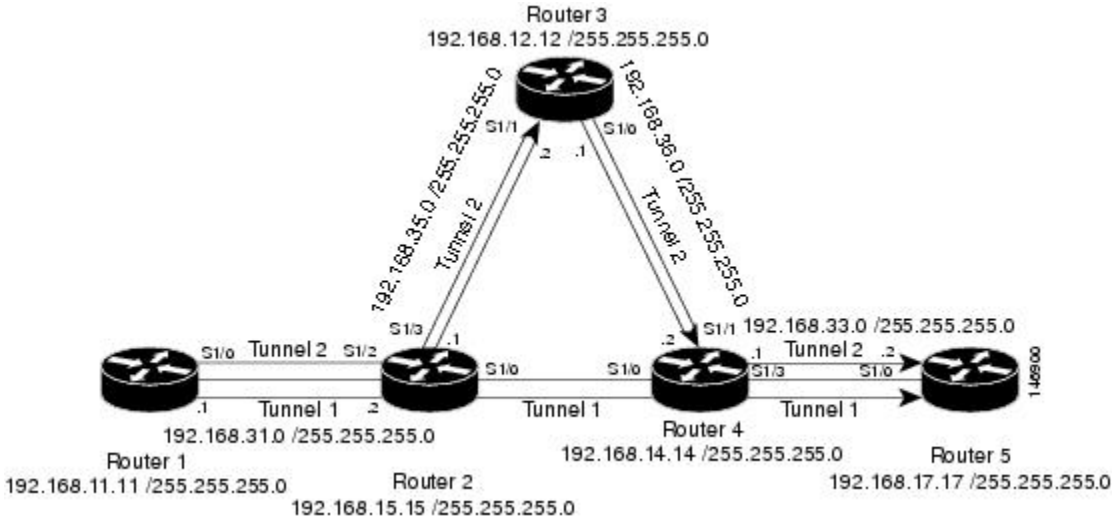
The sample output from the **show running-config** command shows that the value 1500, in the **tunnel mpls traffic-eng bandwidth 1500** command, changes after an adjustment is made.

Example:

Configuration Examples for MPLS TE Automatic Bandwidth Adjustments for TE Tunnels

The figure below illustrates a sample MPLS topology. The following sections contain sample configuration examples to configure automatic bandwidth adjustment for MPLS TE tunnels originating on Router 1 and to enable automatic bandwidth adjustment for Tunnel 1.

Figure 154: Sample MPLS Traffic Engineering Tunnel Configuration



The examples omit some configuration required for MPLS TE, such as the required RSVP and Interior Gateway Protocol (IGP) (IS-IS or OSPF) configuration, because the purpose of these examples is to illustrate the configuration for automatic bandwidth adjustment.

Example: Configuring MPLS Traffic Engineering Automatic Bandwidth

The following example shows how to use the **mpls traffic-eng auto-bw timers** command to enable automatic bandwidth adjustment for Router 1. The command specifies that the output rate is to be sampled every 10 minutes for tunnels configured for automatic bandwidth adjustment.

```
configure terminal
!
ip cef distributed
mpls traffic-eng tunnels
mpls traffic-eng auto-bw timers frequency 600  !Enable automatic bandwidth adjustment
interface loopback 0
ip address 192.168.11.11 255.255.255.0
```

Example: Tunnel Configuration for Automatic Bandwidth

The following example shows how to use the **tunnel mpls traffic-eng auto-bw** command to enable automatic bandwidth adjustment for Tunnel 1. The command specifies a maximum allowable bandwidth of 2000 kbps,

a minimum allowable bandwidth of 1000 kbps, and that the default automatic bandwidth adjustment frequency of once a day be used.

```
interface tunnel1
  ip unnumbered loopback 0
  tunnel destination 192.168.17.17
  tunnel mode mpls traffic-eng
  tunnel mpls traffic-eng bandwidth 1500
  tunnel mpls traffic-eng priority 1 1
  tunnel mpls traffic-eng path-option 1 dynamic
  tunnel mpls traffic-eng auto-bw max-bw 2000 min-bw 1000   !Enable automatic bandwidth
                                                            !adjustment for Tunnel1
```

Additional References

Related Documents

Related Topic	Document Title
IS-IS and OSPF commands	*Cisco IOS IP Routing Protocols Command Reference*
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Quality of service solutions commands	*Cisco IOS Quality of Service Solutions Command Reference*
Quality of service solutions configuration	Quality of Service Overview

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
MPLS Traffic Engineering MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

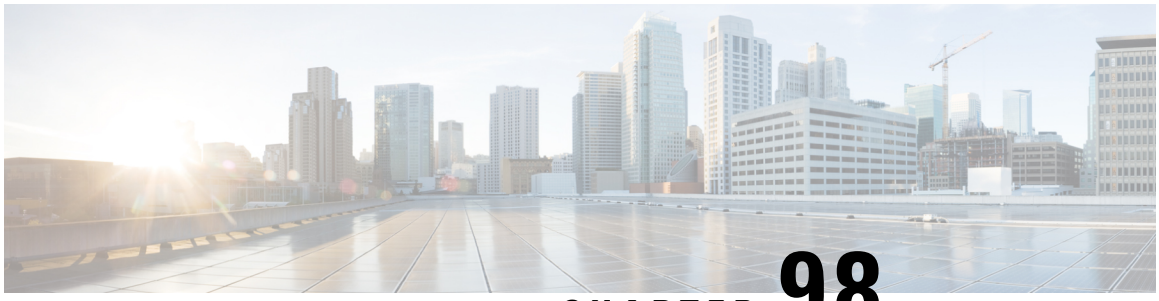
Feature Information for MPLS TE Automatic Bandwidth Adjustment for TE Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 174: Feature Information for MPLS TE Automatic Bandwidth Adjustment for TE Tunnels

Feature Name	Releases	Feature Information
MPLS Traffic Engineering Automatic Bandwidth Adjustment for TE Tunnels	12.2(33)SRE	The MPLS Traffic Engineering Automatic Bandwidth Adjustment for TE Tunnels feature provides the means to automatically adjust the bandwidth allocation for traffic engineering tunnels based on their measured traffic load. The configured bandwidth in the running configuration is changed due to the automatic bandwidth behavior. In Cisco IOS Release 12.2(33)SRE, this feature was introduced. The following commands were introduced or modified to support automatic bandwidth adjustment threshold and overflow threshold: **mpls traffic-eng lsp attributes**, **show mpls traffic-eng tunnels**, **tunnel mpls traffic-eng auto-bw**.



CHAPTER 98

MPLS Traffic Engineering – Bundled Interface Support

The MPLS Traffic Engineering - Bundled Interface Support feature enables Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels over the bundled interfaces—EtherChannel and Gigabit EtherChannel (GEC).

The Resource Reservation Protocol (RSVP) notifies TE about bandwidth changes that occur when member links are added or deleted, or when links become active or inactive. TE notifies other nodes in the network via Interior Gateway Protocol (IGP) flooding. By default, the bandwidth available to TE Label-Switched Paths (LSPs) is 75 percent of the interface bandwidth. You can change the percentage of the global bandwidth available for TE LSPs by using an RSVP command on the bundled interface. Bandwidth reservation and preemption are supported.

The Fast Reroute (FRR) feature is supported on bundled interfaces. FRR is activated when a bundled interface goes down; for example, if you enter the **shutdown** command to shut down the interface or fewer than the required minimum number of links are operational.

Prerequisites for MPLS TE – Bundled Interface Support

- Configure Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels.
- Enable Cisco Express Forwarding in global configuration mode.
- Enable Resource Reservation Protocol (RSVP) feature.
- Configure EtherChannel.
- Configure Gigabit EtherChannel.

Restrictions for MPLS TE – Bundled Interface Support

- Traffic engineering over switch virtual interfaces (SVIs) is not supported unless the SVI consists of a bundle of links that represent a single point-to-point interface.
- There must be a valid IP address configuration on the bundled interface and there must not be an IP address configuration on the member links.

Information About MPLS TE – Bundled Interface Support

Cisco EtherChannel Overview

Cisco EtherChannel technology builds upon standards-based 802.3 full-duplex Fast Ethernet to provide network managers with a reliable, high-speed solution for the campus network backbone. EtherChannel technology provides bandwidth scalability within the campus by providing up to 800 Mbps, 8 Gbps, or 80 Gbps of aggregate bandwidth for a Fast EtherChannel, Gigabit EtherChannel, or 10 Gigabit EtherChannel connection, respectively. Each of these connection speeds can vary in amounts equal to the speed of the links used (100 Mbps, 1 Gbps, or 10 Gbps). Even in the most bandwidth-demanding situations, EtherChannel technology helps to aggregate traffic, keeps oversubscription to a minimum, and provides effective link-resiliency mechanisms.

Cisco EtherChannel Benefits

Cisco EtherChannel technology allows network managers to provide higher bandwidth among servers, routers, and switches than a single-link Ethernet technology can provide.

Cisco EtherChannel technology provides incremental scalable bandwidth and the following benefits:

- Standards-based—Cisco EtherChannel technology builds upon IEEE 802.3-compliant Ethernet by grouping multiple, full-duplex point-to-point links. EtherChannel technology uses IEEE 802.3 mechanisms for full-duplex autonegotiation and autosensing, when applicable.
- Flexible incremental bandwidth—Cisco EtherChannel technology provides bandwidth aggregation in multiples of 100 Mbps, 1 Gbps, or 10 Gbps, depending on the speed of the aggregated links. For example, network managers can deploy EtherChannel technology that consists of pairs of full-duplex Fast Ethernet links to provide more than 400 Mbps between the wiring closet and the data center. In the data center, bandwidths of up to 800 Mbps can be provided between servers and the network backbone to provide large amounts of scalable incremental bandwidth.
- Load balancing—Cisco EtherChannel technology comprises several Fast Ethernet links and is capable of load balancing traffic across those links. Unicast, broadcast, and multicast traffic is evenly distributed across the links, providing improved performance and redundant parallel paths. When a link fails, traffic is redirected to the remaining links within the channel without user intervention and with minimal packet loss.
- Resiliency and fast convergence—When a link fails, Cisco EtherChannel technology provides automatic recovery by redistributing the load across the remaining links. When a link fails, Cisco EtherChannel technology redirects traffic from the failed link to the remaining links in less than one second. This convergence is transparent to the end user—no host protocol timers expire and no sessions are dropped.

Cisco Gigabit EtherChannel Overview

Cisco Gigabit EtherChannel (GEC) is a high-performance Ethernet technology that provides transmission rates in Gigabit per second (Gbps). A Gigabit EtherChannel bundles individual ethernet links (Gigabit Ethernet and 10 Gigabit Ethernet) into a single logical link that provides the aggregate bandwidth up to four physical links. All LAN ports in each EtherChannel must be of the same speed and must be configured as either Layer 2 or Layer 3 LAN ports. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link in the EtherChannel.

Load Balancing and Min-Links in EtherChannel

Load balancing affects the actual and practical bandwidth that can be used for TE. Multilink load balancing uses a per-packet load balancing method. All of the bundle interface bandwidth is available. EtherChannel load balancing has various load balancing methods, depending on the traffic pattern and the load balancing configuration. The total bandwidth available for TE may be limited to the bandwidth of a single member link.

On EtherChannel, min-links is supported only in the Link Aggregation Control Protocol (LACP). For other EtherChannel protocols, the minimum is one link, by default, and it is not configurable. To configure min-links for EtherChannel, use the **port-channel min-links** command.

How to Configure MPLS TE – Bundled Interface Support

Configuring MPLS TE on an EtherChannel Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip address** *ip-address mask* [**secondary**]
5. **mpls traffic-eng tunnels**
6. **mpls traffic-eng backup-path** *tunnel*
7. **port-channel min-links** *min-num*
8. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface port-channel 1`	Creates an EtherChannel bundle, assigns a group number to the bundle, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.4 255.255.255.0`	Specifies an IP address for the EtherChannel group.
Step 5	**mpls traffic-eng tunnels** **Example:** `Device(config-if)# mpls traffic-eng tunnels`	Enables MPLS TE tunnel signaling on an interface. • MPLS TE tunnel should be enabled on the device before enabling the signaling.
Step 6	**mpls traffic-eng backup-path** *tunnel* **Example:** `Device(config-if)# mpls traffic-eng backup-path Tunnel120`	(Optional) Configures the physical interface to use a backup tunnel in the event of a detected failure on that interface.
Step 7	**port-channel min-links** *min-num* **Example:** `Device(config-if)# port-channel min-links 2`	Specifies that a minimum number of bundled ports in an EtherChannel is required before the channel can be active.
Step 8	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth 100`	Enables RSVP for IP on an interface and specifies a percentage of the total interface bandwidth as available in the RSVP bandwidth pool.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for MPLS TE Bundled Interface Support

Example: Configuring MPLS TE on an EtherChannel Interface

```
Device> enable
```

```
Device# configure terminal
Device(config)# interface port-channel 1
Device(config-if)# ip address 10.0.0.4 255.255.255.0
Device(config-if)# mpls traffic-eng tunnels
Device(config-if)# mpls traffic-eng backup-path Tunnel 120
Device(config-if)# port-channel min-links 2
Device(config-if)# ip rsvp bandwidth 100
Device(config-if)# end
```

Example: Configuring MPLS TE - Bundled Interface Support over Gigabit Etherchannel

The following example shows how to enable MPLS TE – bundled interface support over GEC on Cisco devices:

```
Device> enable
Device# configure terminal

! Enable global MPLS TE on routers
Device(config)# router ospf 100
Device(config-router)# network 10.0.0.1 0.0.0.255 area 0
Device(config-router)# mpls traffic-eng area 0
Device(config-router)# mpls traffic-eng router-id Loopback 0
Device(config-router)# exit

! Configure  GEC  interface and enable MPLS TE and RSVP on interface
Device(config)# interface Port-channel 1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# mpls traffic-eng tunnels
Device(config-if)# ip rsvp bandwidth
Device(config-if)# exit

! Define explicit path
Device(config)# ip explicit-path name primary enable
Device(cfg-ip-expl-path)# next-address 172.12.1.2
Device(cfg-ip-expl-path)# next-address 172.23.1.2
Device(cfg-ip-expl-path)# next-address 172.34.1.2
Device(cfg-ip-expl-path)# next-address 10.4.4.4
Device(cfg-ip-expl-path)# exit

! Configure primary tunnel on head-end device
Device(config)# interface Tunnel 14
Device(config-if)# ip unnumbered Loopback 0
Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel destination 10.10.10.0
Device(config-if)# tunnel mpls traffic-eng autoroute announce
Device(config-if)# tunnel mpls traffic-eng path-option 10 explicit name primary
Device(config-if)# tunnel mpls traffic-eng fast-reroute
Device(config-if)# exit

! Configure backup tunnel on head-end or mid-point device
Device(config)# interface Tunnel 23
Device(config-if)# ip unnumbered Looback 0
Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel destination 10.20.10.0
Device(config-if)# tunnel mpls traffic-eng path-option 10 explicit name backup
Device(config-if)# exit
```

```
! Configure backup tunnel on protected GEC interface
Device(config)# interface Port-channel 1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# mpls traffic-eng tunnels
Device(config-if)# mpls traffic-eng backup-path Tunnel 23
Device(config-if)# ip rsvp bandwidth percent 20
Device(config-if)# lacp min-bundle 2
Device(config-if)# exit

! Configure GEC interface
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# no ip address
Device(config-if)# channel-group 1 mode active
Device(config-if)# exit

! Configure GEC interface
Device(config)# interface GigabitEthernet 0/0/2
Device(config-if)# no ip address
Device(config-if)# channel-group 1 mode active
Device(config-if)# exit
```

The **show mpls traffic-eng tunnels** command output displays information about a tunnel or one–line information about all tunnels configured on the device:

```
Device# show mpls traffic-eng tunnels tunnel 14

Name: ASR1013_t14                         (Tunnel10) Destination: 10.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit toR4overR3R3 (Basis for Setup, path weight 3)

  Config Parameters:
    Bandwidth: 0        kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute: enabled  LockDown: disabled Loadshare: 0 [0] bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled


      InLabel  :  -
  OutLabel : Port-channel1, 1608
  Next Hop : 172.16.1.2
  FRR OutLabel : Tunnel23, 4868
  RSVP Signalling Info:
       Src 10.1.1.1, Dst 10.4.4.4, Tun_Id 14, Tun_Instance 35
    RSVP Path Info:
      My Address: 172.12.1.1
      Explicit Route: 172.12.1.2 172.23.1.1 172.23.1.2 172.34.1.1
                      172.34.1.2 10.4.4.4

  History:
    Tunnel:
      Time since created: 17 hours
      Time since path change: 18 minutes, 22 seconds
      Number of LSP IDs (Tun_Instances) used: 35
    Current LSP: [ID: 35]
      Uptime: 18 minutes, 22 seconds
      Selection: reoptimization
    Prior LSP: [ID: 32]
```

```
      ID: path option unknown
      Removal Trigger: signalling shutdown
```

```
Device# show mpls traffic-eng tunnels brief

show mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:           running
    RSVP Process:                   running
    Forwarding:                     enabled
    Periodic reoptimization:        every 3600 seconds, next in 3299 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 299 seconds

P2P TUNNELS/LSPs:
TUNNEL NAME                     DESTINATION      UP IF     DOWN IF   STATE/PROT^M
ASR1013_t14                     10.4.1.1                      -         Po12      up/up
On Mid Router:
P2P TUNNELS/LSPs:
TUNNEL NAME                     DESTINATION      UP IF     DOWN IF   STATE/PROT
ASR1013_t14                      10.4.1.1                   Po12       Po23       up/up
ASR1002F_t23                     10.2.1.1                   Po25        -         up/up
```

The **show mpls traffic-eng fast-reroute** command output displays information about FRR-protected MPLS TE tunnels originating, transmitting, or terminating on this device.

```
Device# show mpls traffic-eng fast-reroute database

P2P Headend FRR information:
Protected tunnel               In-label Out intf/label   FRR intf/label   Status
---------------------------    -------- --------------   --------------   ------

P2P LSP midpoint frr information:
LSP identifier                 In-label Out intf/label   FRR intf/label   Status
---------------------------    -------- --------------   --------------   ------
10.1.1.1 1 [2]                 16        Po23:16          Tu23:16          active
```

Additional References for MPLS TE - Bundled Interface Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS traffic engineering commands	Cisco IOS Multiprotocol Label Switching Command Reference
IPv6 commands	Cisco IOS IPv6 Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS TE - Bundled Interface Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 175: Feature Information for MPLS TE - Bundled Interface Support

Feature Name	Releases	Feature Information
MPLS TE - Bundled Interface Support		The MPLS TE - Bundled Interface Support feature enables MPLS traffic engineering (TE) tunnels over the bundled interfaces EtherChannel and Gigabit EtherChannel (GEC).

Glossary

bundled interface—Generic terms to represent port-channel, multilink, and VLAN interfaces.

Cisco express forwarding —A means for accelerating the forwarding of packets within a router, by storing route lookup information in several data structures instead of in a route cache.

CLNS —Connectionless Network Service. The Open Systems Interconnection (OSI) network layer service that does not require a circuit to be established before data is transmitted. CLNS routes messages to their destination independently of any other messages.

CSPF —Constrained Shortest Path First. A routing protocol that calculates the shortest path based on a set of constraints, such as a minimum bandwidth requirement, maximum number of nodes, or nodes to include or exclude.

enterprise network —A large and diverse network connecting most major points in a company or other organization.

FRR—Fast ReRoute.

headend —The endpoint of a broadband network. All stations send toward the headend; the headend then sends toward the destination stations.

IGP —Interior Gateway Protocol. An Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

interface —A network connection.

IS-IS —Intermediate System to Intermediate System. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, where ISs (routers) exchange routing information based on a single metric, to determine the network topology.

LDN— Link Down Notification.

LSP —Label-Switched Path. A sequence of hops (R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A label-switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

member links—Individual interfaces that are grouped into a bundled interface.

message-pacing —The former name of the rate limiting feature.

MPLS —Formerly known as tag switching, Multiprotocol Label Switching is a method for directing packets primarily through Layer 2 switching rather than Layer 3 routing. In MPLS, packets are assigned short fixed-length labels at the ingress to an MPLS cloud by using the concept of forwarding equivalence classes. Within the MPLS domain, the labels are used to make forwarding decisions mostly without recourse to the original packet headers.

OSPF —Open Shortest Path First. A link-state, hierarchical Interior Gateway Protocol (IGP) routing protocol. derived from the Intermediate System-Intermediate System (IS-IS) protocol. OSPF features are least-cost routing, multipath routing, and load balancing.

router —A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RSVP —Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network.

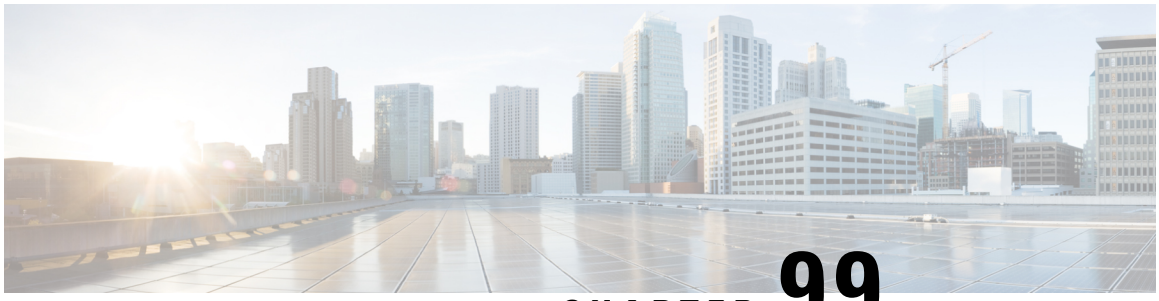
scalability —An indicator showing how quickly some measure of resource usage increases as a network gets larger.

TLV —type, length, value. TLV objects are used in data communication to provide optional information. The type field indicates the type of items in the value field. The length field indicates the length of the value field. The value field is the data portion of the packet.

topology —The physical arrangement of network nodes and media within an enterprise networking structure.

TE (traffic engineering) —Techniques and processes that cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

traffic engineering tunnel —A label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing would cause the tunnel to take.



CHAPTER 99

RSVP Refresh Reduction and Reliable Messaging

The RSVP Refresh Reduction and Reliable Messaging feature includes refresh reduction, which improves the scalability, latency, and reliability of Resource Reservation Protocol (RSVP) signaling to enhance network performance and message delivery.

History for the RSVP Refresh Reduction and Reliable Messaging Feature

Release	Modification
12.2(13)T	This feature was introduced.
12.0(24)S	This feature was integrated into Cisco IOS Release 12.0(24)S.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
12.0(26)S	Two commands, **ip rsvp signalling refresh misses**and **ip rsvp signalling refresh interval**, were added into Cisco IOS Release 12.0(26)S.
12.0(29)S	The *burst* and *max-size* argument defaults for the **ip rsvp signalling rate-limit** command were increased to 8 messages and 2000 bytes, respectively.
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF5	This feature was integrated into Cisco IOS Release 12.2(18)SXF5.
12.2(33)SRB	This feature was integrated into Cisco IOS Release 12.2(33)SRB.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

Prerequisites for RSVP Refresh Reduction and Reliable Messaging

RSVP must be configured on two or more devices within the network before you can use the RSVP Refresh Reduction and Reliable Messaging feature.

Restrictions for RSVP Refresh Reduction and Reliable Messaging

Multicast flows are not supported for the reliable messages and summary refresh features.

Information About RSVP Refresh Reduction and Reliable Messaging

Feature Design of RSVP Refresh Reduction and Reliable Messaging

RSVP is a network-control, soft-state protocol that enables Internet applications to obtain special qualities of service (QoS) for their data flows. As a soft-state protocol, RSVP requires that state be periodically refreshed. If refresh messages are not transmitted during a specified interval, RSVP state automatically times out and is deleted.

In a network that uses RSVP signaling, reliability and latency problems occur when an RSVP message is lost in transmission. A lost RSVP setup message can cause a delayed or failed reservation; a lost RSVP refresh message can cause a delay in the modification of a reservation or in a reservation timeout. Intolerant applications can fail as a result.

Reliability problems can also occur when there is excessive RSVP refresh message traffic caused by a large number of reservations in the network. Using summary refresh messages can improve reliability by significantly reducing the amount of RSVP refresh traffic.

Note RSVP packets consist of headers that identify the types of messages, and object fields that contain attributes and properties describing how to interpret and act on the content.

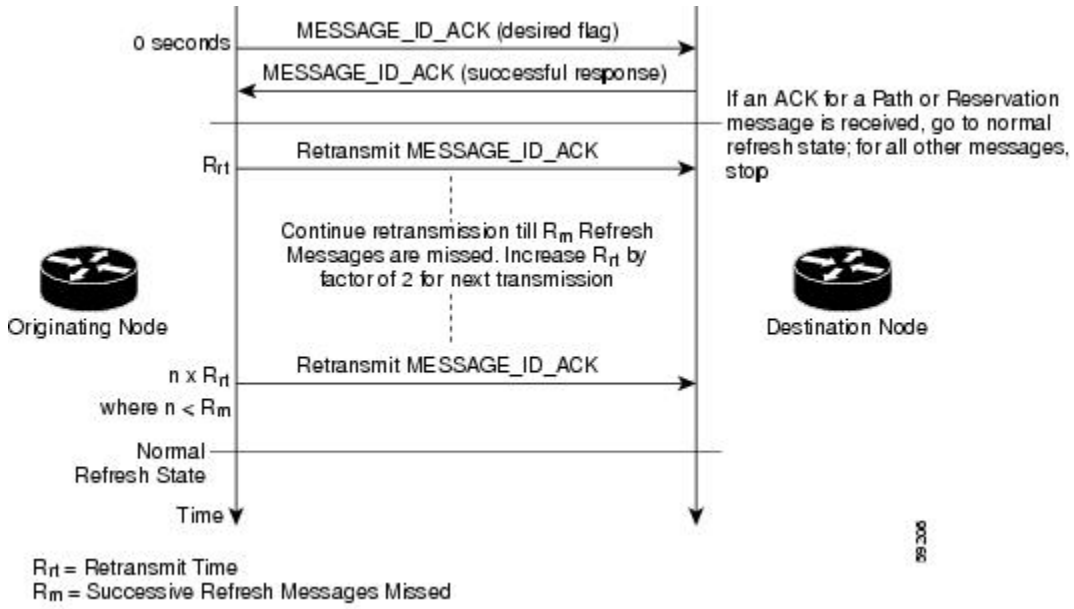
Types of Messages in RSVP Refresh Reduction and Reliable Messaging

The RSVP Refresh Reduction and Reliable Messaging feature (see the figure below) includes refresh reduction, which improves the scalability, latency, and reliability of RSVP signaling by introducing the following extensions:

- Reliable messages (MESSAGE_ID, MESSAGE_ID_ACK objects, and ACK messages)

- Bundle messages (reception and processing only)
- Summary refresh messages (MESSAGE_ID_LIST and MESSAGE_ID_NACK objects)

Figure 155: RSVP Refresh Reduction and Reliable Messaging



Reliable Messages

The reliable messages extension supports dependable message delivery among neighboring devices by implementing an acknowledgment mechanism that consists of a MESSAGE_ID object and a MESSAGE_ID_ACK object. The acknowledgments can be transmitted in an ACK message or piggybacked in other RSVP messages.

Each RSVP message contains one MESSAGE_ID object. If the ACK_Desired flag field is set within the MESSAGE_ID object, the receiver transmits a MESSAGE_ID_ACK object to the sender to confirm delivery.

Bundle Messages

A bundle message consists of several standard RSVP messages that are grouped into a single RSVP message.

A bundle message must contain at least one submessage. A submessage can be any RSVP message type other than another bundle message. Submessage types include Path, PathErr, Resv, ResvTear, ResvErr, ResvConf, and ACK.

Bundle messages are addressed directly to the RSVP neighbor. The bundle header immediately follows the IP header, and there is no intermediate transport header.

When a device receives a bundle message that is not addressed to one of its local IP addresses, it forwards the message.

Note Bundle messages can be received, but not sent.

Summary Refresh Messages

A summary refresh message supports the refreshing of RSVP state without the transmission of conventional Path and Resv messages. Therefore, the amount of information that must be transmitted and processed to maintain RSVP state synchronization is greatly reduced.

A summary refresh message carries a set of MESSAGE_ID objects that identify the Path and Resv states that should be refreshed. When an RSVP node receives a summary refresh message, the node matches each received MESSAGE_ID object with the locally installed Path or Resv state. If the MESSAGE_ID objects match the local state, the state is updated as if a standard RSVP refresh message were received. However, if a MESSAGE_ID object does not match the receiver's local state, the receiver notifies the sender of the summary refresh message by transmitting a MESSAGE_ID_NACK object.

When a summary refresh message is used to refresh the state of an RSVP session, the transmission of conventional refresh messages is suppressed. The summary refresh extension cannot be used for a Path or Resv message that contains changes to a previously advertised state. Also, only a state that was previously advertised in Path or Resv messages containing MESSAGE_ID objects can be refreshed by using a summary refresh message.

Benefits of RSVP Refresh Reduction and Reliable Messaging

Enhanced Network Performance

Refresh reduction reduces the volume of steady-state network traffic generated, the amount of CPU resources used, and the response time, thereby enhancing network performance.

Improved Message Delivery

The MESSAGE_ID and the MESSAGE_ID_ACK objects ensure the reliable delivery of messages and support rapid state refresh when a network problem occurs. For example, MESSAGE_ID_ACK objects are used to detect link transmission losses.

How to Configure RSVP Refresh Reduction and Reliable Messaging

Enabling RSVP on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]] **Example:** `Router(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. • The optional **sub-pool**and *sub-pool-kbps*keyword and argument specify subpool traffic and the amount of bandwidth that can be allocated by RSVP flows. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Enabling RSVP Refresh Reduction

Perform the following task to enable RSVP refresh reduction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling refresh reduction**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling refresh reduction** **Example:** `Device(config)# ip rsvp signalling refresh reduction`	Enables refresh reduction.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying RSVP Refresh Reduction and Reliable Messaging

Perform the following task to verify that the RSVP Refresh Reduction and Reliable Messaging feature is functioning.

SUMMARY STEPS

1. **enable**
2. **clear ip rsvp counters** [**confirm**]
3. **show ip rsvp**
4. **show ip rsvp counters [interface** *interface-unit* | **summary** | **neighbor]**
5. **show ip rsvp interface [** *interface-type interface-number* **] [detail]**
6. **show ip rsvp neighbor [detail]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**clear ip rsvp counters** [**confirm**] **Example:** `Device# clear ip rsvp counters`	(Optional) Clears (sets to zero) all IP RSVP counters that are being maintained by the device.
Step 3	**show ip rsvp** **Example:** `Device# show ip rsvp`	(Optional) Displays RSVP rate-limiting, refresh-reduction, and neighbor information.
Step 4	**show ip rsvp counters [interface** *interface-unit* **| summary | neighbor]** **Example:** `Device# show ip rsvp counters summary`	(Optional) Displays the number of RSVP messages that were sent and received on each interface. • The optional **summary** keyword displays the cumulative number of RSVP messages sent and received by the device over all interfaces.
Step 5	**show ip rsvp interface [** *interface-type interface-number* **] [detail]** **Example:** `Device# show ip rsvp interface detail`	(Optional) Displays information about interfaces on which RSVP is enabled including the current allocation budget and maximum available bandwidth. • The optional **detail** keyword displays the bandwidth and signaling parameters.
Step 6	**show ip rsvp neighbor [detail]** **Example:** `Device# show ip rsvp neighbor detail`	(Optional) Displays RSVP-neighbor information including IP addresses. • The optional **detail** keyword displays the current RSVP neighbors and identifies if the neighbor is using IP, User Datagram Protocol (UDP), or RSVP encapsulation for a specified interface or all interfaces.

Configuration Examples for RSVP Refresh Reduction and Reliable Messaging

Example RSVP Refresh Reduction and Reliable Messaging

In the following example, RSVP refresh reduction is enabled:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface Ethernet1
Device(config-if)# ip rsvp bandwidth 7500 7500
Device(config-if)# exit
Device(config)# ip rsvp signalling refresh reduction
Device(config)# end
```

The following example verifies that RSVP refresh reduction is enabled:

```
Device# show running-config
Building configuration...
Current configuration : 1503 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname Device
!
no logging buffered
logging rate-limit console 10 except errors
!
ip subnet-zero
ip cef
!
ip multicast-routing
no ip dhcp-client network-discovery
lcp max-session-starts 0
mpls traffic-eng tunnels
!
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.0
 ip rsvp bandwidth 1705033 1705033
!
interface Tunnel777
 no ip address
 shutdown
!
interface Ethernet0
 ip address 192.168.0.195 255.0.0.0
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 ip address 192.168.5.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim dense-mode
 no ip mroute-cache
 media-type 10BaseT
 ip rsvp bandwidth 7500 7500
!
interface Ethernet2
 ip address 192.168.1.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim dense-mode
 no ip mroute-cache
 media-type 10BaseT
 mpls traffic-eng tunnels
 ip rsvp bandwidth 7500 7500
!
interface Ethernet3
 ip address 192.168.2.2 255.255.255.0
 ip pim dense-mode
 media-type 10BaseT
```

```
 mpls traffic-eng tunnels
!
!
router eigrp 17
 network 192.168.0.0
 network 192.168.5.0
 network 192.168.12.0
 network 192.168.30.0
 auto-summary
 no eigrp log-neighbor-changes
!
ip classless
no ip http server
ip rsvp signalling refresh reduction
!
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
 transport input pad v120 telnet rlogin udptn
!
end
```

Additional References

The following sections provide references related to the RSVP Refresh Reduction and Reliable Messaging feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	*Resource Reservation Protocol*
RFC 2206	*RSVP Management Information Base Using SMIv2*
RFC 2209	*RSVP--Version 1 Message Processing Rules*
RFC 2210	*The Use of RSVP with IETF Integrated Services*
RFC 2211/2212	*Specification of the Controlled-Load Network Element Service*
RFC 2702	*Requirements for Traffic Engineering over MPLS*
RFC 2749	*Common Open Policy Service (COPS) Usage for RSVP*
RFC 2750	*RSVP Extensions for Policy Control*
RFC 2814	*SBM Subnet Bandwidth Manager: A Protocol for RSVP-based Admission Control over IEEE 802-style Networks*
RFC 2961	*RSVP Refresh Overhead Reduction Extensions*
RFC 2996	*Format of the RSVP DCLASS Object*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 100

MPLS Traffic Engineering—Fast Reroute Link and Node Protection

The MPLS Traffic Engineering--Fast Reroute Link and Node Protection feature provides link protection (backup tunnels that bypass only a single link of the label-switched path (LSP)), node protection (backup tunnels that bypass next-hop nodes along LSPs), and Fast Reroute (FRR) features.

Prerequisites for MPLS Traffic Engineering—Fast Reroute Link and Node Protection

Your network must support the following Cisco IOS XE features:

- IP Cisco Express Forwarding
- Multiprotocol Label Switching (MPLS)

Your network must support at least one of the following protocols:

- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)

Before configuring FRR link and node protection, it is assumed that you have done the following tasks but you do not have to already have configured MPLS traffic engineering (TE) tunnels:

- Enabled MPLS TE on all relevant routers and interfaces

- Configured MPLS TE tunnels

Restrictions for MPLS Traffic Engineering—Fast Reroute Link and Node Protection

- Interfaces must use MPLS Global Label Allocation.
- The router's physical interface for MPLS-TE and Fast RR for Gigabit Ethernet (GE), and Packet over SONET (POS) is supported for enabling a 50 millisecond (ms) failover. However, the GE subinterfaces, logical interfaces and copper interface (e.g. Fast Ethernet interface) are not supported for enabling a 50 ms failover (even though they may be configurable). Also, FRR is not configurable on ATM interface.
- The FRR link protect mode failover time is independent of the number of prefixes pointing to the link.
- Cisco IOS-XE does not support QoS on MPLS-TE tunnels.
- Backup tunnel headend and tailend routers must implement FRR as described in draft-pan-rsvp-fastreroute-00.txt.
- Backup tunnels are not protected. If an LSP is actively using a backup tunnel and the backup tunnel fails, the LSP is torn down.
- LSPs that are actively using backup tunnels are not considered for promotion. If an LSP is actively using a backup tunnel and a better backup tunnel becomes available, the active LSP is not switched to the better backup tunnel.
- You cannot enable FRR Hellos on a router that also has Resource Reservation Protocol (RSVP) Graceful Restart enabled.
- MPLS TE LSPs that are FRR cannot be successfully recovered if the LSPs are FRR active and the Point of Local Repair (PLR) router experiences a stateful switchover (SSO).
- The MPLS TE FRR feature is supported on Cisco 4000 Series ISRs; however, the convergence time of 50 milliseconds is not definite.
- MPLS FRR is not supported on the Cisco Catalyst 8500L Edge Platform.

Information About MPLS Traffic Engineering—Fast Reroute Link and Node Protection

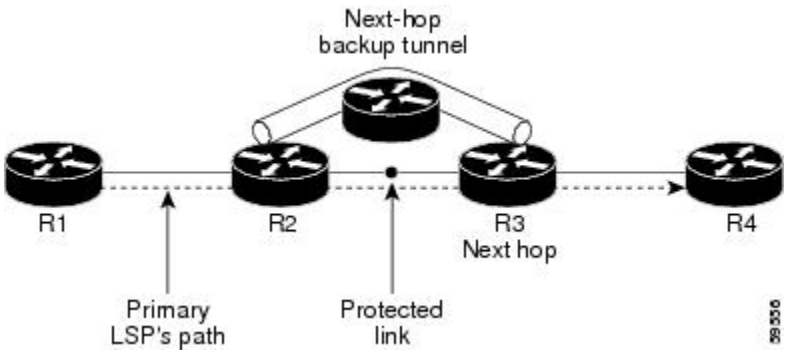
Fast Reroute

Fast Reroute (FRR) is a mechanism for protecting MPLS TE LSPs from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or node.

Link Protection

Backup tunnels that bypass only a single link of the LSP's path provide link protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure. The figure below illustrates an NHOP backup tunnel.

Figure 156: NHOP Backup Tunnel

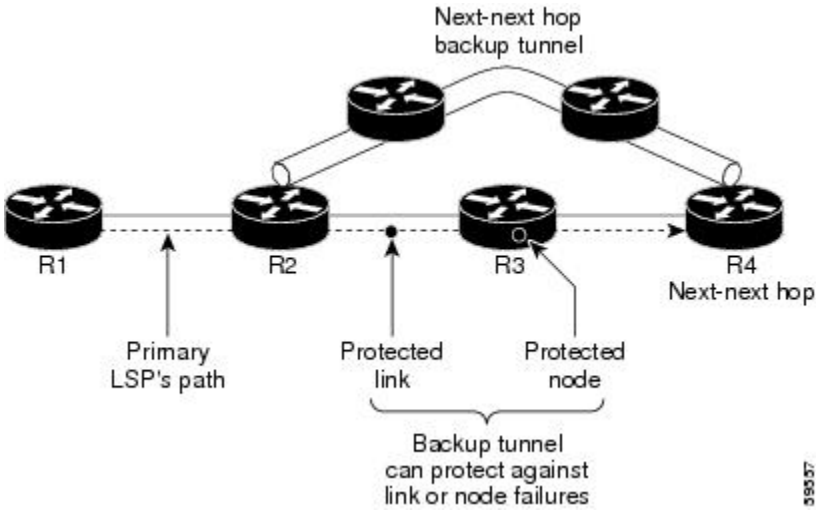


Node Protection

FRR provides node protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called next-next-hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSP paths, thereby bypassing the next-hop node. They protect LSPs if a node along their path fails by enabling the node upstream of the failure to reroute the LSPs and their traffic around the failed node to the next-next hop. FRR supports the use of RSVP Hellos to accelerate the detection of node failures. NNHOP backup tunnels also provide protection from link failures, because they bypass the failed link and the node.

The figure below illustrates an NNHOP backup tunnel.

Figure 157: NNHOP Backup Tunnel



If an LSP is using a backup tunnel and something changes so that the LSP is no longer appropriate for the backup tunnel, the LSP is torn down. Such changes are the following:

- Backup bandwidth of the backup tunnel is reduced.
- Backup bandwidth type of backup tunnel is changed to a type that is incompatible with the primary LSP.
- Primary LSP is modified so that FRR is disabled. (The **no mpls traffic-eng fast-reroute** command is entered.)

Bandwidth Protection

NHOP and NNHOP backup tunnels can be used to provide bandwidth protection for rerouted LSPs. This is referred to as backup bandwidth. You can associate backup bandwidth with NHOP or NNHOP backup tunnels. This informs the router of the amount of backup bandwidth a particular backup tunnel can protect. When a router maps LSPs to backup tunnels, bandwidth protection ensures that an LSP uses a given backup tunnel only if there is sufficient backup bandwidth. The router selects which LSPs use which backup tunnels in order to provide maximum bandwidth protection. That is, the router determines the best way to map LSPs onto backup tunnels in order to maximize the number of LSPs that can be protected. For information about mapping tunnels and assigning backup bandwidth, see the "Backup Tunnel Selection Procedure" section.

LSPs that have the “bandwidth protection desired” bit set have a higher right to select backup tunnels that provide bandwidth protection; that is, those LSPs can preempt other LSPs that do not have that bit set. For more information, see the "Prioritizing Which LSPs Obtain Backup Tunnels with Bandwidth Protection" section.

RSVP Hello Operation

RSVP Hello enables RSVP nodes to detect when a neighboring node is not reachable. This provides node-to-node failure detection. When such a failure is detected, it is handled in a similar manner as a link-layer communication failure.

RSVP Hello can be used by FRR when notification of link-layer failures is not available (for example, with Fast Ethernet), or when the failure detection mechanisms provided by the link layer are not sufficient for the timely detection of node failures.

A node running Hello sends a Hello Request to a neighboring node every interval. If the receiving node is running Hello, it responds with Hello Ack. If four intervals pass and the sending node has not received an Ack or it receives a bad message, the sending node declares that the neighbor is down and notifies FRR.

There are two configurable parameters:

- Hello interval--Use the **ip rsvp signalling hello refresh interval** command.
- Number of acknowledgment messages that are missed before the sending node declares that the neighbor is down--Use the **ip rsvp signalling hello refresh misses** command

RSVP Hello Instance

A Hello instance implements RSVP Hello for a given router interface IP address and remote IP address. A large number of Hello requests are sent; this puts a strain on the router resources. Therefore, create a Hello instance only when it is necessary and delete it when it is no longer needed.

There are two types of Hello instances:

Active Hello Instances

If a neighbor is unreachable when an LSP is ready to be fast rerouted, an active Hello instance is needed. Create an active Hello instance for each neighbor with at least one LSP in this state.

Active Hello instances periodically send Hello Request messages, and expect Hello Ack messages in response. If the expected Ack message is not received, the active Hello instance declares that the neighbor (remote IP address) is unreachable (lost). LSPs traversing that neighbor may be fast rerouted.

If there is a Hello instance with no LSPs for an unreachable neighbor, do not delete the Hello instance. Convert the active Hello instance to a passive Hello instance because there may be an active instance on the neighboring router that is sending Hello requests to this instance.

Passive Hello Instances

Passive Hello instances respond to Hello Request messages (sending Ack messages), but do not initiate Hello Request messages and do not cause LSPs to be fast rerouted. A router with multiple interfaces can run multiple Hello instances to different neighbors or to the same neighbor.

A passive Hello instance is created when a Hello Request is received from a neighbor with a source IP address/destination IP address pair in the IP header for which a Hello instance does not exist.

Delete passive instances if no Hello messages are received for this instance within 10 minutes.

Backup Tunnel Support

Backup tunnel support has the following capabilities:

Backup Tunnels Can Terminate at the Next-Next Hop to Support FRR

Backup tunnels that terminate at the next-next hop protect both the downstream link and node. This provides protection for link and node failures.

Multiple Backup Tunnels Can Protect the Same Interface

There is no limit (except memory limitations) to the number of backup tunnels that can protect a given interface. In many topologies, support for node protection requires supporting multiple backup tunnels per protected interface. These backup tunnels can terminate at the same destination or at different destinations. That is, for a given protected interface, you can configure multiple NHOP or NNHOP backup tunnels. This allows redundancy and load balancing.

In addition to being required for node protection, the protection of an interface by multiple backup tunnels provides the following benefits:

- Redundancy--If one backup tunnel is down, other backup tunnels protect LSPs.
- Increased backup capacity--If the protected interface is a high-capacity link and no single backup path exists with an equal capacity, multiple backup tunnels can protect that one high-capacity link. The LSPs using this link will fail over to different backup tunnels, allowing all of the LSPs to have adequate bandwidth protection during failure (rerouting). If bandwidth protection is not desired, the router spreads LSPs across all available backup tunnels (that is, there is load balancing across backup tunnels).

Backup Tunnels Provide Scalability

A backup tunnel can protect multiple LSPs. Furthermore, a backup tunnel can protect multiple interfaces. This is called many-to-one (N:1) protection. An example of N:1 protection is when one backup tunnel protects 5000 LSPs, each router along the backup path maintains one additional tunnel.

One-to-one protection is when a separate backup tunnel must be used for each LSP needing protection. N:1 protection has significant scalability advantages over one-to-one (1:1) protection. An example of 1:1 protection is when 5000 backup tunnels protect 5000 LSPs, each router along the backup path must maintain state for an additional 5000 tunnels.

Backup Bandwidth Protection

Backup bandwidth protection has the following capabilities:

Bandwidth Protection on Backup Tunnels

Rerouted LSPs not only have their packets delivered during a failure, but the quality of service can also be maintained.

Bandwidth Pool Specifications for Backup Tunnels

You can restrict the types of LSPs that can use a given backup tunnel. Backup tunnels can be restricted so that only LSPs using subpool bandwidth can use them or only LSPs that use global pool bandwidth can use them. This allows different backup tunnels to be used for voice and data. Example: The backup tunnel used for voice could provide bandwidth protection, and the backup tunnel used for data could (optionally) not provide bandwidth protection.

Semidynamic Backup Tunnel Paths

The path of a backup tunnel can be configured to be determined dynamically. This can be done by using the IP explicit address exclusion feature that was added in Release 12.0(14)ST. Using this feature, semidynamic NHOP backup tunnel paths can be specified simply by excluding the protected link; semidynamic NNHOP backup tunnel paths can be configured simply by excluding the protected node.

Prioritizing Which LSPs Obtain Backup Tunnels with Bandwidth Protection

In case there are not enough NHOP or NNHOP backup tunnels or they do not have enough backup bandwidth to protect all LSPs, you can give an LSP priority in obtaining backup tunnels with bandwidth protection. This is especially useful if you want to give LSPs carrying voice a higher priority than those carrying data.

To activate this feature, enter the **tunnel mpls traffic-eng fast-reroute bw-protect**command to set the "bandwidth protection desired" bit. See the configuration task Enabling Fast Reroute on LSPs. The LSPs do not necessarily *receive* bandwidth protection. They have a higher *chance* of receiving bandwidth protection if they need it.

LSPs that do not have the bandwidth protection bit set can be demoted. Demotion is when one or more LSPs are removed from their assigned backup tunnel to provide backup to an LSP that has its bandwidth protection bit set. Demotion occurs only when there is a scarcity of backup bandwidth.

When an LSP is demoted, it becomes unprotected (that is, it no longer has a backup tunnel). During the next periodic promotion cycle, an attempt is made to find the best possible backup tunnels for all LSPs that do not currently have protection, including the LSP that was demoted. The LSP may get protection at the same level or a lower level, or it may get no protection.

For information about how routers determine which LSPs to demote, see the "Backup Protection Preemption Algorithms" section.

RSVP Hello

RSVP Hello enables a router to detect when a neighboring node has gone down but its interface to that neighbor is still operational. This feature is useful when next-hop node failure is not detectable by link layer mechanisms, or when notification of link-layer failures is not available (for example, Gigabit Ethernet). This allows the router to switch LSPs onto its backup tunnels and avoid packet loss.

Fast Reroute Operation

Fast Reroute Activation

Two mechanisms cause routers to switch LSPs onto their backup tunnels:

- Interface down notification
- RSVP Hello neighbor down notification

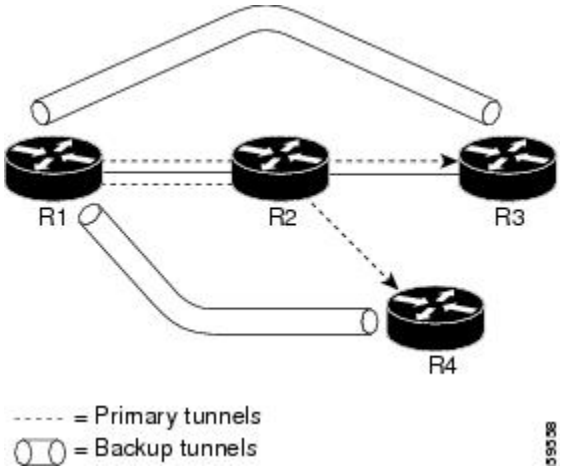
When a router's link or neighboring node fails, the router often detects this failure by an interface down notification. On a GSR Packet over SONET (PoS) interface, this notification is very fast. When a router notices that an interface has gone down, it switches LPSs going out that interface onto their respective backup tunnels (if any).

RSVP Hellos can also be used to trigger FRR. If RSVP Hellos are configured on an interface, messages are periodically sent to the neighboring router. If no response is received, Hellos declare that the neighbor is down. This causes any LSPs going out that interface to be switched to their respective backup tunnels.

Backup Tunnels Terminating at Different Destinations

The figure below illustrates an interface that has multiple backup tunnels terminating at different destinations and demonstrates why, in many topologies, support for node protection requires supporting multiple backup tunnels per protected interface.

Figure 158: Backup Tunnels That Terminate at Different Destinations



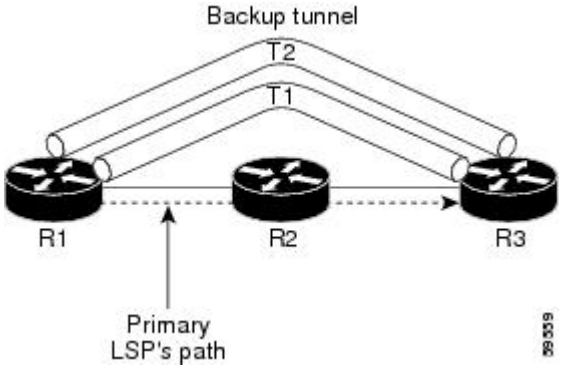
In this illustration, a single interface on R1 requires multiple backup tunnels. LSPs traverse the following routes:

- R1, R2, R3
- R1, R2, R4

To provide protection if node R2 fails, two NNHOP backup tunnels are required: one terminating at R3 and one terminating at R4.

Backup Tunnels Terminating at the Same Destination

The figure below shows how backup tunnels terminating at the same location can be used for redundancy and load balancing. Redundancy and load balancing work for both NHOP and NNHOP backup tunnels.



In this illustration, there are three routers: R1, R2, and R3. At R1, there are two NNHOP backup tunnels (T1 and T2) that go from R1 to R3 without traversing R2.

With redundancy, if R2 fails or the link from R1 to R2 fails, either backup tunnel can be used. If one backup tunnel is down, the other can be used. LSPs are assigned to backup tunnels when the LSPs are first established. This is done before a failure.

With load balancing, if neither backup tunnel has enough bandwidth to back up all LSPs, both tunnels can be used. Some LSPs will use one backup tunnel, other LSPs will use the other backup tunnel. The router decides the best way to fit the LSPs onto the backup tunnels.

Backup Tunnel Selection Procedure

When an LSP is signaled, each node along the LSP path that provides FRR protection for the LSP selects a backup tunnel for the LSP to use if either of the following events occurs:

- The link to the next hop fails.
- The next hop fails.

By having the node select the backup tunnel for an LSP before a failure occurs, the LSP can be rerouted onto the backup tunnel quickly if there is a failure.

For an LSP to be mapped to a backup tunnel, all of the following conditions must exist:

- The LSP is protected by FRR; that is, the LSP is configured with the **tunnel mpls traffic-eng fast-reroute** command.
- The backup tunnel is up.

- The backup tunnel is configured to have an IP address, typically a loopback address.
- The backup tunnel is configured to protect this LSP's outgoing interface; that is, the interface is configured with the **mpls traffic-eng backup-path** command.
- The backup tunnel does not traverse the LSP's protected interface.
- The backup tunnel terminates at the LSP's NHOP or NNHOP. If it is an NNHOP tunnel, it does not traverse the LSP's NHOP.
- The bandwidth protection requirements and constraints, if any, for the LSP and backup tunnel are met. For information about bandwidth protection considerations, see the Bandwidth Protection, on page 2095.

Bandwidth Protection

A backup tunnel can be configured to protect two types of backup bandwidth:

- Limited backup bandwidth--A backup tunnel provides bandwidth protection. The sum of the bandwidth of all LSPs using this backup tunnel cannot exceed the backup tunnel's backup bandwidth. When you assign LSPs to this type of backup tunnel, sufficient backup bandwidth must exist.
- Unlimited backup bandwidth--The backup tunnel does not provide any bandwidth protection (that is, best-effort protection exists). There is no limit to the amount of bandwidth used by the LSPs that are mapped to this backup tunnel. LSPs that allocate zero bandwidth can use only backup tunnels that have unlimited backup bandwidth.

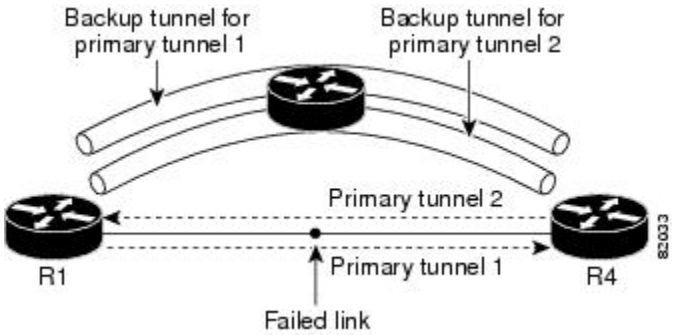
Load Balancing on Limited-Bandwidth Backup Tunnels

There may be more than one backup tunnel that has sufficient backup bandwidth to protect a given LSP. In this case, the router chooses the one that has the least amount of backup bandwidth available. This algorithm limits fragmentation, maintaining the largest amount of backup bandwidth available.

Specifying limited backup bandwidth does not "guarantee" bandwidth protection if there is a link or node failure. For example, the set of NHOP and NNHOP backup tunnels that gets triggered when an interface fails may all share some link on the network topology, and this link may not have sufficient bandwidth to support all LSPs using this set of backup tunnels.

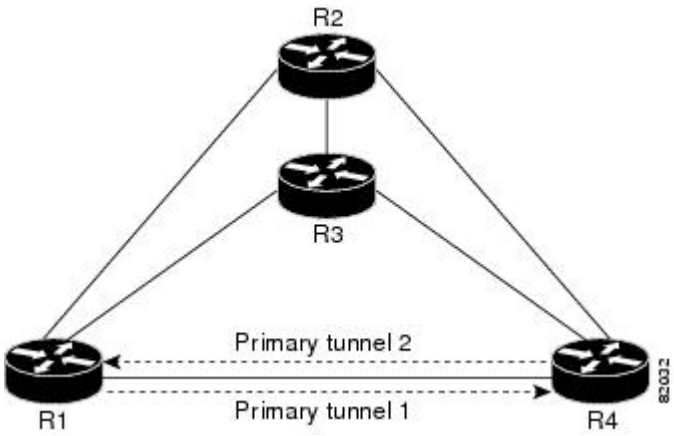
In the figure below, both backup tunnels traverse the same links and hop. When the link between routers R1 and R4 fails, backup tunnels for primary tunnel 1 and primary tunnel 2 are triggered simultaneously. The two backup tunnels may share a link in the network.

Figure 159: Backup Tunnels Share a Link



In the figure below, the backup tunnel for primary tunnel 1 may traverse routers R1-R2-R3-R4, and the backup tunnel for primary tunnel 2 may traverse routers R4-R2-R3-R1. In this case, the link R2-R3 may get overloaded if R1-R4 fails.

Figure 160: Overloaded Link



Load Balancing on Unlimited-Bandwidth Backup Tunnels

More than one backup tunnel, each having unlimited backup bandwidth, can protect a given interface. In this case, when choosing a backup tunnel for a given LSP, the router chooses the backup tunnel that has the least amount of backup bandwidth in use. This algorithm evenly distributes the LSPs across backup tunnels based on an LSP's bandwidth. If an LSP is requesting zero bandwidth, the router chooses the backup tunnel that is protecting the fewest LSPs.

Pool Type and Backup Tunnels

By default, a backup tunnel provides protection for LSPs that allocate from any pool (that is, global or subpool). However, a backup tunnel can be configured to protect only LSPs that use global-pool bandwidth, or only those that use subpool bandwidth.

Tunnel Selection Priorities

This section describes the following:

NHOP Versus NNHOP Backup Tunnels

More than one backup tunnel can protect a given LSP, where one backup tunnel terminates at the LSP's NNHOP, and the other terminates at the LSP's NHOP. In this case, the router chooses the backup tunnel that terminates at the NNHOP (that is, FRR prefers NNHOP over NHOP backup tunnels).

The table below lists the tunnel selection priorities. The first choice is an NNHOP backup tunnel that acquires its bandwidth from a subpool or global pool, and has limited bandwidth. If there is no such backup tunnel, the next choice (2) is a next-next hop backup tunnel that acquires a limited amount of bandwidth from any pool. The preferences go from 1 (best) to 8 (worst), where choice 3 is for an NNHOP backup tunnel with an unlimited amount of subpool or global-pool bandwidth.

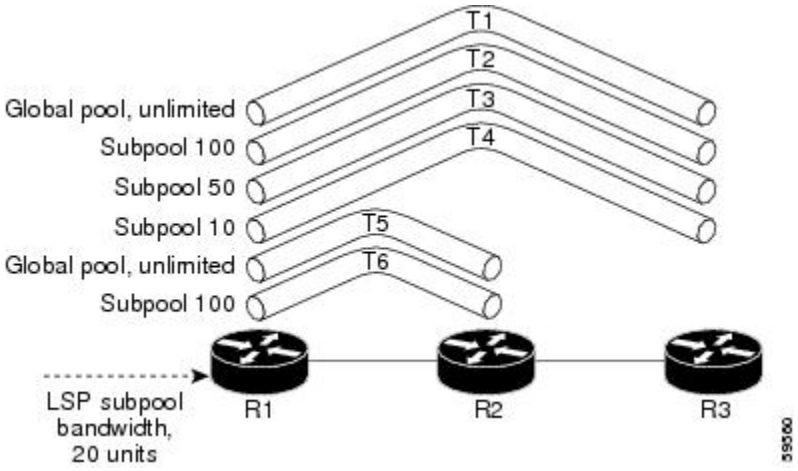
Table 176: Tunnel Selection Priorities

Preference	Backup Tunnel Destination	Bandwidth Pool	Bandwidth Amount
1 (Best)	NNHOP	Subpool or global pool	Limited
2	NNHOP	Any	Limited
3	NNHOP	Subpool or global pool	Unlimited
4	NNHOP	Any	Unlimited
5	NHOP	Subpool or global pool	Limited
6	NHOP	Any	Limited
7	NHOP	Subpool or global pool	Unlimited
8 (Worst)	NHOP	Any	Unlimited

The figure below shows an example of the backup tunnel selection procedure based on the designated amount of global pool and subpool bandwidth currently available.

Note If NHOP and NNHOP backup tunnels do not have sufficient backup bandwidth, no consideration is given to the type of data that the LSP is carrying. For example, a voice LSP may not be protected unless it is signaled before a data LSP. To prioritize backup tunnel usage, see the "Backup Protection Preemption Algorithms" section.

Figure 161: Choosing from Among Multiple Backup Tunnels



In this example, an LSP requires 20 units (kilobits per second) of sub-pool backup bandwidth. The best backup tunnel is selected as follows:

1. Backup tunnels T1 through T4 are considered first because they terminate at the NNHOP.
2. Tunnel T4 is eliminated because it has only ten units of sub-pool backup bandwidth.

3. Tunnel T1 is eliminated because it protects only LSPs using global-pool bandwidth.
4. Tunnel T3 is chosen over T2 because, although both have sufficient backup bandwidth, T3 has the least backup bandwidth available (leaving the most backup bandwidth available on T2).
5. Tunnels T5 and T6 need not be considered because they terminate at an NHOP, and therefore are less desirable than T3, which terminates at an NNHOP.

Promotion

After a backup tunnel has been chosen for an LSP, conditions may change that will cause us to reevaluate this choice. This reevaluation, if successful, is called promotion. Such conditions may include:

1. A new backup tunnel comes up.
2. The currently chosen backup tunnel for this LSP goes down.
3. A backup tunnel's available backup bandwidth increases. For example, an LSP protected by the tunnel has been reoptimized by the headend to use another path.

For cases 1 and 2, the LSP's backup tunnel is evaluated immediately. Case 3 is addressed by periodically reevaluating LSP-to-backup tunnel mappings. By default, background reevaluation is performed every 5 minutes. This interval is configurable via the **mpls traffic-eng fast-reroute timers** command.

Backup Protection Preemption Algorithms

When you set the "bandwidth protection desired" bit for an LSP, the LSP has a higher right to select backup tunnels that provide bandwidth protection and it can preempt other LSPs that do not have that bit set.

If there is insufficient backup bandwidth on NNHOP backup tunnels but not on NHOP backup tunnels, the bandwidth-protected LSP does not preempt NNHOP LSPs; it uses NHOP protection.

If there are multiple LSPs using a given backup tunnel and one or more must be demoted to provide bandwidth, there are two user-configurable methods (algorithms) that the router can use to determine which LSPs are demoted:

- Minimize amount of bandwidth that is wasted.
- Minimize the number of LSPs that are demoted.

For example, If you need ten units of backup bandwidth on a backup tunnel, you can demote one of the following:

- A single LSP using 100 units of bandwidth--Makes available more bandwidth than needed, but results in lots of waste
- Ten LSPs, each using one unit of bandwidth--Results in no wasted bandwidth, but affects more LSPs

The default algorithm is to minimize the number of LSPs that are demoted. To change the algorithm to minimize the amount of bandwidth that is wasted, enter the **mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw** command.

Bandwidth Protection Considerations

There are numerous ways in which bandwidth protection can be ensured. The table below describes the advantages and disadvantages of three methods.

Table 177: Bandwidth Protection Methods

Method	Advantages	Disadvantages
Reserve bandwidth for backup tunnels explicitly.	It is simple.	It is a challenge to allow bandwidth sharing of backup tunnels protecting against independent failures.
Use backup tunnels that are signaled with zero bandwidth.	It provides a way to share bandwidth used for protection against independent failures, so it ensures more economical bandwidth usage.	It may be complicated to determine the proper placement of zero bandwidth tunnels.
Backup bandwidth protection.	It ensures bandwidth protection for voice traffic.	An LSP that does not have backup bandwidth protection can be demoted at any time if there is not enough backup bandwidth and an LSP that has backup bandwidth protection needs bandwidth.

Cisco implementation of FRR does not mandate a particular approach, and it provides the flexibility to use any of the above approaches. However, given a range of configuration choices, be sure that the choices are constant with a particular bandwidth protection strategy.

The following sections describe some important issues in choosing an appropriate configuration:

Using Backup Tunnels with Explicitly Signaled Bandwidth

Two bandwidth parameters must be set for a backup tunnel:

- Actual signaled bandwidth
- Backup bandwidth

To signal bandwidth requirements of a backup tunnel, configure the bandwidth of the backup tunnel by using the **tunnel mpls traffic-eng bandwidth** command.

To configure the backup bandwidth of the backup tunnel, use the **tunnel mpls traffic-eng backup-bw** command.

The signaled bandwidth is used by the LSRs on the path of the backup tunnel to perform admission control and do appropriate bandwidth accounting.

The backup bandwidth is used by the point of local repair (PLR) (that is, the headend of the backup tunnel) to decide how much primary traffic can be rerouted to this backup tunnel if there is a failure.

Both parameters need to be set to ensure proper operation. The numerical value of the signaled bandwidth and the backup bandwidth should be the same.

Protected Bandwidth Pools and the Bandwidth Pool from Which the Backup Tunnel Reserves Its Bandwidth

The **tunnel mpls traffic-eng bandwidth** command allows you to configure the following:

- Amount of bandwidth a backup tunnel reserves
- The DS-TE bandwidth pool from which the bandwidth needs to be reserved

Note Only one pool can be selected (that is, the backup tunnel can explicitly reserve bandwidth from either the global pool or the subpool, but not both).

The **tunnel mpls traffic-eng backup-bw** command allows you to specify the bandwidth pool to which the traffic must belong for the traffic to use this backup tunnel. Multiple pools are allowed.

There is no direct correspondence between the bandwidth pool that is protected and the bandwidth pool from which the bandwidth of the backup tunnel draws its bandwidth.

Bandwidth protection for 10 Kbps of subpool traffic on a given link can be achieved by configuring any of the following command combinations:

- **tunnel mpls traffic-eng bandwidth sub-pool 10**

tunnel mpls traffic-eng backup-bw sub-pool 10

- **tunnel mpls traffic-eng bandwidth global-pool 10**

tunnel mpls traffic-eng backup-bw sub-pool 10 global-pool unlimited

- **tunnel mpls traffic-eng bandwidth global-pool 40**

tunnel mpls traffic-eng backup-bw sub-pool 10 global-pool 30

Using Backup Tunnels Signaled with Zero Bandwidth

Frequently it is desirable to use backup tunnels with zero signaled bandwidth, even when bandwidth protection is required. It may seem that if no bandwidth is explicitly reserved, no bandwidth guarantees can be provided. However, that is not necessarily true.

In the following situation:

- Only link protection is desired.
- Bandwidth protection is desired only for sub-pool traffic.

For each protected link AB with a maximum reservable subpool value of *n* , there may be a path from node A to node B such that the difference between the maximum reservable global and the maximum reservable subpool is at least the value of *n* . If it is possible to find such paths for each link in the network, you can establish all the backup tunnels along such paths without any bandwidth reservations. If there is a single link failure, only one backup tunnel will use any link on its path. Because that path has at least *n* available bandwidth (in the global pool), assuming that marking and scheduling is configured to classify the subpool traffic into a priority queue, the subpool bandwidth is guaranteed.

This approach allows sharing of the global pool bandwidth between backup tunnels protecting independent link failures. The backup tunnels are expected to be used for only a short period of time after a failure (until the headends of affected LSPs reroute those LSPs to other paths with available subpool bandwidth). The probability of multiple unrelated link failures is very small (in the absence of node or shared risk link group (SRLG) failures, which result in multiple link failures). Therefore, it is reasonable to assume that link failures are in practice independent with high probability. This “independent failure assumption” in combination with backup tunnels signaled without explicit bandwidth reservation enables efficient bandwidth sharing that yields substantial bandwidth savings.

Backup tunnels protecting the subpool traffic do now draw bandwidth from any pool. Primary traffic using the global pool can use the entire global pool, and primary traffic using the subpool can use the entire subpool. Yet, subpool traffic has a complete bandwidth guarantee if there is a single link failure.

A similar approach can be used for node and SRLG protection. However, the decision of where to put the backup tunnels is more complicated because both node and SRLG failures effectively result in the simultaneous failure of several links. Therefore, the backup tunnels protecting traffic traversing all affected links cannot be computed independently of each other. The backup tunnels protecting groups of links corresponding to different failures can still be computed independently of each other, which results in similar bandwidth savings.

Signaled Bandwidth Versus Backup Bandwidth

Backup bandwidth is used locally (by the router that is the headend of the backup tunnel) to determine which, and how many, primary LSPs can be rerouted on a particular backup tunnel. The router ensures that the combined bandwidth requirement of these LSPs does not exceed the backup bandwidth.

Therefore, even when the backup tunnel is signaled with zero bandwidth, the backup bandwidth must be configured with the value corresponding to the actual bandwidth requirement of the traffic protected by this backup tunnel. Unlike the case when bandwidth requirements of the backup tunnels are explicitly signaled, the value of the signaled bandwidth (which is zero) is not the same value as the backup bandwidth.

How to Configure MPLS Traffic Engineering—Fast Reroute Link and Node Protection

This section assumes that you want to add FRR protection to a network in which MPLS TE LSPs are configured.

Enabling Fast Reroute on LSPs

LSPs can use backup tunnels only if they have been configured as fast reroutable. To do this, enter the following commands at the headend of each LSP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng fast-reroute** [**bw-protect**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1000`	Enters interface configuration mode for the specified tunnel.
Step 4	**tunnel mpls traffic-eng fast-reroute** [**bw-protect**] **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect`	Enables an MPLS TE tunnel to use an established backup tunnel if there is a link or node failure.

Creating a Backup Tunnel to the Next Hop or to the Next-Next Hop

Creating a backup tunnel is basically no different from creating any other tunnel. To create a backup tunnel to the next hop or to the next-next hop, enter the following commands on the node that will be the headend of the backup tunnel (that is, the node whose downstream link or node may fail). The node on which you enter these commands must be a supported platform. See the Finding Feature Information section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *interface-type interface-number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng path-option** [**protect**] *preference-number*{**dynamic** | **explicit**|{**name** *path-name* | *path-number*}**verbatim**}[**lockdown**]
8. **ip explicit-path name** *word*
9. **exclude-address** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Creates a new tunnel interface and enters interface configuration mode.
Step 4	**ip unnumbered** *interface-type interface-number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Gives the tunnel interface an IP address that is the same as that of interface Loopback0. **Note** This command is not effective until Lookback0 has been configured with an IP address.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 10.3.3.3`	Specifies the IP address of the device where the tunnel will terminate. This address should be the router ID of the device that is the NHOP or NNHOP of LSPs to be protected.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the encapsulation mode of the tunnel to MPLS TE.
Step 7	**tunnel mpls traffic-eng path-option** [**protect**] *preference-number*{**dynamic** | **explicit**|{**name** *path-name* | *path-number*}**verbatim**}[**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 10 explicit avoid-protected-link`	Configures a path option for an MPLS TE tunnel. Enters router configuration mode.
Step 8	**ip explicit-path name** *word* **Example:** `Router(config-router)# ip explicit-path name avoid-protected-link`	Enters the command mode for IP explicit paths and creates the specified path. Enters explicit path command mode.
Step 9	**exclude-address** *ip-address* **Example:** `Router(config-ip-expl-path)# exclude-address 3.3.3.3`	For link protection, specify the IP address of the link to be protected. For node protection, specify the router ID of the node to be protected. **Note** Backup tunnel paths can be dynamic or explicit and they do not have to use exclude-address. Because backup tunnels must avoid the protected link or node, it is convenient to use the **exclude-address** command.

	Command or Action	Purpose
		Note When using the **exclude-address** command to specify the path for a backup tunnel, you must exclude an interface IP address to avoid a link (for creating an NHOP backup tunnel), or a router ID address to avoid a node (for creating an NNHOP backup tunnel).

Assigning Backup Tunnels to a Protected Interface

To assign one or more backup tunnels to a protected interface, enter the following commands on the node that will be the headend of the backup tunnel (that is, the node whose downstream link or node may fail). The node on which you enter these commands must be a supported platform. See the Finding Feature Information section.

Note You must configure the interface to have an IP address and to enable the MPLS TE tunnel feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **mpls traffic-eng backup-path tunnel** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** **Example:** **Example:**	Moves configuration to the physical interface level, directing subsequent configuration commands to the specific physical interface identified by the *type*value. The *slot* and *port* identify the slot and port being configured. The interface must be a supported interface. See the Finding Feature Information section. Enters interface configuration mode.

	Command or Action	Purpose
	Example: **Example:** `Router(config)# interface POS 5/0`	
Step 4	**mpls traffic-eng backup-path tunnel** *interface* **Example:** `Router(config-if)# mpls traffic-eng backup-path tunnel 2`	Allows LSPs going out this interface to use this backup tunnel if there is a link or node failure. **Note** You can enter this command multiple times to associate multiple backup tunnels with the same protected interface.

Associating Backup Bandwidth and Pool Type with a Backup Tunnel

To associate backup bandwidth with a backup tunnel and designate the type of LSP that can use a backup tunnel, enter the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng backup-bw** {*bandwidth* | [**sub-pool** {*bandwidth* | **Unlimited**}] [**global-pool** {*bandwidth* | **Unlimited**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 2`	Enters interface configuration mode for the specified tunnel.

	Command or Action	Purpose
Step 4	**tunnel mpls traffic-eng backup-bw** {*bandwidth* | [**sub-pool** {*bandwidth* | **Unlimited**}] [**global-pool** {*bandwidth* | **Unlimited**}] **Example:** `Router(config-if)# tunnel mpls traffic-eng backup-bw sub-pool 1000`	Associates bandwidth with a backup tunnel and designates whether LSPs that allocate bandwidth from the specified pool can use the tunnel.

Configuring Backup Bandwidth Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **tunnel mpls traffic-eng fast-reroute** [**bw-protect**]
4. **mpls traffic-eng fast-reroute backup-prot-preemption** [**optimize-bw**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters interface configuration mode.
Step 3	**tunnel mpls traffic-eng fast-reroute** [**bw-protect**] **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect`	Enables an MPLS TE tunnel to use an established backup tunnel in the event of a link or node failure. • The **bw-protect** keyword gives an LSP priority for using backup tunnels with bandwidth protection. Enters global configuration mode.
Step 4	**mpls traffic-eng fast-reroute backup-prot-preemption** [**optimize-bw**] **Example:** `Router(config)# mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw`	Changes the backup protection preemption algorithm from minimize the number of LSPs that are demoted to minimize the amount of bandwidth that is wasted.

Configuring an Interface for Fast Link and Node Failure Detection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **pos ais-shut**
5. **pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** `Router(config)# interface pos0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**pos ais-shut** **Example:** `Router(config-if)# pos ais-shut`	Sends the line alarm indication signal (LAIS) when the POS interface is placed in any administrative shutdown state.
Step 5	**pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**} **Example:** `Router(config-if)# pos report lrdi`	Permits selected SONET alarms to be logged to the console for a POS interface.

Verifying That Fast Reroute Is Operational

SUMMARY STEPS

1. **show mpls traffic-eng tunnels brief**
2. **show ip rsvp sender detail**
3. **show mpls traffic-eng fast-reroute database**

4. **show mpls traffic-eng tunnels backup**
5. **show mpls traffic-eng fast-reroute database**
6. **show ip rsvp reservation**

DETAILED STEPS

Step 1 show mpls traffic-eng tunnels brief

Use this command to verify that backup tunnels are up:

Example:

```
Router# show mpls traffic-eng tunnels brief
```

Following is sample output from the **show mpls traffic-eng tunnels brief** command:

Example:

```
Signalling Summary:
    LSP Tunnels Process:           running
    RSVP Process:                  running
    Forwarding:                    enabled
    Periodic reoptimization:       every 3600 seconds, next in 1706 seconds
TUNNEL NAME                      DESTINATION      UP IF     DOWN IF   STATE/PROT
Router_t1                        10.112.0.12      -         PO4/0/1   up/up
Router_t2                        10.112.0.12      -         unknown   up/down
Router_t3                        10.112.0.12      -         unknown   admin-down
Router_t1000                     10.110.0.10      -         unknown   up/down
Router_t2000                     10.110.0.10      -         PO4/0/1   up/up
Displayed 5 (of 5) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Step 2 show ip rsvp sender detail

Use this command to verify that LSPs are protected by the appropriate backup tunnels.

Following is sample output from the **show ip rsvp sender detail** command when the command is entered at the PLR before a failure.

Example:

```
Router# show ip rsvp sender detail

PATH:
 Tun Dest:   10.10.0.6  Tun ID: 100  Ext Tun ID: 10.10.0.1
 Tun Sender: 10.10.0.1  LSP ID: 31
 Path refreshes:
  arriving: from PHOP 10.10.7.1 on Et0/0 every 30000 msecs
 Session Attr:
  Setup Prio: 7, Holding Prio: 7
  Flags: (0x7) Local Prot desired, Label Recording, SE Style
  session Name: R1_t100
 ERO: (incoming)
  10.10.7.2 (Strict IPv4 Prefix, 8 bytes, /32)
  10.10.0.6 (Strict IPv4 Prefix, 8 bytes, /32)
 RRO:
   10.10.7.1/32, Flags:0x0 (No Local Protection)
   10.10.4.1/32, Flags:0x9 (Local Prot Avail/to NNHOP) !Available to NNHOP
   10.10.1.1/32, Flags:0x0 (No Local Protection)
 Traffic params - Rate: 10K bits/sec, Max. burst: 1K bytes
   Min Policed Unit: 0 bytes, Max Pkt Size 4294967295 bytes
```

```
 Fast-Reroute Backup info:
   Inbound  FRR: Not active
   Outbound FRR: No backup tunnel selected
 Path ID handle: 50000416.
 Incoming policy: Accepted. Policy source(s): MPLS/TE
 Status: Proxy-terminated
```

Step 3 **show mpls traffic-eng fast-reroute database**

Enter the **clear ip rsvp hello instance counters**command to verify the following:

- MPLS TE FRR node protection has been enabled.
- A certain type of LSP can use a backup tunnel.

The following command output displays the LSPs that are protected:

Example:

```
Router# show mpls traffic-eng fast-reroute database
Tunnel head end item frr information:
Protected Tunnel      In-label   intf/label       FRR intf/label     Status
Tunnel10              Tun        pos5/0:Untagged  Tu0:12304          ready
Prefix item frr information:
Prefix        Tunnel  In-label   Out intf/label   FRR intf/label     Status
10.0.0.11/32  Tu110   Tun hd     pos5/0:Untagged  Tu0:12304          ready
LSP midpoint frr information:
LSP identifier        In-label   Out intf/label   FRR intf/label     Status
10.0.0.12 1 [459]     16         pos0/1:17        Tu2000:19          ready
```

If LDP is not enabled, separate prefix items are not shown because all prefixes then use a single rewrite. To confirm that a particular IP prefix is FRR protected, even though it is not shown in this display, enter it within the **show mpls forwarding-table** *ip-address* **detail** command. The final line of the display will tell whether that prefix is protected:

Example:

```
Router# show mpls forwarding-table 10.0.0.11 detail

Local    Outgoing    Prefix         Bytes tag    Outgoing     Next Hop
tag      tag or VC   or Tunnel Id   switched     interface
Tun hd   Untagged    10.0.0.11/32   48           pos5/0       point2point
          MAC/Encaps=4/8, MTU=1520, Tag Stack{22}
          48D18847 00016000
          No output feature configured
          Fast Reroute Protection via (Tu0, outgoing label 12304)
```

Step 4 **show mpls traffic-eng tunnels backup**

For backup tunnels to be operational, the LSP must be reroutable. At the headend of the LSP, enter the **show run int tunnel** *tunnel-number* command. The output should include the **tunnel mpls traffic-eng fast-reroute** command. If it does not, enter this command for the tunnel.

On the router where the backup tunnels originate, enter the **show mpls traffic-eng tunnels backup** command. Following is sample command output:

Example:

```
Router# show mpls traffic-eng tunnels backup

Router_t578
  LSP Head, Tunnel578, Admin: up, Oper: up
  Src 10.55.55.55, Dest 10.88.88.88, Instance 1
```

```
  Fast Reroute Backup Provided:
    Protected i/fs: PO1/0, PO1/1, PO3/3
    Protected lsps: 1
    Backup BW: any pool unlimited; inuse: 100 kbps
Router_t5710
  LSP Head, Tunnel5710, Admin: admin-down, Oper: down
  Src 10.55.55.55, Dest 10.7.7.7, Instance 0
  Fast Reroute Backup Provided:
    Protected i/fs: PO1/1
    Protected lsps: 0
    Backup BW: any pool unlimited; inuse: 0 kbps
Router_t5711
  LSP Head, Tunnel5711, Admin up, Oper: up
  Src 10.55.55.55,, Dest 10.7.7.7, Instance 1
  Fast Reroute Backup Provided:
    Protected i/fs: PO1/0
    Protected lsps: 2
    Backup BW: any pool unlimited; inuse: 6010 kbps
```

The command output will allow you to verify the following:

- Backup tunnel exists--Verify that there is a backup tunnel that terminates at this LSP's NHOP or NNHOP. Look for the LSP's NHOP or NNHOP in the Dest field.
- Backup tunnel is up--To verify that the backup tunnel is up, look for "Up" in the State field.
- Backup tunnel is associated with LSP's interface--Verify that the interface for the LSP is allowed to use this backup tunnel. Look for the LSP's output interface in the "protects" field list.
- Backup tunnel has sufficient bandwidth--If you restricted the amount of bandwidth a backup tunnel can hold, verify that the backup tunnel has sufficient bandwidth to hold the LSPs that would use this backup tunnel if there is a failure. The bandwidth of an LSP is defined by the line **tunnel mpls traffic-eng bandwidth** at the headend of the LSP. To determine the available bandwidth on a backup tunnel, look at the "cfg" and "inuse" fields. If there is insufficient backup bandwidth to accommodate the LSPs that would use this backup tunnel in the event of a failure, create an additional backup tunnel or increase the backup bandwidth of the existing tunnel by using the **tunnel mpls traffic-eng bandwidth** command.

Note To determine the sufficient amount of bandwidth, offline capacity planning may be required.

- Backup tunnel has appropriate bandwidth type--If you restricted the type of LSPs (subpool or global pool) that can use this backup tunnel, verify that the LSP is the appropriate type for the backup tunnel. The type of the LSP is defined by the line **tunnel mpls traffic-eng bandwidth**at the headend of this LSP. If this line contains the word "subpool", then it uses sub-pool bandwidth; otherwise, it uses global pool bandwidth. Verify that the type matches the type the backup tunnel can hold by looking in the output of the **tunnel mpls traffic-eng bandwidth**command.

You also can enable debug by entering the **debug ip rsvp fast-reroute** command and the **debug mpls traffic-eng fast-reroute** command on the router that is the headend of the backup tunnel. Then do the following:

a. Enter the **shutdown** command for the primary tunnel.

b. Enter the **no shutdown** command for the primary tunnel.

c. View the debug output.

Step 5 **show mpls traffic-eng fast-reroute database**

Enter the **clear ip rsvp hello instance counters** command to verify the following:

- MPLS TE FRR node protection has been enabled.

- A certain type of LSP can use a backup tunnel.

The following command output displays the LSPs that are protected:

Example:

```
Router# show mpls traffic-eng fast-reroute database
Tunnel head end item frr information:
Protected Tunnel   In-label   intf/label       FRR intf/label      Status
Tunnel10           Tun        pos5/0:Untagged  Tu0:12304           ready
Prefix item frr information:
Prefix        Tunnel  In-label    Out intf/label   FRR intf/label Status
10.0.0.11/32  Tu110   Tun hd      pos5/0:Untagged  Tu0:12304      ready
LSP midpoint frr information:
LSP identifier       In-label Out intf/label   FRR intf/label   Status
10.0.0.12 1 [459]    16       pos0/1:17        Tu2000:19        ready
```

Note If LDP is not enabled, separate prefix items are not shown because all prefixes then use a single rewrite. To confirm that a particular IP prefix is FRR protected, even though it is not shown in this display, enter it within the **show mpls forwarding-table** *ip-address* **detail** command. The final line of the display will tell whether that prefix is protected:

Example:

```
Router# show mpls forwarding-table 10.0.0.11 detail

Local    Outgoing    Prefix         Bytes tag    Outgoing     Next Hop
tag      tag or VC   or Tunnel Id   switched     interface
Tun hd   Untagged    10.0.0.11/32   48           pos5/0       point2point
          MAC/Encaps=4/8, MTU=1520, Tag Stack{22}
          48D18847 00016000
          No output feature configured
          Fast Reroute Protection via (Tu0, outgoing label 12304)
```

Step 6 **show ip rsvp reservation**

Following is sample output from the **show ip rsvp reservation** command entered at the headend of a primary LSP. Entering the command at the headend of the primary LSP shows, among other things, the status of FRR (that is, local protection) at each hop this LSP traverses. The per-hop information is collected in the Record Route Object (RRO) that travels with the Resv message from the tail to the head.

Example:

```
Router# show ip rsvp reservation detail
Reservation:
  Tun Dest: 10.1.1.1  Tun ID: 1  Ext Tun ID: 172.16.1.1
  Tun Sender: 172.16.1.1  LSP ID: 104
  Next Hop: 172.17.1.2 on POS1/0
  Label: 18 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  Average Bitrate is 0 bits/sec, Maximum Burst is 1K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  RRO:
    172.18.1.1/32, Flags:0x1 (Local Prot Avail/to NHOP)
      Label subobject: Flags 0x1, C-Type 1, Label 18
    172.19.1.1/32, Flags:0x0 (Local Prot Avail/In Use/Has BW/to NHOP)
      Label subobject: Flags 0x1, C-Type 1, Label 16
    172.19.1.2/32, Flags:0x0 (No Local Protection)
      Label subobject: Flags 0x1, C-Type 1, Label 0
```

```
Resv ID handle: CD000404.
Policy: Accepted. Policy source(s): MPLS/TE
```

Notice the following about the primary LSP:

- It has protection that uses a NHOP backup tunnel at its first hop.
- It has protection and is actively using an NHOP backup tunnel at its second hop.
- It has no local protection at its third hop.

The RRO display shows the following information for each hop:

- Whether local protection is available (that is, whether the LSP has selected a backup tunnel)
- Whether local protection is in use (that is, whether the LSP is currently using its selected backup tunnel)
- Whether the selected backup tunnel is an NHOP or NNHOP backup tunnel
- Whether the backup tunnel used at this hop provides bandwidth protection

Troubleshooting Tips

This section describes the following:

LSPs Do Not Become Active; They Remain Ready

At a PLR, LSPs transition from Ready to Active if one of the following events occurs:

- Primary interface goes down--If the primary interface (LSP's outbound interface) goes down and the LSP is ready to use a backup tunnel, the LSP will transition to the active state causing its data to flow over the backup tunnel. On some platforms and interface types (for example, GSR POS interfaces), there is fast interface-down logic that detects this event very quickly. On other platforms where this logic does not exist, detection time is slower. On such platforms, it may be desirable to enable RSVP Hello (see the next bulleted item, "Hellos detect next hop is down").
- Hellos detect next hop is down--If Hellos are enabled on the primary interface (LSP's outbound interface), and the LSP's next hop is no longer reachable, the next hop is declared down. This event will cause the LSP to begin actively using its backup tunnel. Notice that a next hop will be declared down even if the primary interface does not go down. For example, if the next hop stops responding due to a reboot or software orr hardware problem, Hellos will trigger the LSPs using this next hop to switch to their backup tunnels. Hellos can also help trigger FRR on interfaces such as Gigabit Ethernet where the interface remains up but is unusable (due to lack of link-layer liveness detection mechanisms).

Primary Tunnel Does Not Select Backup Tunnel That Is Up

If a backup tunnel is up, but it is not selected as a backup tunnel by the primary tunnel (LSP), enter the following commands for the backup tunnel:

- **shutdown**
- **no shutdown**

Note If you change the status of a backup tunnel, the backup tunnel selection algorithm is rerun for the backup tunnel. LSPs that have currently selected (that is, are ready to use) that backup tunnel will be disassociated from it, and then reassociated with that backup tunnel or another backup tunnel. This is generally harmless and usually results in mapping the same LSPs to that backup tunnel. However, if any LSPs are actively using that backup tunnel, shutting down the backup tunnel will tear down those LSPs.

Enhanced RSVP Commands Display Useful Information

The following RSVP commands have been enhanced to display information that can be helpful when you are examining the FRR state or troubleshooting FRR:

- **show ip rsvp request** --Displays upstream reservation state (that is, information related to the Resv messages that this node will send upstream).
- **show ip rsvp reservation** --Displays information about Resv messages received.
- **show ip rsvp sender** --Displays information about path messages being received.

These commands show control plane state; they do not show data state. That is, they show information about RSVP messages (Path and Resv) used to signal LSPs. For information about the data packets being forwarded along LSPs, use the **show mpls forwarding** command.

RSVP Hello Detects When a Neighboring Node Is Not Reachable

The RSVP Hello feature enables RSVP nodes to detect when a neighboring node is not reachable. Use this feature when notification of link-layer failures is not available and unnumbered links are not used, or when the failure detection mechanisms provided by the link layer are not sufficient for timely node failure detection. Hello must be configured both globally on the router and on the specific interface to be operational.

Hello Instances Have Not Been Created

If Hello instances have not been created, do the following:

- Determine if RSVP Hello has been enabled globally on the router. Enter the **ip rsvp signalling hello**(configuration) command.
- Determine if RSVP Hello has been enabled on an interface that the LSPs traverse. Enter the **ip rsvp signalling hello**(interface) command.
- Verify that at least one LSP has a backup tunnel by displaying the output of the **show ip rsvp sender** command. A value of “Ready” indicates that a backup tunnel has been selected.

“No entry at index” (error may self-correct, RRO may not yet have propagated from downstream node of interest)” Error Message Is Printed at the Point of Local Repair

FRR relies on a RRO in Resv messages arriving from downstream. Routers receiving path messages with the SESSION_ATTRIBUTE bit indicating that the LSP is fast-reroutable should include an RRO in the corresponding Resv messages.

If an LSP is configured for FRR, but the Resv arriving from a downstream router contains an incomplete RRO, the “No entry at index (error may self-correct, RRO may not yet have propagated from downstream

node of interest)" message is printed. An incomplete RRO is one in which the NHOP or the NNHOP did not include an entry in the RRO.

This error typically means that backup tunnels to the NHOP or the NNHOP cannot be selected for this LSP because there is insufficient information about the NHOP or NNHOP due to the lack of an RRO entry.

Occasionally there are valid circumstances in which this situation occurs temporarily and the problem is self-corrected. If subsequent Resv messages arrive with a complete RRO, ignore the error message.

To determine whether the error has been corrected, display the RRO in Resv messages by entering the **clear ip rsvp hello instance counters** command. Use an output filter keyword to display only the LSP of interest.

"Couldn't get rsbs" (error may self-correct when Resv arrives)" Error Message Is Printed at the Point of Local Repair

The PLR cannot select a backup tunnel for an LSP until a Resv message has arrived from downstream.

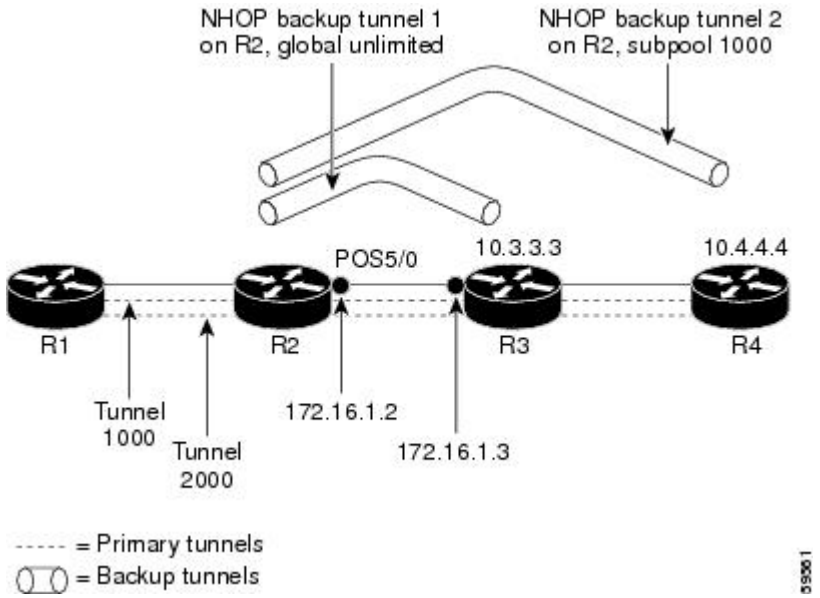
When this error occurs, it typically means that something is wrong. For example, no reservation exists for this LSP. You can troubleshoot this problem by using the **debug ip rsvp reservation** command to enable debug.

Occasionally there are valid circumstances in which this error message occurs and there is no need for concern. One such circumstance is when an LSP experiences a change before any Resv message has arrived from downstream. Changes can cause a PLR to try to select a backup tunnel for an LSP, and the selection will fail (causing this error message) if no Resv message has arrived for this LSP.

Configuration Examples for MPLS Traffic Engineering—Fast Reroute Link and Node Protection

The examples relate to the illustration shown in the figure below.

Figure 162: Backup Tunnels



Enabling Fast Reroute for all Tunnels Example

On router R1, enter interface configuration mode for each tunnel to be protected (Tunnel 1000 and Tunnel 2000). Enable these tunnels to use a backup tunnel in case of a link or node failure along their paths.

Tunnel 1000 will use 10 units of bandwidth from the subpool.

Tunnel 2000 will use five units of bandwidth from the global pool. The "bandwidth protection desired" bit has been set by specifying **bw-prot** in the **tunnel mpls traffic-eng fast-reroute** command.

```
Router(config)# interface Tunnel 1000
Router(config-if)# tunnel mpls traffic-eng fast-reroute
Router(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 10
Router(config)# interface Tunnel2000
Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-prot
Router(config-if)# tunnel mpls traffic-eng bandwidth 5
```

Creating an NHOP Backup Tunnel Example

On router R2, create an NHOP backup tunnel to R3. This backup tunnel should avoid using the link 172.1.1.2.

```
Router(config)# ip explicit-path name avoid-protected-link
Router(cfg-ip-expl-path)# exclude-address 172.1.1.2
Explicit Path name avoid-protected-link:
____1: exclude-address 172.1.1.2
Router(cfg-ip_expl-path)# end
Router(config)# interface Tunnel 1
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel destination 10.3.3.3
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng path-option 10 explicit avoid-protected-link
```

Creating an NNHOP Backup Tunnel Example

On router R2, create an NNHOP backup tunnel to R4. This backup tunnel should avoid R3.

```
Router(config)# ip explicit-path name avoid-protected-node

Router(cfg-ip-expl-path)# exclude-address 10.3.3.3
Explicit Path name avoid-protected-node:
____1: exclude-address 10.3.3.3
Router(cfg-ip_expl-path)# end

Router(config)# interface Tunnel 2

Router(config-if)# ip unnumbered loopback0

Router(config-if)# tunnel destination 10.4.4.4

Router(config-if)# tunnel mode mpls traffic-eng

Router(config-if)# tunnel mpls traffic-eng path-option 10 explicit avoid-protected-node
```

Assigning Backup Tunnels to a Protected Interface

To assign one or more backup tunnels to a protected interface, enter the following commands on the node that will be the headend of the backup tunnel (that is, the node whose downstream link or node may fail). The node on which you enter these commands must be a supported platform. See the Finding Feature Information section.

Note You must configure the interface to have an IP address and to enable the MPLS TE tunnel feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / port*
4. **mpls traffic-eng backup-path tunnel** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / port* **Example:** **Example:** **Example:** **Example:** **Example:** `Router(config)# interface POS 5/0`	Moves configuration to the physical interface level, directing subsequent configuration commands to the specific physical interface identified by the *type*value. The *slot* and *port* identify the slot and port being configured. The interface must be a supported interface. See the Finding Feature Information section. Enters interface configuration mode.
Step 4	**mpls traffic-eng backup-path tunnel** *interface* **Example:**	Allows LSPs going out this interface to use this backup tunnel if there is a link or node failure.

	Command or Action	Purpose
	`Router(config-if)# mpls traffic-eng backup-path tunnel 2`	**Note** You can enter this command multiple times to associate multiple backup tunnels with the same protected interface.

Associating Backup Bandwidth and Pool Type with a Backup Tunnel

To associate backup bandwidth with a backup tunnel and designate the type of LSP that can use a backup tunnel, enter the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng backup-bw** {*bandwidth* | [**sub-pool** {*bandwidth* | **Unlimited**}] [**global-pool** {*bandwidth* | **Unlimited**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 2`	Enters interface configuration mode for the specified tunnel.
Step 4	**tunnel mpls traffic-eng backup-bw** {*bandwidth*	[**sub-pool** {*bandwidth*	**Unlimited**}] [**global-pool** {*bandwidth*	**Unlimited**}] **Example:** `Router(config-if)# tunnel mpls traffic-eng backup-bw sub-pool 1000`	Associates bandwidth with a backup tunnel and designates whether LSPs that allocate bandwidth from the specified pool can use the tunnel.

Configuring Backup Bandwidth Protection Example

In the following example, backup bandwidth protection is configured:

Note This global configuration is required only to change the backup protection preemption algorithm from minimize the number of LSPs that are demoted to minimize the amount of bandwidth that is wasted.

```
Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect
Router(config)# mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw
```

Configuring an Interface for Fast Link and Node Failure Detection Example

In the following example, pos ais-shut is configured:

```
Router(config)# interface pos 0/0

Router(config-if)# pos ais-shut
```

In the following example, report lrdi is configured on OS interfaces:

```
Router(config)# interface pos 0/0
Router(config-if)# pos report lrdi
```

Configuring RSVP Hello and POS Signals Example

Hello must be configured both globally on the router and on the specific interface on which you need FRR protection. To configure Hello, use the following configuration commands:

- **ip rsvp signalling hello** (configuration)--Enables Hello globally on the router.
- **ip rsvp signalling hello** (interface)--Enables Hello on an interface where you need FRR protection.

The following configuration commands are optional:

- **ip rsvp signalling hello dscp** --Sets the differentiated services code point (DSCP) value that is in the IP header of the Hello message.
- **ip rsvp signalling hello refresh misses** --Specifies how many acknowledgments a node can miss in a row before the node considers that communication with its neighbor is down.
- **ip rsvp signalling hello refresh interval** --Configures the Hello request interval.
- **ip rsvp signalling hello statistics** --Enables Hello statistics on the router.

For configuration examples, see the Hello command descriptions in the “Command Reference” section of *MPLS Traffic Engineering (TE): Link and Node Protection, with RSVP Hellos Support* , Release 12.0(24)S.

To configure POS signaling for detecting FRR failures, enter the **pos report all** command or enter the following commands to request individual reports:

```
pos ais-shut
pos report rdool
pos report lais
pos report lrdi
pos report pais
pos report prdi
pos report sd-ber
```

Additional References

The following sections provide references related to the MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) feature.

Related Documents

Related Topic	Document Title
IS-IS	• *Cisco IOS IP Routing Protocols Command Reference* • Configuring a Basic IS-IS Network
MPLS traffic engineering commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
OSPF	• *Cisco IOS IP Routing Protocols Command Reference* • Configuring OSPF
RSVP commands	• *Cisco IOS Multiprotocol Label Switching Command Reference* • *Cisco IOS Quality of Service Solutions Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 4090	Fast Reroute Extensions to RSVP-TE for LSP Tunnels

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering—Fast Reroute Link and Node Protection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 178: Feature Information for MPLS Traffic Engineering—Fast Reroute Link and Node Protection

Feature Name	Releases	Feature Information
MPLS Traffic Engineering--Fast Reroute Link and Node Protection		The MPLS Traffic Engineering--Fast Reroute Link and Node Protection feature supports link protection (backup tunnels that bypass only a single link of the label-switched path (LSP), node protection (backup tunnels that bypass next-hop nodes along LSPs), and the following FRR features: backup tunnel support, backup bandwidth protection, and RSVP Hellos.

Feature Name	Releases	Feature Information
		The following commands were introduced or modified: **clear ip rsvp hello instance counters, clear ip rsvp hello instance statistics, clear ip rsvp hello statistics, debug ip rsvp hello, ip rsvp signalling hello (configuration), ip rsvp signalling hello (interface), ip rsvp signalling hello dscp, ip rsvp signalling hello refresh interval, ip rsvp signalling hello refresh misses, ip rsvp signalling hello statistics, mpls traffic-eng backup-path tunnel, mpls traffic-eng fast-reroute backup-prot-preemption, mpls traffic-eng fast-reroute timers, show ip rsvp fast bw-protect, show ip rsvp fast detail, show ip rsvp hello, show ip rsvp hello instance detail, show ip rsvp hello instance summary, show ip rsvp hello statistics, show ip rsvp interface detail, show ip rsvp request, show ip rsvp reservation, show ip rsvp sender, show mpls traffic tunnel backup, show mpls traffic-eng fast-reroute database, show mpls traffic-eng tunnels, show mpls traffic-eng tunnels summary, tunnel mpls traffic-eng backup-bw, tunnel mpls traffic-eng fast-reroute.**

Glossary

backup bandwidth --The usage of NHOP and NNHOP backup tunnels to provide bandwidth protection for rerouted LSPs.

backup tunnel --An MPLS TE tunnel used to protect other (primary) tunnels' traffic when a link or node failure occurs.

bandwidth --The available traffic capacity of a link.

Cisco Express Forwarding --A means for accelerating the forwarding of packets within a router, by storing route lookup.

enterprise network --A large and diverse network connecting most major points in a company or other organization.

Fast Reroute --Procedures that enable temporary routing around a failed link or node while a new LSP is being established at the headend.

global pool --The total bandwidth allocated to an MPLS traffic engineering link or node.

headend --The router that originates and maintains a given LSP. This is the first router in the LSP's path.

hop --Passage of a data packet between two network nodes (for example, between two routers).

instance --A Hello instance implements the RSVP Hello extensions for a given router interface address and remote IP address. Active Hello instances periodically send Hello Request messages, expecting Hello ACK messages in response. If the expected ACK message is not received, the active Hello instance declares that the neighbor (remote IP address) is unreachable (that is, it is lost). This can cause LSPs crossing this neighbor to be fast rerouted.

interface --A network connection.

Intermediate System-to-Intermediate System --IS-IS. Link-state hierarchical routing protocol that calls for intermediate system (IS) routers to exchange routing information based on a single metric to determine network topology.

link --A point-to-point connection between adjacent nodes. There can be more than one link between adjacent nodes. A link is a network communications channel consisting of a circuit or transmission path and all related equipment between a sender and a receiver. Sometimes referred to as a line or a transmission link.

limited backup bandwidth --Backup tunnels that provide bandwidth protection.

load balancing --A configuration technique that shifts traffic to an alternative link if a certain threshold is exceeded on the primary link. Load balancing is similar to redundancy in that if an event causes traffic to shift directions, alternative equipment must be present in the configuration. In load balancing, the alternative equipment is not necessarily redundant equipment that operates only in the event of a failure.

LSP --label-switched path. A connection between two routers in which MPLS forwards the packets.

merge point --The backup tunnel's tail.

MPLS --Multiprotocol Label Switching. Packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

MPLS global label allocation --There is one label space for all interfaces in the router. For example, label 100 coming in one interface is treated the same as label 100 coming in a different interface.

NHOP --next hop. The next downstream node along an LSP's path.

NHOP backup tunnel --next-hop backup tunnel. Backup tunnel terminating at the LSP's next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link, and is used to protect primary LSPs that were using this link before the failure.

NNHOP --next-next hop. The node after the next downstream node along an LSP's path.

NNHOP backup tunnel --next-next-hop backup tunnel. Backup tunnel terminating at the LSP's next-next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link or node, and is used to protect primary LSPs that were using this link or node before the failure.

node --Endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network. Nodes can be processors, controllers, or workstations.

OSPF --Open Shortest Path First. A link-state hierarchical Interior Gateway Protocol routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

primary LSP --The last LSP originally signaled over the protected interface before the failure. The primary LSP is the LSP before the failure.

primary tunnel --Tunnel whose LSP may be fast rerouted if there is a failure. Backup tunnels cannot be primary tunnels.

promotion --Conditions, such as a new backup tunnel comes up, cause a reevaluation of a backup tunnel that was chosen for an LSP. If the reevaluation is successful, it is called a promotion.

protected interface --An interface that has one or more backup tunnels associated with it.

redundancy --The duplication of devices, services, or connections so that, in the event of a failure, the redundant devices, services, or connections can perform the work of those that failed.

RSVP --Resource Reservation Protocol. A protocol used for signaling requests (setting up reservations) for Internet services by a customer before that customer is permitted to transmit data over that portion of the network.

scalability --An indicator showing how quickly some measure of resource usage increases as a network gets larger.

SRLG --shared risk link group. Sets of links that are likely to go down together.

state --Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

sub-pool --The more restrictive bandwidth in an MPLS traffic engineering link or node. The subpool is a portion of the link or node's overall global pool bandwidth.

tailend --The router upon which an LSP is terminated. This is the last router in the LSP's path.

topology --The physical arrangement of network nodes and media within an enterprise networking structure.

tunnel --Secure communications path between two peers, such as two routers.

unlimited backup bandwidth --Backup tunnels that provide no bandwidth (best-effort) protection (that is, they provide best-effort protection).

CHAPTER 101

MPLS TE Link and Node Protection with RSVP Hellos Support

The MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) feature provides the following Fast Reroute (FRR) capabilities:

- Backup tunnel that terminates at the next-next hop router to protect both the downstream link and node to protect link and node failures. There is no limit (except memory limitations) to the number of backup tunnels that can protect a given interface. A backup tunnel is scalable because it can protect multiple label switched paths (LSPs) and multiple interfaces.
- Backup bandwidth protection allows a priority to be assigned to backup tunnels for LSPs carrying certain kinds of data (such as voice).
- Fast Tunnel Interface Down detection, which forces a "generic" interface tunnel (not specifically a Fast Reroute tunnel) to become disabled immediately if the headend router detects a failed link on an LSP.
- Resource Reservation Protocol (RSVP) Hellos, which are used to accelerate the detection of node failures.

Prerequisites for MPLS TE Link and Node Protection with RSVP Hellos Support

Your network must support the following Cisco IOS XE features to support features described in this document:

- IP Cisco Express Forwarding
- MPLS

Your network must support at least one of the following protocols:

- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)

Restrictions for MPLS TE Link and Node Protection with RSVP Hellos Support

- Interfaces must use MPLS Global Label Allocation.
- Backup tunnel headend and tailend routers must implement FRR as described in this document.
- Backup tunnels are not protected. If an LSP is actively using a backup tunnel and the backup tunnel fails, the LSP is torn down.
- LSPs that are actively using backup tunnels are not considered for promotion. So, if an LSP is actively using a backup tunnel and a better backup tunnel becomes available, the active LSP is not switched to the better backup tunnel.

Information About MPLS TE Link and Node Protection with RSVP Hellos Support

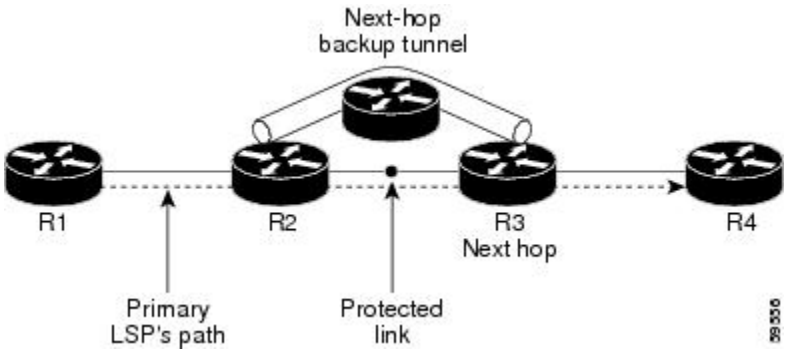
Fast Reroute

Fast Reroute (FRR) is a mechanism for protecting MPLS TE LSPs from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

Link Protection

Backup tunnels that bypass only a single link of the LSP's path provide Link Protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure. The figure below illustrates an NHOP backup tunnel.

Figure 163: NHOP Backup Tunnel

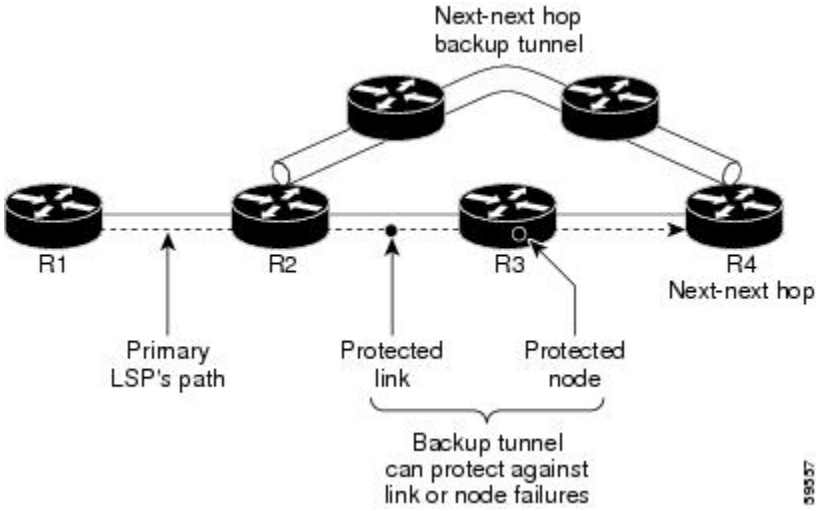


Node Protection

FRR provides Node Protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called next-next-hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSP paths, thereby bypassing the next-hop node. They protect LSPs if a node along their path fails by enabling the node upstream of the failure to reroute the LSPs and their traffic around the failed node to the next-next hop. FRR supports the use of RSVP Hellos to accelerate the detection of node failures. NNHOP backup tunnels also provide protection from link failures, because they bypass the failed link in addition to the node.

The figure below illustrates an NNHOP backup tunnel.

Figure 164: NNHOP Backup Tunnel



If an LSP is using a backup tunnel and something changes so that the LSP is no longer appropriate for the backup tunnel, the LSP is torn down. Such changes include the following:

- Backup bandwidth of the backup tunnel is reduced.
- Backup bandwidth type of backup tunnel is changed to a type that is incompatible with the primary LSP.
- Primary LSP is modified so that FRR is disabled. (The **no mpls traffic-eng fast-reroute**command is entered.)

Bandwidth Protection

NHOP and NNHOP backup tunnels can be used to provide bandwidth protection for rerouted LSPs. This is referred to as backup bandwidth. You can associate backup bandwidth with NHOP or NNHOP backup tunnels. This informs the router of the amount of backup bandwidth a particular backup tunnel can protect. When a router maps LSPs to backup tunnels, bandwidth protection ensures that an LSP uses a given backup tunnel only if there is sufficient backup bandwidth. The router selects which LSPs use which backup tunnels to provide maximum bandwidth protection. That is, the router determines the best way to map LSPs onto backup tunnels to maximize the number of LSPs that can be protected. .

LSPs that have the “bandwidth protection desired” bit set have a higher right to select backup tunnels that provide bandwidth protection; that is, those LSPs can preempt other LSPs that do not have that bit set. For more information, see the "Prioritizing Which LSPs Obtain Backup Tunnels with Bandwidth Protection" section.

Fast Tunnel Interface Down Detection

Fast Tunnel Interface Down detection forces a “generic” interface tunnel (not specifically a Fast Reroute tunnel) to become disabled immediately if the headend router detects a failed link on an LSP.

This feature is configured with the **tunnel mpls traffic-eng interface down delay** command. If this feature is not configured, there is a delay before the tunnel becomes unoperational and before the traffic uses an alternative path chosen by the headend/midpoint router to forward the traffic. This is acceptable for data traffic, but not for voice traffic because it relies on the TE tunnel to go down as soon as the LSP goes down.

RSVP Hello

RSVP Hellos are described in the following sections:

RSVP Hello Operation

RSVP Hello enables RSVP nodes to detect when a neighboring node is not reachable. This provides node-to-node failure detection. When such a failure is detected, it is handled in a similar manner as a link-layer communication failure.

RSVP Hello can be used by FRR when notification of link-layer failures is not available (for example, with Fast Ethernet), or when the failure detection mechanisms provided by the link layer are not sufficient for the timely detection of node failures.

A node running Hello sends a Hello Request to a neighboring node every interval. If the receiving node is running Hello, it responds with Hello Ack. If four intervals pass and the sending node has not received an Ack or it receives a bad message, the sending node declares that the neighbor is down and notifies FRR.

There are two configurable parameters:

- Hello interval, by using the **ip rsvp signalling hello refresh interval**command
- Number of acknowledgment messages that are missed before the sending node declares that the neighbor is down, by using the **ip rsvp signalling hello refresh misses** command

Note If a router's CPU utilization is high due to frequent RSVP Hello processing, there may be false failures due to Hello messages that are not transmitted.

Hello Instance

A Hello instance implements RSVP Hello for a given router interface address and remote IP address. A Hello instance is expensive because of the large number of Hello requests that are sent and the strains they put on the router resources. Therefore, create a Hello instance only when it is necessary and delete it when it is no longer needed.

There are two types of Hello instances:

- Active Hello Instances
- Passive Hello Instances

Active Hello Instances

If a neighbor is unreachable when an LSP is ready to be fast rerouted, an active Hello instance is needed. Create an active Hello instance for each neighbor with at least one LSP in this state.

Active Hello instances periodically send Hello Request messages, and expect Hello Ack messages in response. If the expected Ack message is not received, the active Hello instance declares that the neighbor (remote IP address) is unreachable (lost). LSPs traversing that neighbor may be fast rerouted.

If there is a Hello instance with no LSPs for an unreachable neighbor, do not delete the Hello instance. Convert the active Hello instance to a passive Hello instance because there may be an active instance on the neighboring router that is sending Hello requests to this instance.

Passive Hello Instances

Passive Hello instances respond to Hello Request messages (sending Ack messages), but do not initiate Hello Request messages and do not cause LSPs to be fast rerouted. A router with multiple interfaces can run multiple Hello instances to different neighbors or to the same neighbor.

A passive Hello instance is created when a Hello Request is received from a neighbor with a source IP address/destination IP address pair in the IP header for which a Hello instance does not exist.

Delete passive instances if no Hello messages are received for this instance within 10 minutes.

Hello Commands

RSVP Hello comprises the following commands. For detailed command descriptions, refer to Cisco IOS Multiprotocol Label Switching Command Reference.

- RSVP Hello configuration commands
- RSVP Hello statistics commands
- RSVP Hello show commands
- RSVP Hello debug commands

Features of MPLS TE Link and Node Protection with RSVP Hellos Support

MPLS TE Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) includes the following features:

Backup Tunnel Support

Backup tunnel support has the following capabilities:

Backup Tunnels Can Terminate at the Next-Next Hop to Support FRR

Backup tunnel that terminates at the next-next hop router to protect both the downstream link and node to protect link and node failures. .

Multiple Backup Tunnels Can Protect the Same Interface

There is no limit (except memory limitations) to the number of backup tunnels that can protect a given interface. In many topologies, support for Node Protection requires supporting multiple backup tunnels per protected interface. These backup tunnels can terminate at the same destination or at different destinations. That is, for a given protected interface, you can configure multiple NHOP or NNHOP backup tunnels. This allows redundancy and load balancing.

In addition to being required for Node Protection, this feature provides the following benefits:

- Redundancy--If one backup tunnel is down, other backup tunnels protect LSPs.
- Increased backup capacity--If the protected interface is a high-capacity link and no single backup path exists with an equal capacity, multiple backup tunnels can protect that one high-capacity link. The LSPs using this link will fail over to different backup tunnels, allowing all of the LSPs to have adequate bandwidth protection during failure (rerouting). If bandwidth protection is not desired, the router spreads LSPs across all available backup tunnels (that is, there is load balancing across backup tunnels). .

Scalability

A backup tunnel is scalable because it can protect multiple LSPs and multiple interfaces. It provides many-to-one (N:1) protection, which has significant scalability advantages over one-to-one (1:1) protection, where a separate backup tunnel must be used for each LSP needing protection.

Example of 1:1 protection: When 5,000 backup tunnels protect 5,000 LSPs, each router along the backup path must maintain state for an additional 5,000 tunnels.

Example of N:1 protection: When one backup tunnel protects 5,000 LSPs, each router along the backup path maintains one additional tunnel.

Backup Bandwidth Protection

Backup bandwidth protection has the following capabilities:

Bandwidth Protection on Backup Tunnels

Rerouted LSPs not only have their packets delivered during a failure, but the quality of service can also be maintained.

Bandwidth Pool Specifications for Backup Tunnels

You can restrict the types of LSPs that can use a given backup tunnel. Backup tunnels can be restricted so that only LSPs using subpool bandwidth can use them or only LSPs that use global pool bandwidth can use them. This allows different backup tunnels to be used for voice and data. Example: The backup tunnel used for voice could provide bandwidth protection, and the backup tunnel used for data could (optionally) not provide bandwidth protection.

Semidynamic Backup Tunnel Paths

The path of a backup tunnel can be configured to be determined dynamically. This can be done by using the IP explicit address exclusion feature that was added in Release 12.0(14)ST. Using this feature, semidynamic NHOP backup tunnel paths can be specified simply by excluding the protected link; semidynamic NNHOP backup tunnel paths can be configured simply by excluding the protected node.

Prioritizing Which LSPs Obtain Backup Tunnels with Bandwidth Protection

In case there are not enough NHOP or NNHOP backup tunnels or they do not have enough backup bandwidth to protect all LSPs, you can give an LSP priority in obtaining backup tunnels with bandwidth protection. This is especially useful if you want to give LSPs carrying voice a higher priority than those carrying data.

To activate this feature, enter the **tunnel mpls traffic-eng fast-reroute bw-protect**command to set the “bandwidth protection desired” bit. See the configuration task Enabling Fast Reroute on LSPs. The LSPs do not necessarily *receive* bandwidth protection. They have a higher *chance* of receiving bandwidth protection if they need it.

LSPs that do not have the bandwidth protection bit set can be demoted. Demotion is when one or more LSPs are removed from their assigned backup tunnel to provide backup to an LSP that has its bandwidth protection bit set. Demotion occurs only when there is a scarcity of backup bandwidth.

When an LSP is demoted, it becomes unprotected (that is, it no longer has a backup tunnel). During the next periodic promotion cycle, an attempt is made to find the best possible backup tunnels for all LSPs that do not currently have protection, including the LSP that was demoted. The LSP may get protection at the same level or a lower level, or it may get no protection.

For information about how routers determine which LSPs to demote, see the "Backup Protection Preemption Algorithms" section.

RSVP Hello

RSVP Hello enables a router to detect when a neighboring node has gone down but its interface to that neighbor is still operational. This feature is useful when next-hop node failure is not detectable by link layer mechanisms, or when notification of link-layer failures is not available. This allows the router to switch LSPs onto its backup tunnels and avoid packet loss.

For a more detailed description of RSVP Hello, see the RSVP Hello, on page 2128.

Fast Reroute Operation

This section describes the following:

Fast Reroute Activation

Three mechanisms cause routers to switch LSPs onto their backup tunnels:

- Interface down notification
- Loss of Signal
- RSVP Hello neighbor down notification

When a router's link or neighboring node fails, the router often detects this failure by an interface down notification. On a Packet over SONET (POS) interface, this notification is very fast. When a router notices that an interface has gone down, it switches LPSs going out that interface onto their respective backup tunnels (if any).

Unlike POS interfaces, Gigabit Ethernet does not have any alarms to detect link failures. If a link is down due to a cut cable or because the remote end shuts its laser, the optics module (GBIC or SFPs) on the Gigabit Ethernet card detects a loss of signal (LOS). The LOS is used as a mechanism to detect the failure and begin the switchover.

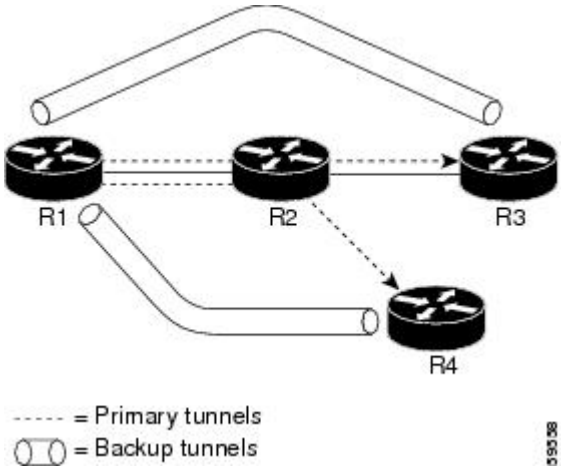
RSVP Hellos can also be used to trigger FRR. If RSVP Hellos are configured on an interface, messages are periodically sent to the neighboring router. If no response is received, Hellos declare that the neighbor is down. This causes any LSPs going out that interface to be switched to their respective backup tunnels.

Fast Reroute also works over ATM interfaces. The interfaces must use RSVP Hello to detect failures.

Backup Tunnels Terminating at Different Destinations

The figure below illustrates an interface that has multiple backup tunnels terminating at different destinations and demonstrates why, in many topologies, support for Node Protection requires supporting multiple backup tunnels per protected interface.

Figure 165: Backup Tunnels that Terminate at Different Destinations



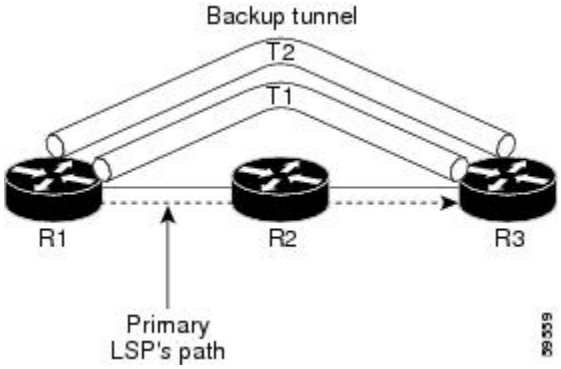
In this illustration, a single interface on R1 requires multiple backup tunnels. LSPs traverse the following routes:

- R1, R2, R3
- R1, R2, R4

To provide protection if node R2 fails, two NNHOP backup tunnels are required: one terminating at R3 and one terminating at R4.

Backup Tunnels Terminating at the Same Destination

The figure below shows how backup tunnels terminating at the same location can be used for redundancy and load balancing. Redundancy and load balancing work for both NHOP and NNHOP backup tunnels.



In this illustration, there are three routers: R1, R2, and R3. At R1, there are two NNHOP backup tunnels (T1 and T2) that go from R1 to R3 without traversing R2.

With redundancy, if R2 fails or the link from R1 to R2 fails, either backup tunnel can be used. If one backup tunnel is down, the other can be used. LSPs are assigned to backup tunnels when the LSPs are first established. This is done before a failure.

With load balancing, if neither backup tunnel has enough bandwidth to back up all LSPs, both tunnels can be used. Some LSPs will use one backup tunnel, other LSPs will use the other backup tunnel. The router decides the best way to fit the LSPs onto the backup tunnels.

Backup Tunnel Selection Procedure

When an LSP is signaled, each node along the LSP path that provides FRR protection for the LSP selects a backup tunnel for the LSP to use if either of the following events occurs:

- The link to the next hop fails.
- The next hop fails.

By having the node select the backup tunnel for an LSP before a failure occurs, the LSP can be rerouted onto the backup tunnel quickly if there is a failure.

For an LSP to be mapped to a backup tunnel, all of the following conditions must exist:

- The LSP is protected by FRR; that is, the LSP is configured with the **tunnel mpls traffic-eng fast-reroute** command.
- The backup tunnel is up.
- The backup tunnel is configured to have an IP address, typically a loopback address.
- The backup tunnel is configured to protect this LSP's outgoing interface; that is, the interface is configured with the **mpls traffic-eng backup-path** command.
- The backup tunnel does not traverse the LSP's protected interface.
- The backup tunnel terminates at the LSP's NHOP or NNHOP. If it is an NNHOP tunnel, it does not traverse the LSP's NHOP.

- The bandwidth protection requirements and constraints, if any, for the LSP and backup tunnel are met. For information about bandwidth protection considerations, see the Bandwidth Protection, on page 2095.

Bandwidth Protection

A backup tunnel can be configured to protect two types of backup bandwidth:

- Limited backup bandwidth--A backup tunnel provides bandwidth protection. The sum of the bandwidth of all LSPs using this backup tunnel cannot exceed the backup tunnel's backup bandwidth. When assigning LSPs to this type of backup tunnel, sufficient backup bandwidth must exist.
- Unlimited backup bandwidth--The backup tunnel does not provide any bandwidth protection (that is, best-effort protection exists). There is no limit to the amount of bandwidth used by the LSPs that are mapped to this backup tunnel. LSPs that allocate zero bandwidth can only use backup tunnels that have unlimited backup bandwidth.

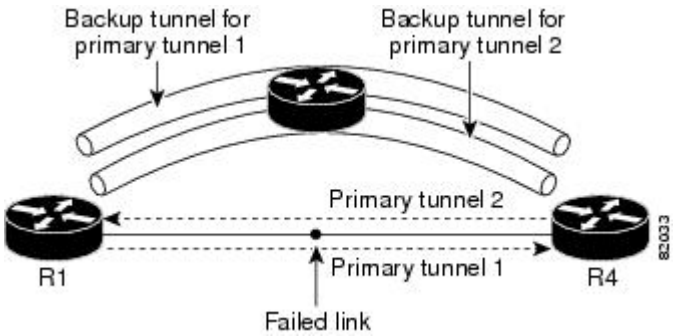
Load Balancing on Limited-bandwidth Backup Tunnels

There may be more than one backup tunnel that has sufficient backup bandwidth to protect a given LSP. In this case, the router chooses the one that has the least amount of backup bandwidth available. This algorithm limits fragmentation, maintaining the largest amount of backup bandwidth available.

Specifying limited backup bandwidth does not "guarantee" bandwidth protection if there is a link or node failure. For example, the set of NHOP and NNHOP backup tunnels that gets triggered when an interface fails may all share some link on the network topology, and this link may not have sufficient bandwidth to support all LSPs using this set of backup tunnels.

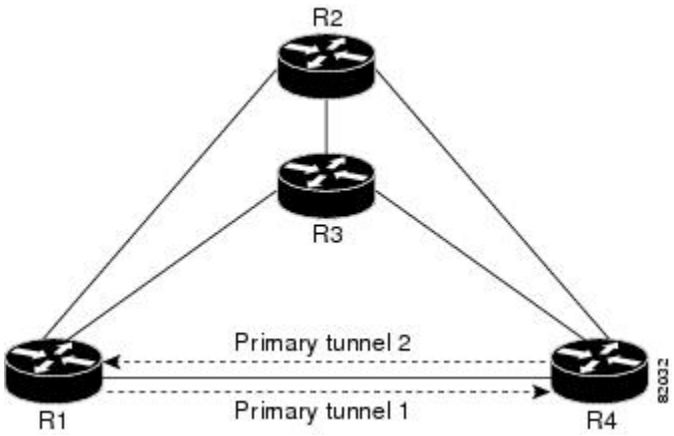
In the figure below, both backup tunnels traverse the same links and hop. When the link between routers R1 and R4 fails, backup tunnels for primary tunnel 1 and primary tunnel 2 are triggered simultaneously. The two backup tunnels may share a link in the network.

Figure 166: Backup Tunnels Share a Link



In the figure below, the backup tunnel for primary tunnel 1 may traverse routers R1-R2-R3-R4, and the backup tunnel for primary tunnel 2 may traverse routers R4-R2-R3-R1. In this case, the link R2-R3 may get overloaded if R1-R4 fails.

Figure 167: Overloaded Link



Load Balancing on Unlimited-bandwidth Backup Tunnels

More than one backup tunnel, each having unlimited backup bandwidth, can protect a given interface. In this case, when choosing a backup tunnel for a given LSP, the router chooses the backup tunnel that has the least amount of backup bandwidth in use. This algorithm evenly distributes the LSPs across backup tunnels based on LSP's bandwidth. If an LSP is requesting zero bandwidth, the router chooses the backup tunnel that is currently protecting the fewest LSPs.

Pool Type and Backup Tunnels

By default, a backup tunnel provides protection for LSPs that allocate from any pool (that is, global or subpool). However, a backup tunnel can be configured to protect only LSPs that use global pool bandwidth, or only those that use subpool bandwidth.

Tunnel Selection Priorities

This section describes the following:

NHOP Versus NNHOP Backup Tunnels

More than one backup tunnel can protect a given LSP, where one backup tunnel terminates at the LSP's NNHOP, and the other terminates at the LSP's NHOP. In this case, the router chooses the backup tunnel that terminates at the NNHOP (that is, FRR prefers NNHOP over NHOP backup tunnels).

The table below lists the tunnel selection priorities. The first choice is an NNHOP backup tunnel that acquires its bandwidth from a subpool or global pool, and has limited bandwidth. If there is no such backup tunnel, the next choice (2) is a next-next hop backup tunnel that acquires a limited amount of bandwidth from any pool. The preferences go from 1 (best) to 8 (worst), where choice 3 is for an NNHOP backup tunnel with an unlimited amount of subpool or global pool bandwidth.

Table 179: Tunnel Selection Priorities

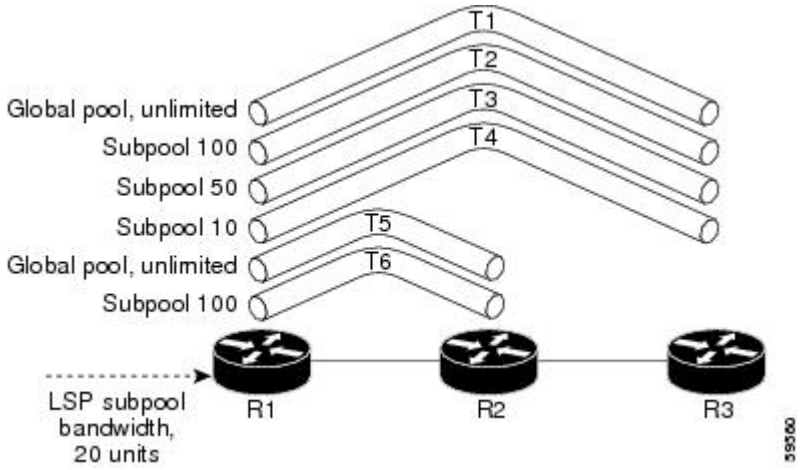
Preference	Backup Tunnel Destination	Bandwidth Pool	Bandwidth Amount
1 (Best)	NNHOP	Subpool or global pool	Limited
2	NNHOP	Any	Limited

Preference	Backup Tunnel Destination	Bandwidth Pool	Bandwidth Amount
3	NNHOP	Subpool or global pool	Unlimited
4	NNHOP	Any	Unlimited
5	NHOP	Subpool or global pool	Limited
6	NHOP	Any	Limited
7	NHOP	Subpool or global pool	Unlimited
8 (Worst)	NHOP	Any	Unlimited

The figure below shows an example of the backup tunnel selection procedure based on the designated amount of global pool and subpool bandwidth currently available.

Note If NHOP and NNHOP backup tunnels do not have sufficient backup bandwidth, no consideration is given to the type of data that the LSP is carrying. For example, a voice LSP may not be protected unless it is signalled before a data LSP. To prioritize backup tunnel usage, see the "Backup Protection Preemption Algorithms" section.

Figure 168: Choosing from Among Multiple Backup Tunnels



In this example, an LSP requires 20 units (kilobits per second) of subpool backup bandwidth. The best backup tunnel is selected as follows:

1. Backup tunnels T1 through T4 are considered first because they terminate at the NNHOP.
2. Tunnel T4 is eliminated because it only has 10 units of subpool backup bandwidth.
3. Tunnel T1 is eliminated because it protects only LSPs using global pool bandwidth.
4. Tunnel T3 is chosen over T2 because, although both have sufficient backup bandwidth, T3 has the least backup bandwidth available (leaving the most backup bandwidth available on T2).

5. Tunnels T5 and T6 need not be considered because they terminate at an NHOP, and therefore are less desirable than T3, which terminates at an NNHOP.

Promotion

After a backup tunnel has been chosen for an LSP, conditions may change that will cause us to reevaluate this choice. This reevaluation, if successful, is called promotion. Such conditions may include:

1. A new backup tunnel comes up.
2. The currently chosen backup tunnel for this LSP goes down.
3. A backup tunnel's available backup bandwidth increases. For example, an LSP protected by the tunnel has been reoptimized by the headend to use another path.
4. A backup tunnel's available backup-bandwidth decreases.

For cases 1 and 2, the LSP's backup tunnel is evaluated immediately. Cases 3 and 4 are addressed by periodically reevaluating LSP-to-backup tunnel mappings. By default, background reevaluation is performed every 5 minutes. This interval is configurable via the **mpls traffic-eng fast-reroute timers** command.

The response to case 4 is as follows:

When the backup tunnel's bandwidth is reduced, promotion will *not* be run so long as the remaining bandwidth is greater than the sum of the bandwidths of all primary paths for which this tunnel is the backup. This policy prevents unnecessary disruption of protection of the primary paths.

When the backup tunnel's bandwidth *does* fall below the required bandwidth needed for it to substitute for all primary paths to which it has been assigned, promotion is run.

Backup Protection Preemption Algorithms

When you set the "bandwidth protection desired" bit for an LSP, the LSP has a higher right to select backup tunnels that provide bandwidth protection and it can preempt other LSPs that do not have that bit set.

If there is insufficient backup bandwidth on NNHOP backup tunnels but not on NHOP backup tunnels, the bandwidth-protected LSP does not preempt NNHOP LSPs; it uses NHOP protection.

If there are multiple LSPs using a given backup tunnel and one or more must be demoted to provide bandwidth, there are two user-configurable methods (algorithms) that the router can use to determine which LSPs are demoted.

- Minimize amount of bandwidth that is wasted.
- Minimize the number of LSPs that are demoted.

For example, If you need 10 units of backup bandwidth on a backup tunnel, you can demote one of the following:

- A single LSP using 100 units of bandwidth--Makes available more bandwidth than needed, but results in lots of waste
- Ten LSPs, each using one unit of bandwidth--Results in no wasted bandwidth, but affects more LSPs

The default algorithm minimizes the number of LSPs that are demoted. To change the algorithm to minimize the amount of bandwidth that is wasted, enter the **mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw** command.

Bandwidth Protection Considerations

There are numerous ways in which bandwidth protection can be ensured. The table below describes the advantages and disadvantages of three methods.

Table 180: Bandwidth Protection Methods

Method	Advantages	Disadvantages
Reserve bandwidth for backup tunnels explicitly.	It is simple.	It is a challenge to allow bandwidth sharing of backup tunnels protecting against independent failures.
Use backup tunnels that are signaled with zero bandwidth.	It provides a way to share bandwidth used for protection against independent failures, so it ensures more economical bandwidth usage.	It may be complicated to determine the proper placement of zero bandwidth tunnels.
Backup bandwidth protection	Ensures bandwidth protection for voice traffic.	An LSP that does not have backup bandwidth protection can be demoted at any time if there is not enough backup bandwidth and an LSP that has backup bandwidth protection needs bandwidth.

Cisco implementation of FRR does not mandate a particular approach, and it provides the flexibility to use any of the above approaches. However, given a range of configuration choices, be sure that the choices are constant with a particular bandwidth protection strategy.

The following sections describe some important issues in choosing an appropriate configuration:

Backup Tunnels with Explicitly Signaled Bandwidth

There are two bandwidth parameters that must be set for a backup tunnel:

- actual signaled bandwidth
- backup-bandwidth

To signal bandwidth requirements of a backup tunnel, configure the bandwidth of the backup tunnel by using the **tunnel mpls traffic-eng bandwidth** command.

To configure the backup bandwidth of the backup tunnel, use the **tunnel mpls traffic-eng backup-bw** command.

The signaled bandwidth is used by the LSRs on the path of the backup tunnel to perform admission control and do appropriate bandwidth accounting.

The backup bandwidth is used by the PLR (the headend of the backup tunnel) to decide how much primary traffic can be rerouted to this backup tunnel if there is a failure.

Both parameters need to be set to ensure proper operation. The numerical value of the signaled bandwidth and the backup-bandwidth should be the same.

Protected Bandwidth Pools and the Bandwidth Pool from Which the Backup Tunnel Reserves Its Bandwidth

The **tunnel mpls traffic-eng bandwidth**command allows you to configure the following:

- Amount of bandwidth a backup tunnel reserves
- The DS-TE bandwidth pool from which the bandwidth needs to be reserved

Note Only one pool can be selected (that is, the backup tunnel can explicitly reserve bandwidth from either the global pool or the subpool, but not both).

The **tunnel mpls traffic-eng backup-bw** command allows you to specify the bandwidth pool to which the traffic must belong for the traffic to use this backup tunnel. Multiple pools are allowed.

There is no direct correspondence between the bandwidth pool that is protected and the bandwidth pool from which the bandwidth of the backup tunnel draws its bandwidth.

Example: In this example, assume the following:

- Bandwidth protection is desired only for subpool traffic, but the best-effort traffic using the global pool does not require bandwidth protection.
- Scheduling is configured so that subpool traffic uses the priority queue, and global pool traffic is served at a lower priority.

Bandwidth protection for 10 Kbps of subpool traffic on a given link can be achieved by any of the following combinations:

- **tunnel mpls traffic-eng bandwidth sub-pool 10**

tunnel mpls traffic-eng backup-bw sub-pool 10

- **tunnel mpls traffic-eng bandwidth global-pool 10**

tunnel mpls traffic-eng backup-bw sub-pool 10 global-pool unlimited

- **tunnel mpls traffic-eng bandwidth global-pool 40**

tunnel mpls traffic-eng backup-bw sub-pool 10 global-pool 30

Backup Tunnels Signaled with Zero Bandwidth

Frequently it is desirable to use backup tunnels with zero signaled bandwidth, even when bandwidth protection is required. It may seem that if no bandwidth is explicitly reserved, no bandwidth guarantees can be provided. However, that is not necessarily true.

In the following situation:

- Only link protection is desired.
- Bandwidth protection is desired only for subpool traffic.

For each protected link AB with a max reservable subpool value of S, there may be a path from node A to node B such that the difference between max reservable global and max reservable subpool is at least S. If it is possible to find such paths for each link in the network, you can establish all the backup tunnels along such paths without any bandwidth reservations. If there is a single link failure, only one backup tunnel will use any

link on its path. Because that path has at least S of available bandwidth (in the global pool), assuming that marking and scheduling is configured to classify the subpool traffic into a priority queue, the subpool bandwidth is guaranteed.

The above approach allows sharing of the global pool bandwidth between backup tunnels protecting independent link failures. The backup tunnels are expected to be used for only a short period of time after a failure (until the headends of affected LSPs reroute those LSPs to other paths with available subpool bandwidth). The probability of multiple unrelated link failures is very small (in the absence of node or SRLG failures, which result in multiple link failures). Therefore, it is reasonable to assume that link failures are in practice independent with high probability. This "independent failure assumption" in combination with backup tunnels signaled without explicit bandwidth reservation enables efficient bandwidth sharing that yields substantial bandwidth savings.

Backup tunnels protecting the subpool traffic do now draw bandwidth from any pool. Primary traffic using the global pool can use the entire global pool, and primary traffic using the subpool can use the entire subpool. Yet, subpool traffic has a complete bandwidth guarantee if there is a single link failure.

A similar approach can be used for node and SRLG protection. However, the decision of where to put the backup tunnels is more complicated because both node and SRLG failures effectively result in the simultaneous failure of several links. Therefore, the backup tunnels protecting traffic traversing all affected links cannot be computed independently of each other. The backup tunnels protecting groups of links corresponding to different failures can still be computed independently of each other, which results in similar bandwidth savings.

Signaled Bandwidth Versus Backup Bandwidth

Backup bandwidth is used locally (by the router that is the headend of the backup tunnel) to determine which, and how many, primary LSPs can be rerouted on a particular backup tunnel. The router ensures that the combined bandwidth requirement of these LSPs does not exceed the backup bandwidth.

Therefore, even when the backup tunnel is signaled with zero bandwidth, the backup bandwidth must be configured with the value corresponding to the actual bandwidth requirement of the traffic protected by this backup tunnel. Unlike the case when bandwidth requirements of the backup tunnels are explicitly signaled, the value of the signaled bandwidth (which is zero) is not the same value as the backup bandwidth.

How to Configure MPLS TE Link and Node Protection with RSVP Hellos Support

This section assumes that you want to add FRR protection to a network in which MPLS TE LSPs are configured.

Make sure that the following tasks have been performed before you perform the configuration tasks, but you do not have to already have configured MPLS TE tunnels:

- Enabled MPLS TE on all relevant routers and interfaces
- Configured MPLS TE tunnels

To review how to configure MPLS TE tunnels, see the Cisco IOS XE Multiprotocol Label Switching Configuration Guide.

The following sections describe how to use FRR to protect LSPs in your network from link or node failures. Each task is identified as either required or optional.

Note You can perform the configuration tasks in any order.

Note An NNHOP backup tunnel must *not* go via the NHOP.

Enabling Fast Reroute on LSPs

LSPs can use backup tunnels only if they have been configured as fast reroutable. To enable fast reroute on an LSP, perform the following task. Enter the commands at the headend of each LSP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng fast-reroute** [**bw-protect**] [**node-protect**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1000`	Enters interface configuration mode for the specified tunnel.
Step 4	**tunnel mpls traffic-eng fast-reroute** [**bw-protect**] [**node-protect**] **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect node-protect`	Enables an MPLS TE tunnel to use an established backup tunnel if there is a link or node failure.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Creating a Backup Tunnel to the Next Hop or to the Next-Next Hop

To create a backup tunnel to the next hop or to the next-next hop, perform the following task. Enter the commands on the node that will be the headend of the backup tunnel (that is, the node whose downstream link or node may fail).

Creating a backup tunnel is basically no different from creating any other tunnel. None of the commands below is new.

Note When using the **exclude-address** command to specify the path for a backup tunnel, you must exclude an interface address to avoid a link (for creating an NHOP backup tunnel), or a router-ID address to avoid a node (for creating an NNHOP backup tunnel).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** *A.B.C.D*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}} [**lockdown**]
8. **ip explicit-path name** *name*
9. **exclude-address** *address*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Creates a new tunnel interface and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered loopback0`	Gives the tunnel interface an IP address that is the same as that of interface Loopback0. **Note** This command is not effective until Lookback0 has been configured with an IP address.
Step 5	**tunnel destination** *A.B.C.D* **Example:** `Router(config-if)# tunnel destination 10.3.3.3`	Specifies the IP address of the device where the tunnel will terminate. • That address should be the router ID of the device that is the NHOP or NNHOP of LSPs to be protected.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets encapsulation mode of the tunnel to MPLS TE.
Step 7	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 300 explicit name avoid-protected-link`	Configures a path option for an MPLS TE tunnel.
Step 8	**ip explicit-path name** *name* **Example:** `Router(config)# ip explicit-path name avoid-protected-link`	Enters the subcommand mode for IP explicit paths to create the named path.
Step 9	**exclude-address** *address* **Example:** `Router(cfg-ip-expl-path)# exclude-address 10.3.3.3`	For Link Protection, specifies the IP address of the link to be protected. • For Node Protection, this command specifies the router ID of the node to be protected. **Note** Backup tunnel paths can be dynamic or explicit and they do not have to use exclude-address. Because backup tunnels must avoid the protected link or node, it is convenient to use an exclude-address.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(cfg-ip-expl-path)# end`	Exits to privileged EXEC mode.

Assigning Backup Tunnels to a Protected Interface

To assign one or more backup tunnels to a protected interface, perform the following task. Enter the following commands on the node that will be the headend of the backup tunnel (that is, the node whose downstream link or node may fail).

Note You must configure the interface to have an IP address and to enable the MPLS TE tunnel feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot / subslot / port* [**.** *subinterface-number*]
4. **mpls traffic-eng backup-path tunnel** *tunnel-id*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / subslot / port* [**.** *subinterface-number*] **Example:** `Router(config)# interface POS1/0/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the chassis slot number. Refer to the appropriate hardware manual for slot information. For SIPs, refer to the platform-specific SPA hardware installation guide or the corresponding “Identifying

	Command or Action	Purpose
		Slots and Subslots for SIPs and SPAs" topic in the platform-specific SPA software configuration guide. • The / *subslot* keyword and argument pair is the secondary slot number on a SIP where a SPA is installed. The slash (/) is required. Refer to the platform-specific SPA hardware installation guide and the corresponding "Specifying the Interface Address on a SPA" topic in the platform-specific SPA software configuration guide for subslot information. • The / *port* keyword and argument pair is the port or interface number. The slash (/) is required. Refer to the appropriate hardware manual for port information. For SPAs, refer to the corresponding "Specifying the Interface Address on a SPA" topics in the platform-specific SPA software configuration guide • The **.** *subinterface-number*keyword and argument pair is the subinterface number in the range 1 to 4294967293. The number that precedes the period (.) must match the number to which this subinterface belongs.
Step 4	**mpls traffic-eng backup-path tunnel** *tunnel-id* **Example:** `Router(config-if)# mpls traffic-eng backup-path tunnel2`	Allows LSPs going out this interface to use this backup tunnel if there is a link or node failure. **Note** You can enter this command multiple times to associate multiple backup tunnels with the same protected interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Associating Backup Bandwidth and Pool Type with a Backup Tunnel

To associate backup bandwidth with a backup tunnel and designate the type of LSP that can use a backup tunnel, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*

4. **tunnel mpls traffic-eng backup-bw** {*bandwidth* | [**sub-pool** {*bandwidth* | **unlimited**}][**global-pool** {*bandwidth* | **unlimited**}]} [**any** {*bandwidth* | **unlimited**}]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 2`	Enters interface configuration mode for the specified tunnel.
Step 4	**tunnel mpls traffic-eng backup-bw** {*bandwidth* | [**sub-pool** {*bandwidth* | **unlimited**}][**global-pool** {*bandwidth* | **unlimited**}]} [**any** {*bandwidth* | **unlimited**}] **Example:** `Router(config-if)# tunnel mpls traffic-eng backup-bw sub-pool 1000`	Associates bandwidth with a backup tunnel and designates whether LSPs that allocate bandwidth from the specified pool can use the tunnel.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuring Backup Bandwidth Protection

To configure the backup bandwidth protection, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng fast-reroute** [**bw-protect**]
5. **exit**
6. **mpls traffic-eng fast-reroute backup-prot-preemption** [**optimize-bw**]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 2`	Enters interface configuration mode for the specified tunnel.
Step 4	**tunnel mpls traffic-eng fast-reroute** [**bw-protect**] **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect`	Enables an MPLS TE tunnel to use an established backup tunnel in the event of a link or node failure. • The **bw-protect** keyword gives an LSP priority for using backup tunnels with bandwidth protection.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 6	**mpls traffic-eng fast-reroute backup-prot-preemption** [**optimize-bw**] **Example:** `Router(config)# mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw`	Changes the backup protection preemption algorithm from minimize the number of LSPs that are demoted to minimize the amount of bandwidth that is wasted.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits to privileged EXEC mode.

Configuring an Interface for Fast Link and Node Failure Detection

To configure an interface for fast link and node failure detection, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type slot / subslot / port* [**.** *subinterface-number*]
4. **pos ais-shut**
5. **pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot / subslot / port* [**.** *subinterface-number*] **Example:** `Router(config)# interface pos0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**pos ais-shut** **Example:** `Router(config-if)# pos ais-shut`	Sends the line alarm indication signal (LAIS) when the Packet-over-SONET (POS) interface is placed in any administrative shutdown state.
Step 5	**pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**} **Example:** `Router(config-if)# pos report lrdi`	Permits selected SONET alarms to be logged to the console for a POS interface.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuring an Interface for Fast Tunnel Interface Down

To configure an interface for fast tunnel interface down, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng interface down delay** *time*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1000`	Configures an interface type and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng interface down delay** *time* **Example:** `Router(config-if)# tunnel mpls traffic-eng interface down delay 0`	Forces a tunnel to go down as soon as the headend router detects that the LSP is down.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Verifying That Fast Reroute Is Operational

To verify that FRR can function, perform the following task.

SUMMARY STEPS

1. **show mpls traffic-eng tunnels brief**
2. **show ip rsvp sender detail**
3. **show mpls traffic-eng fast-reroute database**
4. **show mpls traffic-eng tunnels backup**
5. **show mpls traffic-eng fast-reroute database**
6. **show ip rsvp reservation**

DETAILED STEPS

Step 1 show mpls traffic-eng tunnels brief

Use this command to verify that backup tunnels are up:

Example:

```
Router# show mpls traffic-eng tunnels brief

Signalling Summary:
    LSP Tunnels Process:           running
    RSVP Process:                  running
    Forwarding:                    enabled
    Periodic reoptimization:       every 3600 seconds, next in 1706 seconds
TUNNEL NAME                      DESTINATION      UP IF     DOWN IF   STATE/PROT
Router_t1                        10.112.0.12      -         PO2/0/1   up/up
Router_t2                        10.112.0.12      -         unknown   up/down
Router_t3                        10.112.0.12      -         unknown   admin-down
Router_t1000                     10.110.0.10      -         unknown   up/down
Router_t2000                     10.110.0.10      -         PO2/0/1   up/up
Displayed 5 (of 5) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Step 2 show ip rsvp sender detail

Use this command to verify that LSPs are protected by the appropriate backup tunnels.

Following is sample output from the **show ip rsvp sender detail** command when the command is entered at the PLR before a failure:

Example:

```
Router# show ip rsvp sender detail

PATH:
 Tun Dest:   10.10.0.6  Tun ID: 100  Ext Tun ID: 10.10.0.1
 Tun Sender: 10.10.0.1  LSP ID: 31
 Path refreshes:
  arriving: from PHOP 10.10.7.1 on FE0/0/0 every 30000 msecs
 Session Attr:
  Setup Prio: 7, Holding Prio: 7
  Flags: (0x7) Local Prot desired, Label Recording, SE Style
  session Name: R1_t100
 ERO: (incoming)
  10.10.7.2 (Strict IPv4 Prefix, 8 bytes, /32)
  10.10.0.6 (Strict IPv4 Prefix, 8 bytes, /32)
 RRO:
   10.10.7.1/32, Flags:0x0 (No Local Protection)
   10.10.4.1/32, Flags:0x9 (Local Prot Avail/to NNHOP) !Available to NNHOP
   10.10.1.1/32, Flags:0x0 (No Local Protection)
 Traffic params - Rate: 10K bits/sec, Max. burst: 1K bytes
   Min Policed Unit: 0 bytes, Max Pkt Size 4294967295 bytes
 Fast-Reroute Backup info:
   Inbound  FRR: Not active
   Outbound FRR: No backup tunnel selected
 Path ID handle: 50000416.
 Incoming policy: Accepted. Policy source(s): MPLS/TE
 Status: Proxy-terminated
```

Step 3 show mpls traffic-eng fast-reroute database

Enter the **clear ip rsvp hello instance counters** command to verify the following:

- MPLS TE FRR Node Protection has been enabled.
- A certain type of LSP can use a backup tunnel.

The following command output displays the LSPs that are protected:

Example:

```
Router# show mpls traffic-eng fast-reroute database
Tunnel head end item frr information:
Protected tunnel              In-label Out intf/label     FRR intf/label   Status
Tunnel500                     Tun hd   AT2/0/0.100:Untagg Tu501:20         ready
Prefix item frr information:
Prefix             Tunnel     In-label Out intf/label     FRR intf/label   Status
10.0.0.8/32        Tu500      18       AT2/0/0.100:Pop ta Tu501:20          ready
10.0.8.8/32        Tu500      19       AT2/0/0.100:Untagg Tu501:20          ready
10.8.9.0/24        Tu500      22       AT2/0/0.100:Untagg Tu501:20          ready
LSP midpoint item frr information:
LSP identifier                In-label Out intf/label   FRR intf/label   Status
```

If LDP is not enabled, separate prefix items are not shown because all prefixes then use a single rewrite. To confirm that a particular IP prefix is FRR protected, even though it is not shown in this display, enter it within the **show mpls forwarding-table** *ip-address* **detail** command. The final line of the display will tell whether that prefix is protected:

Example:

```
Router# show mpls forwarding-table 10.0.0.11 32 detail

Local    Outgoing    Prefix         Bytes tag    Outgoing     Next Hop
tag      tag or VC   or Tunnel Id   switched     interface
Tun hd   Untagged    10.0.0.11/32   48
      point2point
          MAC/Encaps=4/8, MTU=1520, Tag Stack{22}
          48D18847 00016000
          No output feature configured
          Fast Reroute Protection via (Tu0, outgoing label 12304)
```

The following command output displays the LSPs that are protected when the FRR *backup* tunnel is over an ATM interface:

Example:

```
Router# show mpls traffic-eng fast-reroute database

Tunnel head end item frr information:
Protected tunnel In-label   Out intf/label FRR intf/label Status
Tunnel500 Tun hd PO0/2/0:Untagged Tu501:20 ready
Prefix item frr information:
Prefix Tunnel In-label Out intf/label FRR intf/label Status
10.0.0.8/32 Tu500 18 PO0/2/0:Pop tag  Tu501:20 ready
10.0.8.8/32 Tu500 19 PO0/2/0:Untagged Tu501:20 ready
10.8.9.0/24 Tu500 22 PO0/2/0:Untagged Tu501:20 ready
LSP midpoint item frr information:
LSP identifier In-label Out intf/label FRR intf/label Status
```

Step 4 **show mpls traffic-eng tunnels backup**

The following conditions must exist for backup tunnels to be operational:

- **LSP is reroutable** --At the headend of the LSP, enter the **show run int tunnel** *tunnel-number* command. The output should include the **tunnel mpls traffic-eng fast-reroute** command. If it does not, enter this command for the tunnel.

On the router where the backup tunnels originate, enter the **show mpls traffic-eng tunnels backup**command. Following is sample command output:

Example:

```
Router# show mpls traffic-eng tunnels backup
Router_t578
  LSP Head, Tunnel578, Admin: up, Oper: up
  Src 10.55.55.55, Dest 10.88.88.88, Instance 1
  Fast Reroute Backup Provided:
    Protected i/fs:
    Protected lsps: 1
    Backup BW: any pool unlimited; inuse: 100 kbps
Router_t5710
  LSP Head, Tunnel5710, Admin: admin-down, Oper: down
  Src 10.55.55.55, Dest 10.7.7.7, Instance 0
  Fast Reroute Backup Provided:
    Protected i/fs:
    Protected lsps: 0
    Backup BW: any pool unlimited; inuse: 0 kbps
Router_t5711
  LSP Head, Tunnel5711, Admin: up, Oper: up
  Src 10.55.55.55, Dest 10.7.7.7, Instance 1
  Fast Reroute Backup Provided:
    Protected i/fs:
    Protected lsps: 2
    Backup BW: any pool unlimited; inuse: 6010 kbps
```

The command output will allow you to verify the following:

- Backup tunnel exists--Verify that there is a backup tunnel that terminates at this LSP's NHOP or NNHOP. Look for the LSP's NHOP or NNHOP in the Dest field.
- Backup tunnel is up--To verify that the backup tunnel is up, look for "Up" in the State field.
- Backup tunnel is associated with LSP's I/F--Verify that the interface for the LSP is allowed to use this backup tunnel. Look for the LSP's output interface in the "protects" field list.
- Backup tunnel has sufficient bandwidth--If you restricted the amount of bandwidth a backup tunnel can hold, verify that the backup tunnel has sufficient bandwidth to hold the LSPs that would use this backup tunnel if there is a failure. The bandwidth of an LSP is defined by the line **tunnel mpls traffic-eng bandwidth** at the headend of the LSP. To determine the available bandwidth on a backup tunnel, look at the "cfg" and "inuse" fields. If there is insufficient backup bandwidth to accommodate the LSPs that would use this backup tunnel in the event of a failure, create an additional backup tunnel or increase the backup bandwidth of the existing tunnel by using the **tunnel mpls traffic-eng bandwidth** command.

Note To determine how much bandwidth is sufficient, offline capacity planning may be required.

- Backup tunnel has appropriate bandwidth type--If you restricted the type of LSPs (subpool or global pool) that can use this backup tunnel, verify that the LSP is the appropriate type for the backup tunnel. The type of the LSP is defined by the line **tunnel mpls traffic-eng bandwidth** at the headend of this LSP. If this line contains the word "subpool", then it uses subpool bandwidth; otherwise, it uses global pool bandwidth. Verify that the type matches the type the backup tunnel can hold by looking in the output of the above command.

If none of the above actions works, enable debug by entering the **debug ip rsvp fast-reroute** command and the **debug mpls traffic-eng fast-reroute** command on the router that is the headend of the backup tunnel. Then do the following:

a. Enter the **shutdown** command for the primary tunnel.

b. Enter the **no shutdown** command for the primary tunnel.

c. View the debug output.

Step 5 **show mpls traffic-eng fast-reroute database**

Enter the **clear ip rsvp hello instance counters** command to verify the following:

- MPLS TE FRR Node Protection has been enabled.
- A certain type of LSP can use a backup tunnel.

The following command output displays the LSPs that are protected:

Example:

```
Router# show mpls traffic-eng fast-reroute database
Tunnel head end item frr information:
Protected Tunnel   In-label   intf/label         FRR intf/label      Status
Tunnel10           Tun         :Untagged  Tu0:12304           ready
Prefix item frr information:
Prefix        Tunnel  In-label    Out intf/label     FRR intf/label Status
10.0.0.11/32  Tu110   Tun hd       :Untagged  Tu0:12304      ready
LSP midpoint frr information:
LSP identifier       In-label Out intf/label     FRR intf/label   Status
10.0.0.12 1 [459]    16        :17        Tu2000:19        ready
```

Note If LDP is not enabled, separate prefix items are not shown because all prefixes then use a single rewrite. To confirm that a particular IP prefix is FRR protected, even though it is not shown in this display, enter it within the **show mpls forwarding-table** *ip-address* **detail** command. The final line of the display will tell whether that prefix is protected:

Example:

```
Router# show mpls forwarding-table 10.0.0.11 32 detail

Local    Outgoing    Prefix         Bytes tag    Outgoing       Next Hop
tag      tag or VC   or Tunnel Id   switched     interface
Tun hd   Untagged    10.0.0.11/32   48                   point2point
          MAC/Encaps=4/8, MTU=1520, Tag Stack{22}
          48D18847 00016000
          No output feature configured
          Fast Reroute Protection via (Tu0, outgoing label 12304)
```

Step 6 **show ip rsvp reservation**

Following is sample output from the **show ip rsvp reservation** command entered at the headend of a primary LSP. Entering the command at the head-end of the primary LSP shows, among other things, the status of FRR (that is, local protection) at each hop this LSP traverses. The per-hop information is collected in the Record Route Object (RRO) that travels with the Resv message from the tail to the head.

Example:

```
Router# show ip rsvp reservation detail
Reservation:
```

```
Tun Dest:   10.1.1.1  Tun ID: 1  Ext Tun ID: 10.1.1.1
Tun Sender: 10.1.1.1  LSP ID: 104
Next Hop: 10.1.1.2 on
Label: 18 (outgoing)
Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
Average Bitrate is 0 bits/sec, Maximum Burst is 1K bytes
Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
RRO:
  10.1.1.1/32, Flags:0x1 (Local Prot Avail/to NHOP)
    Label subobject: Flags 0x1, C-Type 1, Label 18
  10.1.1.1/32, Flags:0x0 (Local Prot Avail/In Use/Has BW/to NHOP)
    Label subobject: Flags 0x1, C-Type 1, Label 16
  10.1.1.2/32, Flags:0x0 (No Local Protection)
    Label subobject: Flags 0x1, C-Type 1, Label 0
Resv ID handle: CD000404.
Policy: Accepted. Policy source(s): MPLS/TE
```

Notice the following about the primary LSP:

- It has protection that uses a NHOP backup tunnel at its first hop.
- It has protection and is actively using an NHOP backup tunnel at its second hop.
- It has no local protection at its third hop.

The RRO display shows the following information for each hop:

- Whether local protection is available (that is, whether the LSP has selected a backup tunnel)
- Whether local protection is in use (that is, whether the LSP is currently using its selected backup tunnel)
- Whether the selected backup tunnel is an NHOP or NNHOP backup tunnel
- Whether the backup tunnel used at this hop provides bandwidth protection

Troubleshooting Tips

This section describes the following:

LSPs Do Not Become Active; They Remain Ready

At a PLR, LSPs transition from Ready to Active if one of the following events occurs:

- Primary interface goes down--If the primary interface (LSP's outbound interface) goes down and the LSP is ready to use a backup tunnel, the LSP will transition to the active state causing its data to flow over the backup tunnel. On some platforms and interface types (for example, GSR POS interfaces), fast interface-down logic has been added to detect this event very quickly. On other platforms where this logic does not exist, detection time is slower. On such platforms, it may be desirable to enable RSVP Hello (see the next bulleted item, "Hellos detect next hop is down").
- Hellos detect next hop is down--If Hellos are enabled on the primary interface (LSP's outbound interface), and the LSP's next hop is no longer reachable, the next hop is declared down. This event will cause the LSP to begin actively using its backup tunnel. Notice that a next hop will be declared down even if the primary interface does not go down. For example, if the next hop stops responding due to a reboot or software/hardware problem, Hellos will trigger the LSPs using this next hop to switch to their backup tunnels. Hellos can also help trigger FRR on interfaces such as Gigabit Ethernet where the interface remains up but is unusable (due to lack of link-layer liveness detection mechanisms).

Primary Tunnel Does Not Select Backup Tunnel That Is Up

If a backup tunnel is up, but it is not selected as a backup tunnel by the primary tunnel (LSP), enter the following commands for the backup tunnel:

- **shutdown**
- **no shutdown**

Note If you change the status of a backup tunnel, the backup tunnel selection algorithm is rerun for the backup tunnel. LSPs that have currently selected (that is, are ready to use) that backup tunnel will be disassociated from it, and then reassociated with that backup tunnel or another backup tunnel. This is generally harmless and usually results in mapping the same LSPs to that backup tunnel. However, if any LSPs are actively using that backup tunnel, shutting down the backup tunnel will tear down those LSPs.

Enhanced RSVP Commands

The following RSVP commands have been enhanced to display information that can be helpful when examining FRR state or when troubleshooting FRR:

- **show ip rsvp request** --Displays upstream reservation state (that is, information related to the Resv messages that this node will send upstream).
- **show ip rsvp reservation** --Displays information about Resv messages received.
- **show ip rsvp sender** --Displays information about Path messages being received.

These commands show control plane state; they do not show data state. That is, they show information about RSVP messages (Path and Resv) used to signal LSPs. For information about the data packets being forwarded along LSPs, use the **show mpls forwarding** command.

RSVP Hello

The RSVP Hello feature enables RSVP nodes to detect when a neighboring node is not reachable. Use this feature when notification of link-layer failures is not available and unnumbered links are not used, or when the failure detection mechanisms provided by the link layer are not sufficient for timely node failure detection. Hello must be configured both globally on the router and on the specific interface to be operational.

Hello Instances Have Not Been Created

If Hello instances have not been created, do the following:

- Determine if RSVP Hello has been enabled globally on the router. Enter the **ip rsvp signalling hello**(configuration) command.
- Determine if RSVP Hello has been enabled on an interface that the LSPs traverse. Enter the **ip rsvp signalling hello**(interface) command.
- Verify that at least one LSP has a backup tunnel by viewing the output of the **show ip rsvp sender**command. A value of “Ready” indicates that a backup tunnel has been selected.

No entry at index (error may self-correct, RRO may not yet have propagated from downstream node of interest)" Error Message Is Printed at the Point of Local Repair

FRR relies on a Record Route Object (RRO) in Resv messages arriving from downstream. Routers receiving Path messages with the SESSION_ATTRIBUTE bit indicating that the LSP is fast-reroutable should include an RRO in the corresponding Resv messages.

If an LSP is configured for FRR, but the Resv arriving from a downstream router contains an incomplete RRO, the "No entry at index (error may self-correct, RRO may not yet have propagated from downstream node of interest)" message is printed. An incomplete RRO is one in which the NHOP or the NNHOP did not include an entry in the RRO.

This error typically means that backup tunnels to the NHOP or the NNHOP cannot be selected for this LSP because there is insufficient information about the NHOP or NNHOP due to the lack of an RRO entry.

Occasionally there are valid circumstances in which this situation occurs temporarily and the problem is self-corrected. If subsequent Resv messages arrive with a complete RRO, ignore the error message.

To determine whether the error has been corrected, view the RRO in Resv messages by entering the **clear ip rsvp hello instance counters** command. Use an output filter keyword to view only the LSP of interest.

Couldn't get rsbs (error may self-correct when Resv arrives)" Error Message Is Printed at the Point of Local Repair

The PLR cannot select a backup tunnel for an LSP until a Resv message has arrived from downstream.

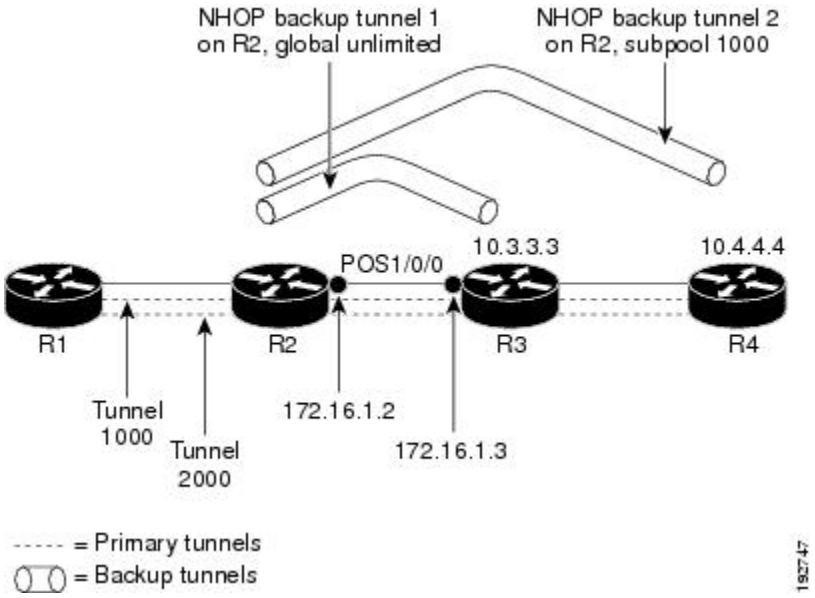
When this error occurs, it typically means that something is truly wrong. For example, no reservation exists for this LSP. You can troubleshoot this problem by using the **debug ip rsvp reservation** command to enable debug.

Occasionally there are valid circumstances in which this error message occurs and there is no need for concern. One such circumstance is when an LSP experiences a change before any Resv message has arrived from downstream. Changes can cause a PLR to try to select a backup tunnel for an LSP, and the selection will fail (causing this error message) if no Resv message has arrived for this LSP.

Configuration Examples for Link and Node Protection with RSVP Hellos Support

The examples relate to the illustration shown in the figure below.

Figure 169: Backup Tunnels



Enabling Fast Reroute for All Tunnels Example

On router R1, enter interface configuration mode for each tunnel to be protected (Tunnel 1000 and Tunnel 2000). Enable these tunnels to use a backup tunnel in case of a link or node failure along their paths.

Tunnel 1000 will use 10 units of bandwidth from the subpool.

Tunnel 2000 will use 5 units of bandwidth from the global pool. The "bandwidth protection desired" bit and the "node protection desired bit" have been set by specifying **bw-prot** and **node-prot**, respectively, in the **tunnel mpls traffic-eng fast-reroute** command.

```
Router(config)# interface Tunnel1000
Router(config-if)# tunnel mpls traffic-eng fast-reroute
Router(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 10
Router(config-if)# exit
Router(config)# interface Tunnel2000
Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-prot node-prot
Router(config-if)# tunnel mpls traffic-eng bandwidth 5
Router(config-if)# end
```

Creating an NHOP Backup Tunnel Example

On router R2, create an NHOP backup tunnel to R3. This backup tunnel should avoid using the link 10.1.1.2.

```
Router(config)# ip explicit-path name avoid-protected-link
Router(cfg-ip-expl-path)# exclude-address 10.1.1.2

Explicit Path name avoid-protected-link:
____1: exclude-address 10.1.1.2
Router(cfg-ip_expl-path)# end

Router(config)# interface Tunnel1
```

```
Router(config-if)# ip unnumbered loopback0

Router(config-if)# tunnel destination 10.3.3.3
Router(config-if)# tunnel mode mpls traffic-eng0

Router(config-if)# tunnel mpls traffic-eng path-option explicit avoid-protected-link
```

Creating an NNHOP Backup Tunnel Example

On router R2, create an NNHOP backup tunnel to R4. This backup tunnel should avoid R3.

```
Router(config)# ip explicit-path name avoid-protected-node

Router(cfg-ip-expl-path)# exclude-address 10.3.3.3

Explicit Path name avoid-protected-node:
____1: exclude-address 10.3.3.3
Router(cfg-ip_expl-path)# end

Router(config)# interface Tunnel2

Router(config-if)# ip unnumbered loopback0

Router(config-if)# tunnel destination 10.4.4.4

Router(config-if)# tunnel mode mpls traffic-eng0

Router(config-if)# tunnel mpls traffic-eng path-option explicit avoid-protected-node
```

Assigning Backup Tunnels to a Protected Interface Example

On router R2, associate both backup tunnels with interface POS1/0/0.

```
Router(config)# interface POS1/0/0

Router(config-if)# mpls traffic-eng backup-path tunnel1

Router(config-if)# mpls traffic-eng backup-path tunnel2
```

Associating Backup Bandwidth and Pool Type with Backup Tunnels Example

Backup tunnel 1 is to be used only by LSPs that take their bandwidth from the global pool. It does not provide bandwidth protection. Backup tunnel 2 is to be used only by LSPs that take their bandwidth from the subpool. Backup tunnel 2 provides bandwidth protection for up to 1000 units.

```
Router(config)# interface Tunnel1

Router(config-if)# tunnel mpls traffic-eng backup-bw global-pool Unlimited

Router(config)# interface Tunnel2

Router(config-if)# tunnel mpls traffic-eng backup-bw sub-pool 1000
```

Configuring Backup Bandwidth Protection Example

In the following example, backup bandwidth protection is configured.

Note This global configuration is required only to change the backup protection preemption algorithm from minimize the number of LSPs that are demoted to minimize the amount of bandwidth that is wasted.

```
Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect
Router(config)# mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw
```

Configuring an Interface for Fast Link and Node Failure Detection Example

In the following example, pos ais-shut is configured:

```
Router(config)# interface pos0/0/0

Router(config-if)# pos ais-shut
```

In the following example, report lrdi is configured on OS interfaces:

```
Router(config)# interface pos0/0/0
Router(config-if)# pos report lrdi
```

Configuring an Interface for Fast Tunnel Interface Down Example

In the following example, tunnel 1000 goes down as soon as the headend router detects that the LSP is down:

```
Router(config)# interface tunnel 1000

Router(config-if)# tunnel mpls traffic-eng interface down delay 0
```

Configuring RSVP Hello and POS Signals Example

Hello must be configured both globally on the router and on the specific interface on which you need FRR protection. To configure Hello, use the following configuration commands:

- **ip rsvp signalling hello** (configuration)--Enables Hello globally on the router.
- **ip rsvp signalling hello** (interface)--Enables Hello on an interface where you need FRR protection.

The following configuration commands are optional:

- **ip rsvp signalling hello dscp** --Sets the DSCP value that is in the IP header of the Hello message.
- **ip rsvp signalling hello refresh misses** --Specifies how many acknowledgments a node can miss in a row before the node considers that communication with its neighbor is down.
- **ip rsvp signalling hello refresh interval** --Configures the Hello request interval.

- **ip rsvp signalling hello statistics** --Enables Hello statistics on the router.

To configure POS signaling for detecting FRR failures, enter **pos report all** or enter the following commands to request individual reports:

- **pos ais-shut**
- **pos report rdool**
- **pos report lais**
- **pos report lrdi**
- **pos report pais**
- **pos report prdi**
- **pos report sd-ber**

Additional References

The following sections provide references related to the MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) feature.

Related Documents

Related Topic	Document Title
IS-IS	• *Cisco IOS IP Routing Protocols Command Reference* • Configuring a Basic IS-IS Network
MPLS traffic engineering commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
OSPF	• *Cisco IOS IP Routing Protocols Command Reference* • Configuring OSPF
RSVP commands	• *Cisco IOS Multiprotocol Label Switching Command Reference* • *Cisco IOS Quality of Service Solutions Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 4090	Fast Reroute Extensions to RSVP-TE for LSP Tunnels

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Link and Node Protection with RSVP Hellos Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 181: Feature Information for MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection

Feature Name	Releases	Feature Information
MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection	Cisco IOS XE Release 2.3	The MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) feature provides the following Fast Reroute (FRR) capabilities: • A backup tunnel terminates at the next-next hop router to protect both the downstream link and node to protect link and node failures. There is no limit (except memory limitations) to the number of backup tunnels that can protect a given interface. A backup tunnel is scalable because it can protect multiple LSPs and multiple interfaces. • Backup bandwidth protection allows a priority to be assigned to backup tunnels for LSPs carrying certain kinds of data (such as voice). • Fast Tunnel Interface Down detection, which forces a “generic” interface tunnel (not specifically a Fast Reroute tunnel) to become disabled immediately if the headend router detects a failed link on an LSP. • Resource Reservation Protocol (RSVP) Hellos, which are used to accelerate the detection of node failures. In Cisco IOS Release XE 2.3, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced or modified: **tunnel mpls traffic-eng interface down delay**.

Glossary

backup bandwidth --The usage of NHOP and NNHOP backup tunnels to provide bandwidth protection for rerouted LSPs.

backup tunnel --An MPLS TE tunnel used to protect other (primary) tunnels’ traffic when a link or node failure occurs.

bandwidth --The available traffic capacity of a link.

Cisco Express Forwarding --A means for accelerating the forwarding of packets within a router, by storing route lookup.

enterprise network --A large and diverse network connecting most major points in a company or other organization.

Fast Reroute --Procedures that enable temporary routing around a failed link or node while a new LSP is being established at the head end.

Gigabit Ethernet --Standard for a high-speed Ethernet, approved by the IEEE (Institute of Electrical and Electronics Engineers) 802.3z standards committee in 1996.

global pool --The total bandwidth allocated to an MPLS Traffic Engineering link or node.

headend --The router that originates and maintains a given LSP. This is the first router in the LSP's path.

hop --Passage of a data packet between two network nodes (for example, between two routers).

instance --A Hello instance implements the RSVP Hello extensions for a given router interface address and remote IP address. Active Hello instances periodically send Hello Request messages, expecting Hello ACK messages in response. If the expected Ack message is not received, the active Hello instance declares that the neighbor (remote IP address) is unreachable (that is, it is lost). This can cause LSPs crossing this neighbor to be fast rerouted.

interface --A network connection.

Intermediate System-to-Intermediate System --IS-IS. Link-state hierarchical routing protocol that calls for intermediate system (IS) routers to exchange routing information based on a single metric to determine network topology.

link --A point-to-point connection between adjacent nodes. There can be more than one link between adjacent nodes. A network communications channel consisting of a circuit or transmission path and all related equipment between a sender and a receiver. Sometimes referred to as a line or a transmission link.

limited backup bandwidth --Backup tunnels that provide bandwidth protection.

load balancing --A configuration technique that shifts traffic to an alternative link if a certain threshold is exceeded on the primary link. Load balancing is similar to redundancy in that if an event causes traffic to shift directions, alternative equipment must be present in the configuration. In load balancing, the alternative equipment is not necessarily redundant equipment that only operates in the event of a failure.

LSP --label switched path. A configured connection between two routers, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

merge point --The backup tunnel's tail.

MPLS --Multiprotocol Label Switching. Packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

MPLS global label allocation --There is one label space for all interfaces in the router. For example, label 100 coming in one interface is treated the same as label 100 coming in a different interface.

NHOP --next hop. The next downstream node along an LSP's path.

NHOP backup tunnel --next-hop backup tunnel. Backup tunnel terminating at the LSP's next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link, and is used to protect primary LSPs that were using this link before the failure.

NNHOP --next-next hop. The node after the next downstream node along an LSP's path.

NNHOP backup tunnel --next-next-hop backup tunnel. Backup tunnel terminating at the LSP's next-next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link or node, and is used to protect primary LSPs that were using this link or node before the failure.

node --Endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network. Computers on a network, or any endpoint or a junction common to two or more lines in a network. Nodes can be processors, controllers, or workstations.

OSPF --Open Shortest Path First. A link-state hierarchical Interior Gateway Protocol routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

primary LSP --The last LSP originally signaled over the protected interface before the failure. The LSP before the failure.

primary tunnel --Tunnel whose LSP may be fast rerouted if there is a failure. Backup tunnels cannot be primary tunnels.

promotion --Conditions, such as a new backup tunnel comes up, cause a reevaluation of a backup tunnel that was chosen for an LSP. If the reevaluation is successful, it is called a promotion.

protected interface --An interface that has one or more backup tunnels associated with it.

redundancy --The duplication of devices, services, or connections so that, in the event of a failure, the redundant devices, services, or connections can perform the work of those that failed.

RSVP --Resource Reservation Protocol. An IETF protocol used for signaling requests (setting up reservations) for Internet services by a customer before that customer is permitted to transmit data over that portion of the network.

scalability --An indicator showing how quickly some measure of resource usage increases as a network gets larger.

state --Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

subpool --The more restrictive bandwidth in an MPLS Traffic Engineering link or node. The subpool is a portion of the link or node's overall global pool bandwidth.

tailend --The router upon which an LSP is terminated. This is the last router in the LSP's path.

topology --The physical arrangement of network nodes and media within an enterprise networking structure.

tunnel --Secure communications path between two peers, such as two routers.

unlimited backup bandwidth --Backup tunnels that provide no bandwidth (best-effort) protection (that is, they provide best-effort protection).

CHAPTER 102

MPLS Traffic Engineering-Autotunnel Primary and Backup

The MPLS Traffic Engineering-Autotunnel Primary and Backup feature enables a router to dynamically build backup tunnels and to dynamically create one-hop primary tunnels on all interfaces that have been configured with Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels.

A router with primary one-hop autotunnels and backup autotunnels can be configured with stateful switchover (SSO) redundancy.

Prerequisites for MPLS Traffic Engineering-Autotunnel Primary and Backup

- Configure TE on the routers.

Restrictions for MPLS Traffic Engineering-Autotunnel Primary and Backup

- You cannot configure a static route to route traffic over TE autotunnels. For autotunnels, you should use only the autoroute for tunnel selection.

Information About MPLS Traffic Engineering-Autotunnel Primary and Backup

Overview of MPLS Traffic Engineering-Autotunnel Primary and Backup

The MPLS Traffic Engineering-Autotunnel Primary and Backup feature has the following features:

- Backup autotunnel-Enables a router to dynamically build backup tunnels.
- Primary one-hop autotunnel-Enables a router to dynamically create one-hop primary tunnels on all interfaces that have been configured with MPLS TE tunnels.

If no backup tunnels exist, the following types of backup tunnels are created:

- Next hop (NHOP)
- Next-next hop (NNHOP)

Benefits of MPLS Traffic Engineering-Autotunnel Primary and Backup Feature

- Backup tunnels are built automatically, eliminating the need for users to preconfigure each backup tunnel and then assign the backup tunnel to the protected interface.
- The dynamic creation of one-hop primary tunnels eliminates the need to configure an MPLS TE tunnel with the Fast Reroute (FRR) option for the tunnel to be protected.
- Protection is expanded; FRR does not protect IP traffic that is not using the TE tunnel or Label Distribution Protocol (LDP) labels that are not using the TE tunnel.

MPLS Traffic Engineering

MPLS is an Internet Engineering Task Force (IETF)-specified framework that provides for the efficient designation, routing, forwarding, and switching of traffic flows through the network.

TE is the process of adjusting bandwidth allocations to ensure that enough bandwidth is left for high-priority traffic.

In MPLS TE, the upstream router creates a network tunnel for a particular traffic stream, then sets the bandwidth available for that tunnel.

MPLS Traffic Engineering Backup Autotunnels

MPLS backup autotunnels protect fast reroutable TE label switched paths (LSPs). Without MPLS backup autotunnels to protect a LSP you had to do the following:

- Preconfigure each backup tunnel.
- Assign the backup tunnels to the protected interfaces.

An LSP requests backup protection from Resource Reservation Protocol (RSVP) FRR in the following situations:

- Receipt of the first RSVP Resv message
- Receipt of an RSVP path message with the protection attribute after the LSP has been established without the protection attribute
- Detection that a Record Route Object (RRO) changed

If there was no backup tunnel protecting the interface used by the LSP, the LSP remained unprotected.

Backup autotunnels enable a router to dynamically build backup tunnels when they are needed. This prevents you from having to build MPLS TE tunnels statically.

Backup tunnels may not be available for the following reasons:

- Static backup tunnels are not configured.
- Static backup tunnels are configured, but cannot protect the LSP. The backup tunnel may not have enough available bandwidth, the tunnel may protect a different pool, or the tunnel may be down.

If a backup tunnel is not available, the following two backup tunnels are created dynamically:

- NHOP--Protects against link failure
- NNHOP--Protects against node failure

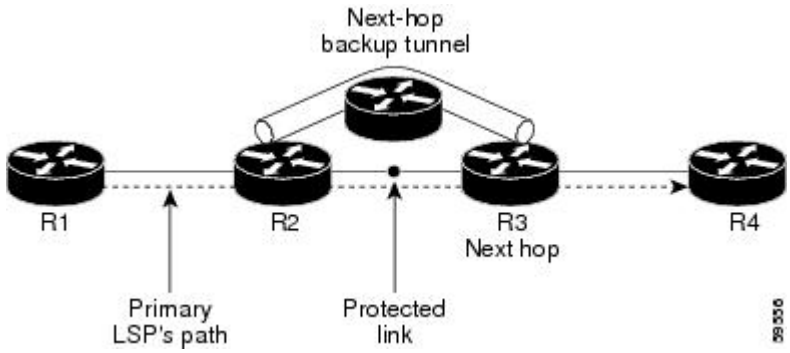
Note At the penultimate hop, only an NHOP backup tunnel is created.

Note If two LSPs share the same output interface and NHOP, three (not four) backup tunnels are created. They share an NHOP backup tunnel.

Link Protection

Backup tunnels that bypass only a single link of the LSP's path provide Link Protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure. The figure below illustrates an NHOP backup tunnel.

Figure 170: NHOP Backup Tunnel

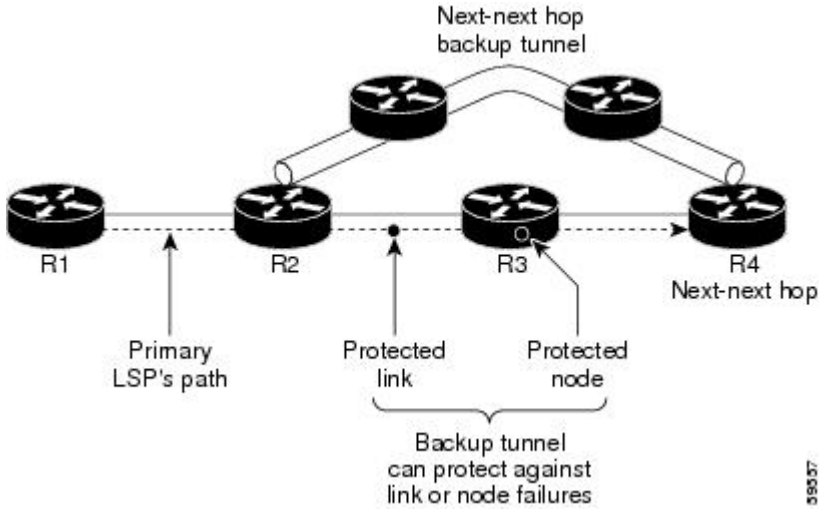


Node Protection

Backup tunnels that bypass next-hop nodes along LSP paths are called NNHOP backup tunnels because they terminate at the node following the next-hop node of the LSPs, thereby bypassing the next-hop node. They protect LSPs by enabling the node upstream of a link or node failure to reroute the LSPs and their traffic around the failure to the next-hop node. NNHOP backup tunnels also provide protection from link failures because they bypass the failed link and the node.

The figure below illustrates an NNHOP backup tunnel.

Figure 171: Next-Next Hop Backup Tunnel



Explicit Paths

Explicit paths are used to create backup autotunnels as follows:

- NHOP excludes the protected link's IP address.
- NNHOP excludes the NHOP router ID.
- The explicit-path name is _auto-tunnel_tunnel*xxx* , where *xxx* matches the dynamically created backup tunnel ID.

- The interface used for the **ip unnumbered** command defaults to Loopback0. You can configure this to use a different interface.

Range for Backup Autotunnels

The tunnel range for backup autotunnels is configurable. By default, the last 100 TE tunnel IDs are used; that is 65,436 to 65,535. Autotunnels detect tunnel IDs that are being used. IDs are allocated starting with the lowest number.

For example, if you configure a tunnel range 1000 to 1100 and statically configured TE tunnels are in that range, routers do not use those IDs. If those static tunnels are removed, the MPLS TE dynamic tunnel software can use those IDs.

MPLS Traffic Engineering Primary Autotunnels

The MPLS Traffic Engineering-Autotunnel Primary and Backup feature enables a router to dynamically create one-hop primary tunnels on all interfaces that have been configured with MPLS traffic. The tunnels are created with zero bandwidth. The constraint-based shortest path first (CSPF) is the same as the shortest path first (SPF) when there is zero bandwidth, so the router's choice of the autorouted one-hop primary tunnel is the same as if there were no tunnel. Because it is a one-hop tunnel, the encapsulation is tag-implicit (that is, there is no tag header).

Explicit Paths

Explicit paths are used to create autotunnels as follows:

- The explicit path is dynamically created.
- The explicit path includes the IP address for the interface connected to the next hop.
- The explicit-path name is _auto-tunnel_tunnel*xxx* , where *xxx* matches the dynamically created one-hop tunnel ID.
- Interfaces used for the **ip unnumbered** command default to Loopback0. You can configure this to use a different interface.

Range for Autotunnels

The tunnel range is configurable. By default, the last 100 TE tunnel IDs are used; that is 65,436 to 65,535. Autotunnels detect tunnel IDs that are being used. IDs are allocated starting with the lowest number.

For example, if you configure a tunnel range 100 to 200 and statically configured TE tunnels are in that range, routers do not use those IDs. If those static tunnels are removed, the IDs become available for use by the MPLS TE dynamic tunnel software.

MPLS Traffic Engineering Label-Based Forwarding

Routers receive a packet, determine where it needs to go by examining some fields in the packet, and send it to the appropriate output device. A label is a short, fixed-length identifier that is used to forward packets. A label switching device normally replaces the label in a packet with a new value before forwarding the packet to the next hop. For this reason, the forwarding algorithm is called label swapping. A label switching device,

referred to as an LSR, runs standard IP control protocols (that is, routing protocols, RSVP, and so forth) to determine where to forward packets.

Benefits of MPLS Traffic Engineering Protection

The following sections describe the benefits of MPLS traffic engineering protection:

DeliveryofPacketsDuringaFailure

Backup tunnels that terminate at the NNHOP protect both the downstream link and node. This provides protection for link and node failures.

Multiple Backup Tunnels Protecting the Same Interface

In addition to being required for node protection, the autotunnel primary and backup feature provides the following benefits:

- Redundancy--If one backup tunnel is down, other backup tunnels protect LSPs.
- Increased backup capacity--If the protected interface is a high-capacity link and no single backup path exists with an equal capacity, multiple backup tunnels can protect that one high-capacity link. The LSPs using this link will fail over to different backup tunnels, allowing all of the LSPs to have adequate bandwidth protection during failure (rerouting). If bandwidth protection is not desired, the router spreads LSPs across all available backup tunnels (that is, there is load balancing across backup tunnels).

Scalability

A backup tunnel can protect multiple LSPs. Furthermore, a backup tunnel can protect multiple interfaces. This is called many-to-one (N:1) protection. N:1 protection has significant scalability advantages over one-to-one (1:1) protection, where a separate backup tunnel must be used for each LSP needing protection.

An example of N:1 protection is that when one backup tunnel protects 5000 LSPs, each router along the backup path maintains one additional tunnel.

An example of 1:1 protection is that when 5000 backup tunnels protect 5000 LSPs, each router along the backup path must maintain state for an additional 5000 tunnels.

RSVP Hello

RSVP Hello allows a router to detect when its neighbor has gone down but its interface to that neighbor is still operational. When Layer 2 link protocols are unable to detect that the neighbor is unreachable, Hellos provide the detection mechanism; this allows the router to switch LSPs onto its backup tunnels and avoid packet loss.

SSO Redundancy Overview

The SSO feature is an incremental step within an overall program to improve the availability of networks constructed with Cisco IOS routers.

SSO is particularly useful at the network edge. It provides protection for network edge devices with dual route processors (RPs) that represent a single point of failure in the network design, and where an outage might result in loss of service for customers.

In specific Cisco networking devices that support dual RPs, SSO takes advantage of RP redundancy to increase network availability. The feature establishes one of the RPs as the active processor while the other RP is designated as the standby processor, and then synchronizes critical state information between them. Following an initial synchronization between the two processors, SSO dynamically maintains RP state information between them.

A switchover from the active to the standby processor occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance.

Affinity and Link Attributes with Autotunnel Backup

In Cisco IOS Release 15.1(1)S and later releases, you can use affinity and link attributes with the MPLS TE Autotunnel Backup feature to include or exclude links when configuring dynamic backup paths.

For a link, you can configure up to 32 bits of attribute flags, as shown in the following example:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet0/0
Router(config-if)# mpls traffic-eng attribute-flags 0x22
```

The attribute flags are compared to the tunnel's affinity bits during selection of the path.

When you enable the auto-tunnel backup feature, you can optionally specify the affinity and mask, as shown in the following example. If you do not specify an affinity and mask, the default for affinity is 0 and for the mask it is 0xFFFF is used. To ignore link affinity, use affinity and mask of 0. See the **mpls traffic-eng auto-tunnel backup config affinity**command for more information.

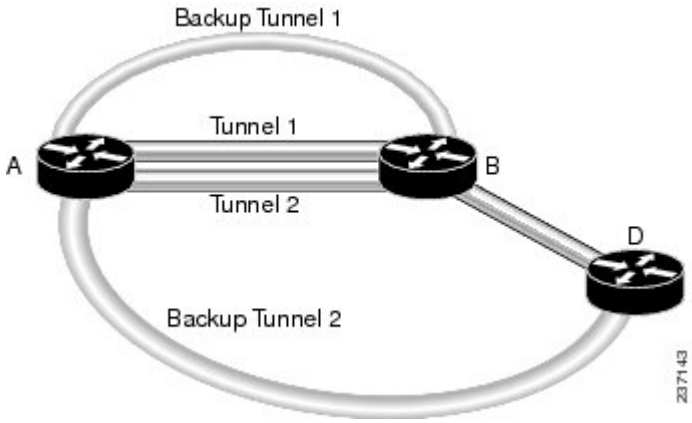
```
Router> enable
Router# configure terminal
Router(config)# mpls traffic-eng auto-tunnel backup

Router(config)# mpls traffic-eng auto-tunnel backup config affinity 0x13 mask 0x13
```

The affinity/mask configured by the **mpls traffic-eng auto-tunnel backup config affinity**command is used for all dynamically created backup tunnels. The attribute mask determines which link attributes are relevent. If a bit in the mask is 0, the attribute is irrelevant. If a bit in the mask is 1, the attribute value of a link and the configured affinity of the tunnel for that bit must match.

In the figure below, there are two primary tunnels. One tunnel travels from router A to router B. The other primary tunnel travels from router A to router B and then router D. All the the links are configured with attribute flags 0x22. Both tunnels require fast reroute protection. To automatically create backup tunnels, enable the autotunnel backup feature with the **mpls traffic-eng auto-tunnel backup** command. However, the dynamically created backup tunnels do not come up, because attribute flags are configured on the links. To enable the dynamically created backup tunnels, you must also issue the following command:

Router(config)# mpls traffic-eng auto-tunnel backup config affinity 0x22 mask 0x22

Figure 172: Specifying Link Attributes and Affinity with Autotunnel Backup



How to Configure MPLS Traffic Engineering Autotunnel Primary and Backup

Establishing MPLS Backup Autotunnels to Protect Fast Reroutable TE LSPs

To establish an MPLS backup autotunnel to protect fast reroutable TE LSPs, perform the following task.

Note Only Steps 1 through 3 are required. If you perform additional steps, you can perform them in any order after Step 3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel backup**
4. **mpls traffic-eng auto-tunnel backup nhop-only**
5. **mpls traffic-eng auto-tunnel backup tunnel-num** [**min** *num*] [**max** *num*]
6. **mpls traffic-eng auto-tunnel backup timers removal unused** *sec*
7. **mpls traffic-eng auto-tunnel backup config unnumbered-interface** *interface*
8. **mpls traffic-eng auto-tunnel backup config affinity** *affinity-value* **mask** *mask-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng auto-tunnel backup** **Example:** `Router(config)# mpls traffic-eng auto-tunnel backup`	Automatically builds NHOP and NNHOP backup tunnels.
Step 4	**mpls traffic-eng auto-tunnel backup nhop-only** **Example:** `Router(config)# mpls traffic-eng auto-tunnel backup nhop-only`	Enables the creation of dynamic NHOP backup tunnels.
Step 5	**mpls traffic-eng auto-tunnel backup tunnel-num** [**min** *num*] [**max** *num*] **Example:** `Router(config)# mpls traffic-eng auto-tunnel backup tunnel-num min 1000 max 1100`	Configures the range of tunnel interface numbers for backup autotunnels.
Step 6	**mpls traffic-eng auto-tunnel backup timers removal unused** *sec* **Example:** `Router(config)# mpls traffic-eng auto-tunnel backup timers removal unused 50`	Configures how frequently a timer will scan backup autotunnels and remove tunnels that are not being used. The value for **auto-tunnel backup timers removal unused** cannot be less than 10 mins.
Step 7	**mpls traffic-eng auto-tunnel backup config unnumbered-interface** *interface* **Example:** `Router(config)# mpls traffic-eng auto-tunnel backup config unnumbered-interface ethernet1/0`	Enables IP processing on the specified interface without an explicit address.
Step 8	**mpls traffic-eng auto-tunnel backup config affinity** *affinity-value* **mask** *mask-value*] **Example:** `Router(config)# mpls traffic-eng auto-tunnel backup config affinity 0x22 mask 0x22`	Specifies the affinity values and mask flags. The affinity determines the attribute of the link that the tunnel will use. That is, the attribute for which the tunnel has an affinity. The mask determines which link attribute the router should check. If a bit in the mask is 0, an attribute value of a link or that bit is irrelevant. If a bit in the mask is 1, the attribute values of a link and the required affinity of the tunnel for that bit must match.

Establishing MPLS One-Hop Tunnels to All Neighbors

To establish MPLS one-hop tunnels to all neighbors, perform the following task.

Note Only Steps 1 through 3 are required. If you perform additional steps, you can perform them in any order after Step 3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel primary onehop**
4. **mpls traffic-eng auto-tunnel primary tunnel-num** [**min** *num*] [**max***num*]
5. **mpls traffic-eng auto-tunnel primary timers removal rerouted** *sec*
6. **mpls traffic-eng auto-tunnel primary config unnumbered** *interface*
7. **mpls traffic-eng auto-tunnel primary config mpls ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng auto-tunnel primary onehop** **Example:** `Router(config)# mpls traffic-eng auto-tunnel primary onehop`	Automatically creates primary tunnels to all next hops.
Step 4	**mpls traffic-eng auto-tunnel primary tunnel-num** [**min** *num*] [**max***num*] **Example:** `Router(config)# mpls traffic-eng auto-tunnel primary tunnel-num min 2000 max 2100`	Configures the range of tunnel interface numbers for primary autotunnels.
Step 5	**mpls traffic-eng auto-tunnel primary timers removal rerouted** *sec* **Example:**	Configures how many seconds after a failure primary autotunnels will be removed.

	Command or Action	Purpose
	`Router(config)# mpls traffic-eng auto-tunnel primary timers removal rerouted 400`	
Step 6	**mpls traffic-eng auto-tunnel primary config unnumbered** *interface* **Example:** `Router(config)# mpls traffic-eng auto-tunnel primary config unnumbered ethernet1/0`	Enables IP processing on the specified interface without an explicit address.
Step 7	**mpls traffic-eng auto-tunnel primary config mpls ip** **Example:** `Router(config)# mpls traffic-eng auto-tunnel primary config mpls ip`	Enables LDP on primary autotunnels.

Configuration Examples for MPLS Traffic Engineering-Autotunnel Primary and Backup

Establishing MPLS Backup Autotunnels to Protect Fast Reroutable TE LSPs Example

Note This example does not include the **mpls traffic-eng auto-tunnel backup nhop-only** command because autotunneling would not be able to create any backup tunnels.

To determine if there are any backup tunnels, enter the **show ip rsvp fast-reroute** command. This example shows that there is a static configured primary tunnel and no backup tunnels.

```
Router(config)# show ip rsvp fast-reroute
Primary     Protect   BW         Backup
Tunnel      I/F       BPS:Type   Tunnel:Label  State  Level   Type
------      -------   -------    ------------  ------ -----   ----
R3-PRP_t0   PO3/1     0:G        None          None   None
```

The following command causes autotunnels to automatically configure NHOP and NNHOP backup tunnels:

```
Router(config)# mpls traffic-eng auto-tunnel backup
```

As illustrated in the **show ip interface brief** command output, autotunneling created two backup tunnels that have tunnel IDs 65436 and 65437:

```
Router# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol
```

```
POS2/0                 10.0.0.14       YES NVRAM  down                    down
POS2/1                 10.0.0.49       YES NVRAM  up                      up
POS2/2                 10.0.0.45       YES NVRAM  up                      up
POS2/3                 10.0.0.57       YES NVRAM  administratively down   down
POS3/0                 10.0.0.18       YES NVRAM  down                    down
POS3/1                 10.0.0.33       YES NVRAM  up                      up
POS3/2                 unassigned      YES NVRAM  administratively down   down
POS3/3                 unassigned      YES NVRAM  administratively down   down
GigabitEthernet4/0     10.0.0.37       YES NVRAM  up                      up
GigabitEthernet4/1     unassigned      YES NVRAM  administratively down   down
GigabitEthernet4/2     unassigned      YES NVRAM  administratively down   down
Loopback0              10.0.3.1        YES NVRAM  up                      up
Tunnel0                10.0.3.1        YES unset  up                      up
Tunnel65436            10.0.3.1        YES unset  up                      up
Tunnel65437            10.0.3.1        YES unset  up                      up
Ethernet0              10.3.38.3       YES NVRAM  up                      up
Ethernet1              unassigned      YES NVRAM  administratively down   down
R3-PRP#
```

The following command prevents autotunneling from creating NNHOP backup tunnels:

```
Router# mpls traffic-eng auto-tunnel backup nhop-only
```

The “Type” field in the following **show ip rsvp fast-reroute** command shows that there is only an NHOP tunnel:

```
Router# show ip rsvp fast-reroute

Primary    Protect  BW Backup
Tunnel     I/F      BPS:Type  Tunnel:Label  State  Level   Type
------     -------  --------  ------------ ------ ------- ----
R3-PRP_t0  PO3/1    0:G       Tu65436:24    Ready  any-unl Nhop
```

The following command changes the minimum and maximum tunnel interface numbers to 1000 and 1100, respectively:

```
Router# mpls traffic-eng auto-tunnel backup tunnel-num min 1000 max 1100
```

You can verify the ID numbers and autotunnel backup range ID by entering the **show ip rsvp fast-reroute** and **show ip interface brief** commands. In this example, only one backup tunnel is protecting the primary tunnel:

```
Router# show ip rsvp fast-reroute

Primary                  Protect BW         Backup
Tunnel                   I/F     BPS:Type   Tunnel:Label  State  Level   Type
------                   ------- --------   ------------ ------ -----   ------
R3-PRP_t0                PO3/1   0:G        Tu1000:24     Ready  any-unl Nhop
Router# show ip interface brief
Interface            IP-Address     OK?  Method  Status                  Protocol
POS2/0               10.0.0.14      YES  NVRAM   down                    down
POS2/1               10.0.0.49      YES  NVRAM   up                      up
POS2/2               10.0.0.45      YES  NVRAM   up                      up
POS2/3               10.0.0.57      YES  NVRAM   administratively down   down
POS3/0               10.0.0.18      YES  NVRAM   down                    down
POS3/1               10.0.0.33      YES  NVRAM   up                      up
POS3/2               unassigned     YES  NVRAM   administratively down   down
POS3/3               unassigned     YES  NVRAM   administratively down   down
GigabitEthernet4/0   10.0.0.37      YES  NVRAM   up                      up
GigabitEthernet4/1   unassigned     YES  NVRAM   administratively down   down
GigabitEthernet4/2   unassigned     YES  NVRAM   administratively down   down
```

```
Loopback0              10.0.3.1       YES  NVRAM   up                      up
Tunnel0                10.0.3.1       YES  unset   up                      up
Tunnel65436            10.0.3.1       YES  unset   up                      up
Ethernet0              10.3.38.3      YES  NVRAM   up                      up
Ethernet1              unassigned     YES  NVRAM   administratively down   down
```

The default tunnel range for autotunnel backup tunnels is 65,436 through 65,535. The following **show ip rsvp fast-reroute** command changes the tunnel range IDs:

```
Router# show ip rsvp fast-reroute

Primary     Protect  BW         Backup
Tunnel       I/F     BPS:Type   Tunnel:Label   State   Level    Type
------      -------  --------   ------------- ------   -----    ------
R3-PRP_t0   PO3/1    0:G        Tu1001:0       Ready   any-unl  N-Nhop
```

The results are shown in the **show ip interface brief** command:

```
Router# show ip interface
Router# show ip interface brief

Interface              UP-Address   OK?  Method  Status                  Protocol
POS2/0                 10.0.0.14    YES  NVRAM   down                    down
POS2/1                 10.0.0.49    YES  NVRAM   up                      up
POS2/2                 10.0.0.45    YES  NVRAM   up                      up
POS2/3                 10.0.0.57    YES  NVRAM   up                      up
POS3/0                 10.0.0.18    YES  NVRAM   up                      up
POS3/1                 10.0.0.33    YES  NVRAM   up                      up
POS3/2                 unassigned   YES  NVRAM   administratively down  down
POS3/3                 unassigned   YES  NVRAM   administratively down  down
Loopback0              10.0.3.1     YES  NVRAM   up                      up
Tunnel0                10.0.3.1     YES  unset   up                      up
Tunnel1000             10.0.3.1     YES  unset   up                      up
Tunnel1001             10.0.3.1     YES  unset   up                      up
Ethernet0              10.3.38.3    YES  NVRAM   up                      up
Ethernet1              unassigned   YES  NVRAM   administratively down  down
```

The following **mpls traffic-eng auto-tunnel backup timers removal unused** command specifies that a timer will scan backup autotunnels every 50 seconds and the timer will remove tunnels that are not being used:

```
Router(config)# mpls traffic-eng auto-tunnel backup timers removal unused 50
```

The following **mpls traffic-eng auto-tunnel backup config unnumbered-interface** command enables IP processing on POS interface 3/1:

```
Router(config)# mpls traffic-eng auto-tunnel backup config unnumbered-interface POS3/1
```

To verify that IP processing is enabled on POS3/1, enter the **show interfaces tunnel** command:

```
Router# show interfaces tunnel 1001
Tunnel1001 is up, line protocol is up
  Hardware is Tunnel
  Interface is unnumbered.  Using address of POS3/1 (10.0.0.33)
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.0, destination 10.0.5.1
  Tunnel protocol/transport Label Switching, sequencing disabled
  Key disabled
  Checksumming of packets disabled
  Last input never, output never, output hang never
```

```
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

The following **mpls traffic-eng auto-tunnel backup config affinity**command specifies affinity and link attributes that help in the calculation of the dynamically created backup tunnel:

Router(config)# **mpls traffic-eng auto-tunnel backup config affinity 0x22 mask 0x22**

To display the affinity and link attributes assigned to a dynamically created backup tunnel, enter the s**how mpls traffic-eng auto-tunnel backup**command:

```
Router# show mpls traffic-eng auto-tunnel backup

State: Enabled
  Tunnel Count: 3 (up:2, down: 1)
  Tunnel ID Range: 65436-65535
  Create Nhop only: Yes
  SRLG: Not configured
  Delete unused tunnels after: 50 Seconds
  Config:
    Unnumbered i/f: Looback0
    Affinity: 0x22/0x22
```

Establishing MPLS One-Hop Tunnels to Neighbors Example

For autotunneling to automatically create primary tunnels to all next hops, you must enter the following command:

```
Router(config)# mpls traffic-eng auto-tunnel primary onehop
```

In this example there are four primary tunnels and no backup tunnels. To verify that configuration, enter the **show ip rsvp fast-reroute** command and the **show ip interface brief** command:

```
Router# show ip rsvp fast-reroute
Primary                   Protect BW         Backup
Tunnel                    I/F     BPS:Type   Tunnel:Label  State  Level   Type
------                    ------- --------   ------------- ------ -----   ------
R3-PRP_t65337             PO2/2   0:G        None          None   None
R3-PRP_t65338             PO3/1   0:G        None          None   None
R3-PRP_t65339             Gi4/0   0:G        None          None   None
R3-PRP_t65336             PO2/1   0:G        None          None   None
Router# show ip interface brief
Interface              IP-Address    OK?  Method Status                Protocol
POS2/0                 10.0.0.14     YES  NVRAM  down                  down
POS2/1                 10.0.0.49     YES  NVRAM  up                    up
POS2/2                 10.0.0.45     YES  NVRAM  up                    up
POS2/3                 10.0.0.57     YES  NVRAM  administratively down  down
POS3/0                 10.0.0.18     YES  NVRAM  down                  down
POS3/1                 10.0.0.33     YES  NVRAM  up                    up
POS3/2                 unassigned    YES  NVRAM  administratively down  down
POS3/3                 unassigned    YES  NVRAM  administratively down  down
```

```
GigabitEthernet4/0     10.0.0.37     YES  NVRAM  up                    up
GigabitEthernet4/1     unassigned    YES  NVRAM  administratively down  down
GigabitEthernet4/2     unassigned    YES  NVRAM  administratively down  down
Loopback0              10.0.3.1      YES  NVRAM  up                    up
Tunnel0                10.0.3.1      YES  unset  administratively down  down
Tunnel65336            10.0.3.1      YES  unset  up                    up
Tunnel65337            10.0.3.1      YES  unset  up                    up
Tunnel65338            10.0.3.1      YES  unset  up                    up
Tunnel65339            10.0.3.1      YES  unset  up                    up
Ethernet0              10.3.38.3     YES  NVRAM  up                    up
Ethernet1              unassigned    YES  NVRAM  administratively down  down
R3-PRP#
```

The default tunnel range for primary autotunnels is 65,336 through 65,435. The following **mpls traffic-eng auto-tunnel primary tunnel-num** command changes the range to 2000 through 2100:

```
Router(config)# mpls traffic-eng auto-tunnel primary tunnel-num min 2000 max 2100
```

The following sample output from the **show ip rsvp fast-reroute** command and the **show ip interface brief** command shows that the tunnel IDs are 2000, 2001, 2002, and 2003:

```
Router# show ip rsvp fast-reroute
Primary                   Protect BW         Backup
Tunnel                    I/F     BPS:Type   Tunnel:Label  State  Level   Type
------                    ------- --------   ------------- ------ -----   ------
R3-PRP_t2001              PO2/2   0:G        None          None   None
R3-PRP_t2002              PO3/1   0:G        None          None   None
R3-PRP_t2003              Gi4/0   0:G        None          None   None
R3-PRP_t2000              PO2/1   0:G        None          None   None
Router# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol
POS2/0                 10.0.0.14       YES NVRAM  down                  down
POS2/1                 10.0.0.49       YES NVRAM  up                    up
POS2/2                 10.0.0.45       YES NVRAM  up                    up
POS2/3                 10.0.0.57       YES NVRAM  administratively down down
POS3/0                 10.0.0.18       YES NVRAM  down                  down
POS3/1                 10.0.0.33       YES NVRAM  up                    up
POS3/2                 unassigned      YES NVRAM  administratively down down
POS3/3                 unassigned      YES NVRAM  administratively down down
GigabitEthernet4/0     10.0.0.37       YES NVRAM  up                    up
GigabitEthernet4/1     unassigned      YES NVRAM  administratively down down
GigabitEthernet4/2     unassigned      YES NVRAM  administratively down down
Loopback0              10.0.3.1        YES NVRAM  up                    up
Tunnel0                10.0.3.1        YES unset  administratively down down
Tunnel2000             10.0.3.1        YES unset  up                    up
Tunnel2001             10.0.3.1        YES unset  up                    up
Tunnel2002             10.0.3.1        YES unset  up                    up
Tunnel2003             10.0.3.1        YES unset  up                    up
Ethernet0              10.3.38.3 '     YES NVRAM  up                    up
Ethernet1              unassigned      YES NVRAM  administratively down down
```

The following **mpls traffic-eng auto-tunnel primary timers**command specifies that a timer will scan backup autotunnels every 50 seconds and remove tunnels that are not being used:

```
Router(config)# mpls traffic-eng auto-tunnel primary timers removal rerouted 50
```

The following **mpls traffic-eng auto-tunnel primary config unnumbered** command enables IP processing on POS interface 3/1:

```
Router(config)# mpls traffic-eng auto-tunnel primary config unnumbered POS3/1
```

To specify that autotunneling remove all primary autotunnels and re-create them, enter the following command:

```
Router(config)# clear mpls traffic-eng auto-tunnel primary
```

Additional References

The following sections provide references related to the MPLS Traffic Engineering-Autotunnel Primary and Backup feature.

Additional References

Related Topic	Document Title
Backup tunnels	MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection)
Link protection	MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection)
MPLS traffic engineering commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
SSO	*Cisco IOS High Availability Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering-Autotunnel Primary and Backup

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 182: Feature Information for MPLS Traffic Engineering-Autotunnel Primary and Backup

Feature Name	Releases	Feature Configuration Information
MPLS Traffic Engineering-Autotunnel Primary and Backup	12.0(27)S 12.2(33)SRA 12.2(33)SXH 12.4(20)T 12.2(33)SRE 15.1(1)S Cisco IOS XE Release 2.3	The MPLS Traffic Engineering-Autotunnel Primary and Backup feature enables a router to dynamically build backup tunnels and to dynamically create one-hop primary tunnels on all interfaces that have been configured with MPLS TE tunnels. In Cisco IOS Release 12.0(27)S, this feature was introduced. In Cisco IOS Release 12.2(33)SRA, this feature was integrated. In Cisco IOS Release 12.2(33)SXH, support was added. In Cisco IOS Release 12.4(20)T, this feature was integrated. In Cisco IOS Release 12.2(33)SRE. this feature was integrated. A router with primary one-hop autotunnels and backup autotunnels can be configured with SSO redundancy. In Cisco IOS Release 15.1(1)S, this feature was updated to allow you to specify affinity/mask for dynamically created MPLS TE backup tunnels. In Cisco IOS XE Release 2.3, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced: **affinity**, **mpls traffic-eng auto-tunnel backup config**, **show mpls traffic-eng auto-tunnel backup**.
MPLS TE - Autotunnel/Automesh SSO Coexistence	15.2(1)T Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, this feature was integrated. In Cisco IOS Release 15.2(1)T, this feature was integrated. **Note** Starting with Cisco IOS Release 15.2(2)S and Cisco IOS XE Release 3.6S, the SSO Support for MPLS TE Autotunnel and Automesh feature replaces the MPLS TE - Autotunnel/Automesh SSO Coexistence feature. For more information, see the *MPLS High Availability Configuration Guide* for the new implementation.

Glossary

backup tunnel --An MPLS traffic engineering tunnel used to protect other (primary) tunnel's traffic when a link or node failure occurs.

egress router --A router at the edge of the network where packets are leaving.

Fast Reroute --Fast Reroute (FRR) is a mechanism for protecting MPLS traffic engineering (TE) LSPs from link and node failure by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

hop --Passage of a data packet between two network nodes (for example, between two routers).

interface --A network connection.

IP address --A 32-bit address assigned to hosts using TCP/IP. An IP address belongs to one of five classes (A, B, C, D, or E) and is written as four octets separated by periods (dotted decimal format). Each address consists of a network number, an optional subnetwork number, and a host number. The network and subnetwork numbers together are used for routing, and the host number is used to address an individual host within the network or subnetwork. A subnet mask is used to extract network and subnetwork information from the IP address.

IP explicit path --A list of IP addresses, each representing a node or link in the explicit path.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

link --Point-to-point connection between adjacent nodes.

LSP --label switched path. A path that is followed by a labeled packet over several hops, starting at an ingress LSR and ending at an egress LSR.

LSR --label switch router. A Layer 3 router that forwards a packet based on the value of a label encapsulated in the packet.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets. ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

node --Endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network.

penultimate router --The second-to-last router; that is, the router that is immediately before the egress router.

primary tunnel --An MPLS tunnel whose LSP can be fast rerouted if there is a failure.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

router ID --Something by which a router originating a packet can be uniquely distinguished from all other routers. For example, an IP address from one of the router's interfaces.

scalability --An indicator showing how quickly some measure of resource usage increases as a network gets larger.

traffic engineering --The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

CHAPTER 103

MPLS Traffic Engineering (TE) Path Protection

The MPLS Traffic Engineering (TE): Path Protection feature provides an end-to-end failure recovery mechanism (that is, full path protection) for Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels.

Prerequisites for MPLS Traffic Engineering (TE) Path Protection

- Ensure that your network supports MPLS TE, Cisco Express Forwarding, and Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).
- Enable MPLS.
- Configure TE on the routers.
- Configure a TE tunnel with a primary path option by using the **tunnel mpls traffic-eng path-option** command.
- If your router supports SSO, configure Resource Reservation Protocol (RSVP) Graceful Restart in full mode on the routers.
- If your router supports SSO, for NSF operation you must have configured SSO on the device.

Restrictions for MPLS Traffic Engineering (TE) Path Protection

- There can be only one secondary path for each primary path option.
- The secondary path will not be signaled with the Fast Reroute (FRR) flag.

- Dynamic diverse paths are not supported.
- Do not use link and node protection with path protection on the headend router.
- Do not configure path protection on an automesh tunnel template because the destinations are different and you cannot use the same path option to reach multiple destinations.
- A lockdown option is not supported in protected path options.
- After an SSO event, path protection will not be immediately available on tunnels. Only a single label switched path (LSP) is checkpointed and recovered for the tunnel; the path-protected LSP will not be signaled until the end of the RSVP High Availability (HA) recovery period.

Information About MPLS Traffic Engineering (TE) Path Protection

Traffic Engineering Tunnels

MPLS TE lets you build label switched paths (LSPs) across your network for forwarding traffic.

MPLS TE LSPs, also called TE tunnels, let the headend of a TE tunnel control the path its traffic takes to a particular destination. This method is more flexible than forwarding traffic based only on a destination address.

Some tunnels are more important than others. For example, you may have tunnels carrying VoIP traffic and tunnels carrying data traffic that are competing for the same resources. MPLS TE allows you to have some tunnels preempt others. Each tunnel has a priority, and more-important tunnels take precedence over less-important tunnels.

Path Protection

Path protection provides an end-to-end failure recovery mechanism (that is, full path protection) for MPLS TE tunnels. A secondary LSP is established, in advance, to provide failure protection for the protected LSP that is carrying a tunnel's TE traffic. When there is a failure on the protected LSP, the headend router immediately enables the secondary LSP to temporarily carry the tunnel's traffic. If there is a failure on the secondary LSP, the tunnel no longer has path protection until the failure along the secondary path is cleared. Path protection can be used with a single area (OSPF or IS-IS), or Inter-AS (Border Gateway Protocol (BGP), external BGP (eBGP,) and static).

The failure detection mechanisms that trigger a switchover to a secondary tunnel include the following:

- Path error or resv tear from Resource Reservation Protocol (RSVP) signaling
- Notification from the RSVP hello that a neighbor is lost
- Notification from the Bidirectional Forwarding Detection (BFD) protocol that a neighbor is lost
- Notification from the Interior Gateway Protocol (IGP) that the adjacency is down
- Local teardown of the protected tunnel's LSP due to preemption in order to signal higher priority LSPs, a Packet over SONET (POS) alarm, online insertion and removal (OIR), and so forth

An alternate recovery mechanism is Fast Reroute (FRR), which protects MPLS TE LSPs only from link and node failures by locally repairing the LSPs at the point of failure.

Although not as fast as link or node protection, presignaling a secondary LSP is faster than configuring a secondary primary path option or allowing the tunnel's headend router to dynamically recalculate a path. The actual recovery time is topology-dependent, and affected by delay factors such as propagation delay or switch fabric latency.

Enhanced Path Protection

Enhanced path protection provides support of multiple backup path options per primary path option. You can configure up to eight backup path options for a given primary path option. Only one of the configured backup path options is actively signaled at any time.

After you enter the **mpls traffic-eng path-option list** command, you can enter the backup path priority in the *number* argument of the **path-option** command. A lower identifier represents a higher priority. Priorities are configurable for each backup path option. Multiple backup path options and a single backup path option cannot coexist to protect a primary path option.

ISSU

Cisco ISSU allows you to perform a Cisco IOS XE software upgrade or downgrade while the system continues to forward packets. ISSU takes advantage of the Cisco IOS XE high availability infrastructure--Cisco NSF with SSO and hardware redundancy--and eliminates downtime associated with software upgrades or version changes by allowing changes while the system remains in service. That lowers the impact that planned maintenance activities have on network service availability; there is less downtime and better access to critical systems.

When Path Protection is enabled and an ISSU upgrade is performed, path protection performance is similar to other TE features.

NSF/SSO

Cisco NSF with SSO provides continuous packet forwarding, even during a network processor hardware or software failure.

SSO takes advantage of Route Processor (RP) redundancy to increase network availability by establishing one of the RPs as the active processor while the other RP is designated as the secondary processor, and then synchronizing critical state information between them. Following an initial synchronization between the two processors, SSO dynamically maintains RP state information between them. A switchover from the active to the secondary processor occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance.

Cisco NSF works with SSO to minimize the amount of time a network is unavailable to users after a switchover. The main purpose of NSF is to continue forwarding IP packets after an RP switchover. Cisco NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

The MPLS Traffic Engineering: Path Protection feature can recover after SSO. A tunnel configured for path protection may have two LSPs signaled simultaneously: the primary LSP that is carrying the traffic and the secondary LSP that carries traffic in case there is a failure along the primary path. Only information associated with one of those LSPs, the one that is currently carrying traffic, is synched to the standby RP. The standby RP, upon recovery, can determine from the checkpointed information whether the LSP was the primary or secondary.

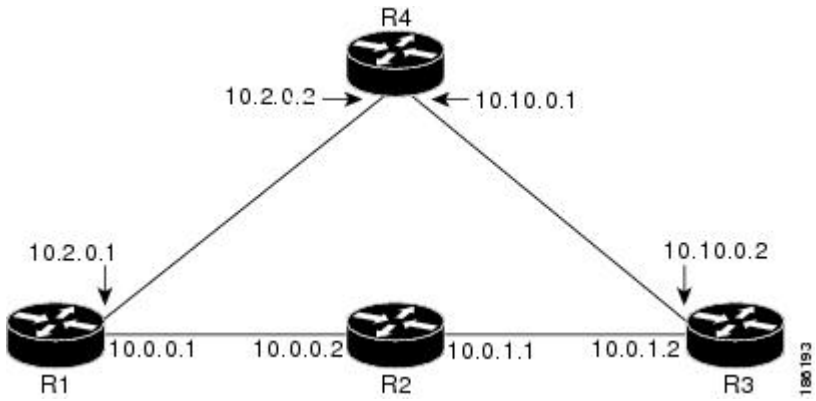
If the primary LSP was active during the switchover, only the primary LSP is recovered. The secondary LSP that was signaled and that provided path protection is resignaled after the TE recovery period is complete. This does not impact traffic on the tunnel because the secondary LSP was not carrying traffic.

How to Configure MPLS Traffic Engineering (TE) Path Protection

Regular Path Protection Configuration Tasks

This section contains the following tasks which are shown in the figure below.

Figure 173: Network Topology--Path Protection



Configuring Explicit Paths for Secondary Paths

To specify a secondary path that does not include common links or nodes associated with the primary path in case those links or nodes go down, configure an explicit path by performing the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip explicit-path** {**name** *path-name*| **identifier** *number*} [**enable** | **disable**]
4. **index** *index command ip-address*
5. **exit**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip explicit-path** {**name** *path-name*| **identifier** *number*} [**enable** | **disable**] **Example:** `Router(config)# ip explicit-path name path3441 enable`	Creates or modifies the explicit path and enters IP explicit path command mode.
Step 4	**index** *index command ip-address* **Example:** `Router(cfg-ip-exp1-path)# index 1 next-address 10.0.0.1`	Inserts or modifies a path entry at a specific index. The IP address represents the node ID. **Note** Enter this command once for each router.
Step 5	**exit** **Example:** `Router(cfg-ip-exp1-path)# exit`	Exits IP explicit path command mode and enters global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Assigning a Secondary Path Option to Protect a Primary Path Option

Assign a secondary path option in case there is a link or node failure along a path and all interfaces in your network are not protected.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng path-option protect** *number* **explicit** {**name** *path-name* | **identifier** *path-number*} [**verbatim**] [**attributes** *string*] [**bandwidth** *kb/s*| **sub-pool** *kb/s*]
5. **exit**
6. **exit**

DETAILED STEPS

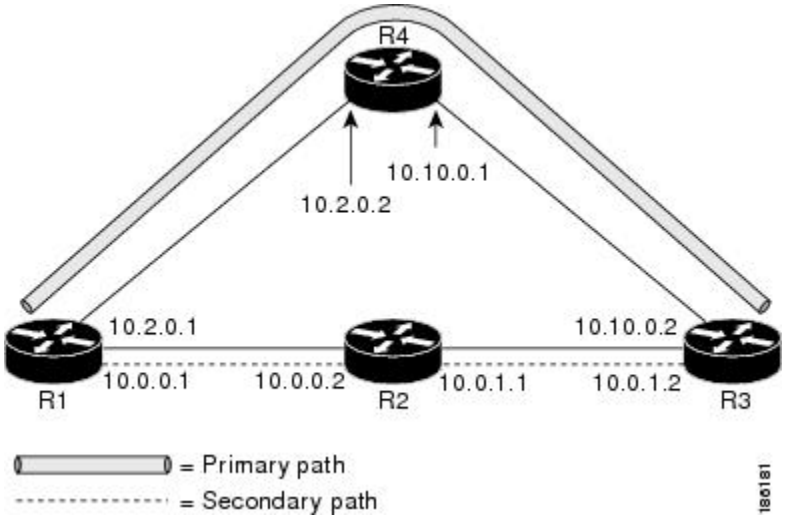
	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel500`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng path-option protect** *number* **explicit** {**name** *path-name* | **identifier** *path-number*} [**verbatim**] [**attributes** *string*] [**bandwidth** *kb/s*| **sub-pool** *kb/s*] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option protect 10 explicit name path344`	Configures a secondary path option for an MPLS TE tunnel.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the Configuration of MPLS Traffic Engineering Path Protection

To verify the configuration of path protection, perform the following steps. In Steps 1 and 2, refer to the figure below.

Figure 174: Network Topology Verification



SUMMARY STEPS

1. **show running interface tunnel** *tunnel-number*
2. **show mpls traffic-eng tunnels** *tunnel-interface*
3. **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**] **protection**
4. **show ip rsvp high-availability database** {**hello** | **link-management** {**interfaces** | **system**} | **lsp** [**filter destination** *ip-address*| **filter lsp-id** *lsp-id*| **filter source** *ip-address* | **filter tunnel-id** *tunnel-id*] | **lsp-head** [**filter** *number*] | **summary**}

DETAILED STEPS

Step 1 **show running interface tunnel** *tunnel-number*

This command shows the configuration of the primary path and protection path options.

Note To show the status of both LSPs (that is, both the primary path and the protected path), use the **show mpls traffic-eng tunnels protection** command.

Example:

```
Router# show running interface tunnel500

Building configuration...
Current configuration : 497 bytes
!
interface Tunnel500
 ip unnumbered Loopback0
 tunnel destination 10.0.0.9
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng path-option 10 explicit name path344
 tunnel mpls traffic-eng path-option 20 explicit name path345
 tunnel mpls traffic-eng path-option protect 10 explicit name path3441
```

```
 tunnel mpls traffic-eng path-option protect 20 explicit name path348
end
```

Step 2 **show mpls traffic-eng tunnels** *tunnel-interface*

This command shows tunnel path information.

The Common Link(s) field shows the number of links shared by both the primary and secondary paths, from the headend router to the tailend router.

The Common Node(s) field shows the number of nodes shared by both the primary and secondary paths, excluding the headend and tailend routers.

As shown in the following output, there are no common links or nodes:

Example:

```
Router# show mpls traffic-eng tunnels tunnel500
Name: R1_t500 (Tunnel500) Destination: 10.0.0.9
 Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 10, type explicit path344 (Basis for Setup, path weight 20)
  path option 20, type explicit path345
  Path Protection: 0 Common Link(s), 0 Common Node(s)
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kb/s (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
  State: explicit path option 10 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet1/0/0, 16
 RSVP Signalling Info:
    Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 19
  RSVP Path Info:
    My Address: 10.2.0.1
    Explicit Route: 10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
 Shortest Unconstrained Path Info:
  Path Weight: 20 (TE)
  Explicit Route: 10.2.0.1  10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
 History:
  Tunnel:
    Time since created: 11 minutes, 17 seconds
    Time since path change: 8 minutes, 5 seconds
    Number of LSP IDs (Tun_Instances) used: 19
  Current LSP:
    Uptime: 8 minutes, 5 seconds
```

Step 3 **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**] **protection**

Use this command, with the **protection** keyword specified, to show the status of both LSPs (that is, both the primary path and the protected path).

Note Deleting a primary path option has the same effect as shutting down a link. Traffic will move to the protected path in use.

The following command output shows that the primary LSP is up, and the secondary LSP is up and providing protection:

Example:

```
Router# show mpls traffic-eng tunnels tunnel500 protection
R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 19
 Fast Reroute Protection: None
 Path Protection: 0 Common Link(s), 0 Common Node(s)
  Primary lsp path:10.2.0.1  10.2.0.2
                   10.10.0.1 10.10.0.2
                   10.0.0.9
  Protect lsp path:10.0.0.1  10.0.0.2
                   10.0.1.1  10.0.1.2
                   10.0.0.9
  Path Protect Parameters:
   Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
  InLabel : -
  OutLabel : FastEthernet0/0/0, 16
  RSVP Signalling Info:
    Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 27
   RSVP Path Info:
    My Address: 10.0.0.1
    Explicit Route: 10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
   RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
```

The following command output shows that the primary LSP is down, and the secondary LSP is up and is actively carrying traffic:

Example:

```
Router# show mpls traffic-eng tunnels tunnel500 protection
R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 27
 Fast Reroute Protection: None
 Path Protection: Backup lsp in use.
```

Step 4 **show ip rsvp high-availability database** {**hello** | **link-management** {**interfaces** | **system**} | **lsp** [**filter destination** *ip-address*| **filter lsp-id** *lsp-id*| **filter source** *ip-address* | **filter tunnel-id** *tunnel-id*] | **lsp-head** [**filter** *number*] | **summary**}

The **show ip rsvp high-availability database** command displays the contents of the RSVP high availability (HA) read and write databases used in TE. If you specify the **lsp-head** keyword, the command output includes path protection information.

Example:

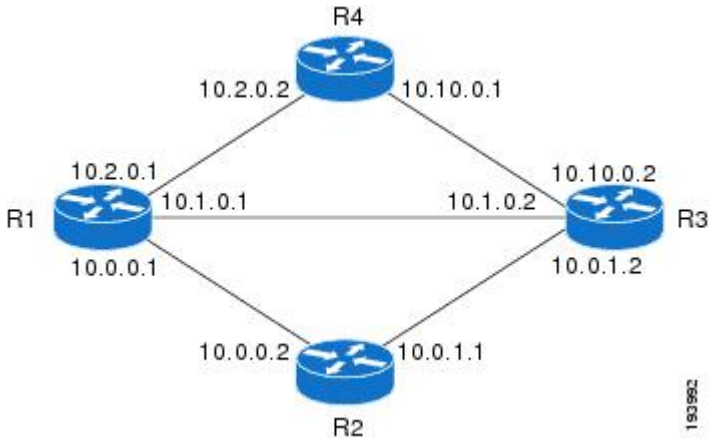
```
Router# show ip rsvp high-availability database lsp-head
LSP_HEAD WRITE DB
 Tun ID: 500
 Header:
  State: Checkpointed  Action: Add
```

```
 Seq #: 3             Flags:  0x0
Data:
 lsp_id: 5, bandwidth: 100, thead_flags: 0x1, popt: 1
 feature_flags: path protection active
 output_if_num: 5, output_nhop: 10,0,0,1
 RRR path setup info
  Destination: 10.0.0.9, Id: 10.0.0.9 Router Node (ospf) flag:0x0
  IGP: ospf, IGP area: 0, Number of hops: 5, metric: 2
  Hop 0: 10.0.0.1, Id: 10.0.0.1 Router Node (ospf), flag:0x0
  Hop 1: 10.0.0.2, Id: 10.0.0.7 Router Node (ospf), flag:0x0
  Hop 2: 10.0.1.1, Id: 10.0.0.7 Router Node (ospf), flag:0x0
  Hop 3: 10.0.1.2, Id: 10.0.0.9 Router Node (ospf), flag:0x0
  Hop 4: 10.0.0.9, Id: 10.0.0.9 Router Node (ospf), flag:0x0
```

Enhanced Path Protection Configuration Tasks

This section contains the following tasks which are shown in the figure below.

Figure 175: Network Topology - Enhanced Path Protection



Creating a Path Option List

Perform the following task to create a path option list of backup paths for a primary path option.

Note To use a secondary path instead, perform the steps in the Configuring Explicit Paths for Secondary Paths section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng path-option list** [**name** *pathlist-name* | **identifier** *pathlist-number*]
4. **path-option** *number* **explicit** [**name** *pathoption-name* | **identifier***pathoption-number*]
5. **list**

6. **no** [*pathoption-name* | *pathoption-number*]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng path-option list** [**name** *pathlist-name* | **identifier** *pathlist-number*] **Example:** `Router(config)# mpls traffic-eng path-option list name pathlist-01`	Configures a path option list, and enters path-option list configuration mode. • You can enter the following commands: **path-option**, **list**, **no**, and **exit**.
Step 4	**path-option** *number* **explicit** [**name** *pathoption-name* | **identifier***pathoption-number*] **Example:** `Router(cfg-pathoption-list)# path-option 10 explicit identifier 200`	(Optional) Specifies the name or identification number of the path option to add, edit, or delete. The *pathoption-number* value can be from 1 through 65535.
Step 5	**list** **Example:** `Router(cfg-pathoption-list)# list`	(Optional) Lists all of the path options.
Step 6	**no** [*pathoption-name* | *pathoption-number*] **Example:** `Router(cfg-pathoption-list)# no 10`	(Optional) Deletes a specified path option.
Step 7	**exit** **Example:** `Router(cfg-pathoption-list)# exit`	(Optional) Exits path-option list configuration mode and enters global configuration mode.

Assigning a Path Option List to Protect a Primary Path Option

Assign a path option list in case there is a link or node failure along a path and all interfaces in your network are not protected. See the third figure above.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng path-option protect** *number* [**attributes** *lsp-attributes* | **bandwidth** {*kbps* | **subpool** *kbps*} | **explicit** {**identifier** *path-number* | **name** *path-name*} | **list** {*pathlist-name* **name** | **identifier** *pathlist-identifier*}]
5. **exit**

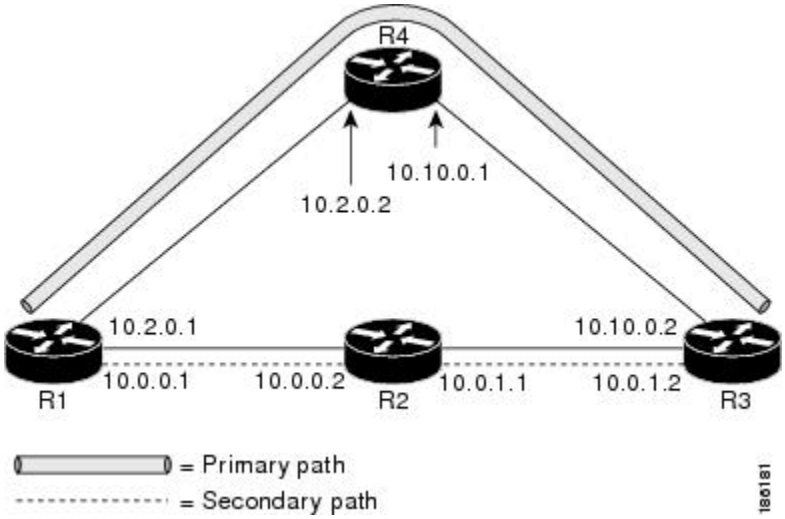
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel500`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng path-option protect** *number* [**attributes** *lsp-attributes* | **bandwidth** {*kbps* | **subpool** *kbps*} | **explicit** {**identifier** *path-number* | **name** *path-name*} | **list** {*pathlist-name* **name** | **identifier** *pathlist-identifier*}] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option protect 10 list name pathlist-01`	Configures a path option list to protect primary path option 10.
Step 5	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits interface configuration mode and enters global configuration mode.

Verifying the Configuration of MPLS Traffic Engineering Path Protection

To verify the configuration of path protection, perform the following steps. In Steps 1 and 2, refer to the figure below.

Figure 176: Network Topology Verification



SUMMARY STEPS

1. **show running interface tunnel** *tunnel-number*
2. **show mpls traffic-eng tunnels** *tunnel-interface*
3. **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**] **protection**
4. **show ip rsvp high-availability database** {**hello** | **link-management** {**interfaces** | **system**} | **lsp** [**filter destination** *ip-address*| **filter lsp-id** *lsp-id*| **filter source** *ip-address* | **filter tunnel-id** *tunnel-id*] | **lsp-head** [**filter** *number*] | **summary**}

DETAILED STEPS

Step 1 **show running interface tunnel** *tunnel-number*

This command shows the configuration of the primary path and protection path options.

Note To show the status of both LSPs (that is, both the primary path and the protected path), use the **show mpls traffic-eng tunnels protection** command.

Example:

```
Router# show running interface tunnel500

Building configuration...
Current configuration : 497 bytes
!
interface Tunnel500
 ip unnumbered Loopback0
 tunnel destination 10.0.0.9
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng path-option 10 explicit name path344
 tunnel mpls traffic-eng path-option 20 explicit name path345
 tunnel mpls traffic-eng path-option protect 10 explicit name path3441
```

```
 tunnel mpls traffic-eng path-option protect 20 explicit name path348
end
```

Step 2 **show mpls traffic-eng tunnels** *tunnel-interface*

This command shows tunnel path information.

The Common Link(s) field shows the number of links shared by both the primary and secondary paths, from the headend router to the tailend router.

The Common Node(s) field shows the number of nodes shared by both the primary and secondary paths, excluding the headend and tailend routers.

As shown in the following output, there are no common links or nodes:

Example:

```
Router# show mpls traffic-eng tunnels tunnel500
Name: R1_t500 (Tunnel500) Destination: 10.0.0.9
 Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 10, type explicit path344 (Basis for Setup, path weight 20)
  path option 20, type explicit path345
  Path Protection: 0 Common Link(s), 0 Common Node(s)
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kb/s (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
  State: explicit path option 10 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet1/0/0, 16
 RSVP Signalling Info:
    Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 19
  RSVP Path Info:
    My Address: 10.2.0.1
    Explicit Route: 10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
 Shortest Unconstrained Path Info:
  Path Weight: 20 (TE)
  Explicit Route: 10.2.0.1  10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
 History:
  Tunnel:
    Time since created: 11 minutes, 17 seconds
    Time since path change: 8 minutes, 5 seconds
    Number of LSP IDs (Tun_Instances) used: 19
  Current LSP:
    Uptime: 8 minutes, 5 seconds
```

Step 3 **show mpls traffic-eng tunnels** *tunnel-interface* [**brief**] **protection**

Use this command, with the **protection** keyword specified, to show the status of both LSPs (that is, both the primary path and the protected path).

Note Deleting a primary path option has the same effect as shutting down a link. Traffic will move to the protected path in use.

The following command output shows that the primary LSP is up, and the secondary LSP is up and providing protection:

Example:

```
Router# show mpls traffic-eng tunnels tunnel500 protection
R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 19
 Fast Reroute Protection: None
 Path Protection: 0 Common Link(s), 0 Common Node(s)
  Primary lsp path:10.2.0.1  10.2.0.2
                   10.10.0.1 10.10.0.2
                   10.0.0.9
  Protect lsp path:10.0.0.1  10.0.0.2
                   10.0.1.1  10.0.1.2
                   10.0.0.9
  Path Protect Parameters:
   Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
  InLabel : -
  OutLabel : FastEthernet1/2/0, 16
  RSVP Signalling Info:
    Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 27
   RSVP Path Info:
    My Address: 10.0.0.1
    Explicit Route: 10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
   RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
```

The following command output shows that the primary LSP is down, and the secondary LSP is up and is actively carrying traffic:

Example:

```
Router# show mpls traffic-eng tunnels tunnel500 protection
R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 27
 Fast Reroute Protection: None
 Path Protection: Backup lsp in use.
```

Step 4 **show ip rsvp high-availability database** {**hello** | **link-management** {**interfaces** | **system**} | **lsp** [**filter destination** *ip-address*| **filter lsp-id** *lsp-id*| **filter source** *ip-address* | **filter tunnel-id** *tunnel-id*] | **lsp-head** [**filter** *number*] | **summary**}

The **show ip rsvp high-availability database** command displays the contents of the RSVP high availability (HA) read and write databases used in TE. If you specify the **lsp-head** keyword, the command output includes path protection information.

Example:

```
Router# show ip rsvp high-availability database lsp-head
LSP_HEAD WRITE DB
 Tun ID: 500
 Header:
  State: Checkpointed  Action: Add
```

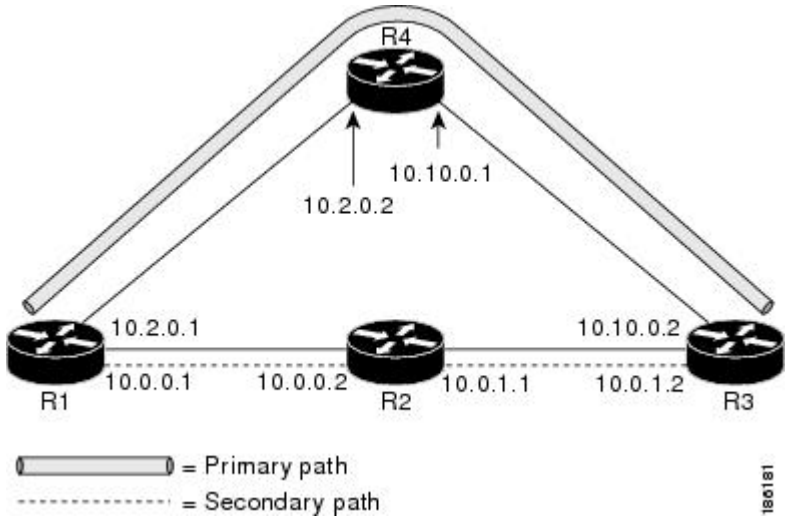
```
 Seq #: 3             Flags:  0x0
Data:
 lsp_id: 5, bandwidth: 100, thead_flags: 0x1, popt: 1
 feature_flags: path protection active
 output_if_num: 5, output_nhop: 10,0,0,1
 RRR path setup info
  Destination: 10.0.0.9, Id: 10.0.0.9 Router Node (ospf) flag:0x0
  IGP: ospf, IGP area: 0, Number of hops: 5, metric: 2
  Hop 0: 10.0.0.1, Id: 10.0.0.1 Router Node (ospf), flag:0x0
  Hop 1: 10.0.0.2, Id: 10.0.0.7 Router Node (ospf), flag:0x0
  Hop 2: 10.0.1.1, Id: 10.0.0.7 Router Node (ospf), flag:0x0
  Hop 3: 10.0.1.2, Id: 10.0.0.9 Router Node (ospf), flag:0x0
  Hop 4: 10.0.0.9, Id: 10.0.0.9 Router Node (ospf), flag:0x0
```

Configuration Examples for MPLS Traffic Engineering (TE): Regular Path Protection

Example Configuring Explicit Paths for Secondary Paths

The figure below illustrates a primary path and a secondary path. If there is a failure, the secondary path is used.

Figure 177: Primary Path and Secondary Path



In the following example the explicit path is named path3441. There is an **index** command for each router. If there is failure, the secondary path is used.

```
Router(config)# ip explicit-path name path3441 enable
Router(cfg-ip-expl-path)# index 1 next 10.0.0.1
Explicit Path name path3441:
    1: next-address 10.0.0.1
Router(cfg-ip-expl-path)# index 2 next 10.0.0.2
Explicit Path name path3441:
```

```
    1: next-address 10.0.0.1
    2: next-address 10.0.0.2
Router(cfg-ip-expl-path)# index 3 next 10.0.1.1
Explicit Path name path3441:
    1: next-address 10.0.0.1
    2: next-address 10.0.0.2
    3: next-address 10.0.1.1
Router(cfg-ip-expl-path)# index 4 next 10.0.1.2
Explicit Path name path3441:
    1: next-address 10.0.0.1
    2: next-address 10.0.0.2
    3: next-address 10.0.1.1
    4: next-address 10.0.1.2
Router(cfg-ip-expl-path)# exit
```

Example Assigning a Secondary Path Option to Protect a Primary Path Option

In the following example a traffic engineering tunnel is configured:

```
Router> enable
Router# configure terminal
Router(config-if)# interface tunnel500
Router(config-if)# tunnel mpls traffic-eng path-option protect 10 explicit name path344
```

The following **show running interface** command output shows that path protection has been configured. Tunnel 500 has path option 10 using path344 and protected by path 3441, and path option 20 using path345 and protected by path348.

```
Router# show running interface tunnel500
Router# interface tunnel 500
Building configuration...
Current configuration : 497 bytes
!
interface Tunnel500
 ip unnumbered Loopback0
 tunnel destination 10.0.0.9
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng path-option 10 explicit name path344
 tunnel mpls traffic-eng path-option 20 explicit name path345
 tunnel mpls traffic-eng path-option protect 10 explicit name path3441
 tunnel mpls traffic-eng path-option protect 20 explicit name path348
end
```

Example Configuring Tunnels Before and After Path Protection

The **show mpls traffic-eng tunnels** command shows information about the primary (protected) path. The following sample output shows that path protection has been configured.

```
Router# show mpls traffic-eng tunnels tunnel500
Name: R1_t500   (Tunnel500)   Destination: 10.0.0.9
 Status:
  Admin: up  Oper: up  Path: valid  Signalling: connected
  path option 10, type explicit path344 (Basis for Setup, path weight 20)
  path option 20, type explicit path345
  Path Protection: 0 Common Link(s), 0 Common Node(s)
```

```
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
    State: explicit path option 10 is active
BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet1/0/0, 16
 RSVP Signalling Info:
   Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 43
  RSVP Path Info:
   My Address: 10.2.0.1
   Explicit Route: 10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
   Record Route: NONE
   Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
   Record Route: NONE
   Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  Shortest Unconstrained Path Info:
   Path Weight: 20 (TE)
   Explicit Route: 10.0.0.1  10.0.0.2  10.0.1.1  10.0.1.2
                   10.0.0.9
 History:
  Tunnel:
   Time since created: 18 minutes, 22 seconds
   Time since path change: 19 seconds
   Number of LSP IDs (Tun_Instances) used: 43
  Current LSP:
   Uptime: 22 seconds
   Selection: reoptimization
  Prior LSP:
   ID: path option 10 [27]
   Removal Trigger: reoptimization completed
```

The following **show mpls traffic-eng tunnels** command output shows information about the secondary path. Tunnel500 is protected. The protection path is used, and the primary path is down. The command output shows the IP explicit paths of the primary LSP and the secondary LSP.

```
Router# show mpls traffic-eng tunnels tunnel500 protection

R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 43
 Fast Reroute Protection: None
 Path Protection: 0 Common Link(s), 0 Common Node(s)
  Primary lsp path:10.2.0.1  10.2.0.2
                   10.10.0.1 10.10.0.2
                   10.0.0.9
  Protect lsp path:10.0.0.1  10.0.0.2
                   10.0.1.1  10.0.1.2
                   10.0.0.9
  Path Protect Parameters:
   Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
  InLabel : -
  OutLabel : FastEthernet0/0/0, 17
  RSVP Signalling Info:
   Src 10.1.1.1,  Dst 10.0.0.9,  Tun_Id 500,  Tun_Instance 44
  RSVP Path Info:
```

```
   My Address: 10.0.0.1
   Explicit Route: 10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
   Record Route: NONE
   Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
   Record Route: NONE
   Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
R1#
```

The following **shutdown**command shuts down the interface to use path protection:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet1/0/0
Router(config-if)# shutdown
Router(config-if)# end
Router#
```

The following **show mpls traffic-eng tunnels** command shows that the protection path is used, and the primary path is down:

```
Router# show mpls traffic-eng tunnels tunnel500
Name: R1_t500 (Tunnel500) Destination: 10.0.0.9
 Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path option 10, type explicit path344
  path option 20, type explicit path345
  Path Protection: Backup lsp in use.
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
  State: explicit path option 10 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet0/0/0, 17
 RSVP Signalling Info:
    Src 10.1.1.1,  Dst 10.0.0.9,  Tun_Id 500,  Tun_Instance 44
  RSVP Path Info:
   My Address: 10.0.0.1
   Explicit Route: 10.0.0.2 10.0.1.1 10.0.1.2 10.0.0.9
   Record Route: NONE
   Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
   Record Route: NONE
   Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
 Shortest Unconstrained Path Info:
  Path Weight: 20 (TE)
  Explicit Route: 10.0.0.1  10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
 History:
  Tunnel:
   Time since created: 23 minutes, 28 seconds
   Time since path change: 50 seconds
   Number of LSP IDs (Tun_Instances) used: 44
 Current LSP:
  Uptime: 5 minutes, 24 seconds
  Selection:
```

```
 Prior LSP:
  ID: path option 10 [43]
  Removal Trigger: path error
  Last Error: PCALC:: Explicit path has unknown address, 10.2.0.1
R1#
```

The "up" value in the Oper field of the **show mpls traffic-eng tunnels protection** command shows that protection is enabled:

```
Router# show mpls traffic-eng tunnels tunnel500 protection

R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 44
 Fast Reroute Protection: None
 Path Protection: Backup lsp in use.
R1#
```

The **no shutdown** command in the following command sequence causes the interface to be up again and activates the primary path:

```
Router> enable

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet1/0/0
Router(config-if)# no shutdown
Router(config-if)# end
```

The following command output shows that path protection has been reestablished and the primary path is being used:

```
Router# show mpls traffic-eng tunnels tunnel500

Name: R1_t500 (Tunnel500) Destination: 10.0.0.9
 Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 10, type explicit path344 (Basis for Setup, path weight 20)
  path option 20, type explicit path345
  Path Protection: 0 Common Link(s), 0 Common Node(s)
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
  State: explicit path option 10 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet1/0/0, 16
 RSVP Signalling Info:
    Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 52
  RSVP Path Info:
    My Address: 10.2.0.1
    Explicit Route: 10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
```

```
  Shortest Unconstrained Path Info:
   Path Weight: 20 (TE)
   Explicit Route: 10.0.0.1  10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
  History:
   Tunnel:
    Time since created: 25 minutes, 26 seconds
    Time since path change: 23 seconds
    Number of LSP IDs (Tun_Instances) used: 52
   Current LSP:
   Uptime: 26 seconds
    Selection: reoptimization
   Prior LSP:
    ID: path option 10 [44]
    Removal Trigger: reoptimization completed
R1#
```

Following is sample **show mpls traffic-eng tunnels** command output. Tunnel500 is protected. After a failure, the primary LSP is protected.

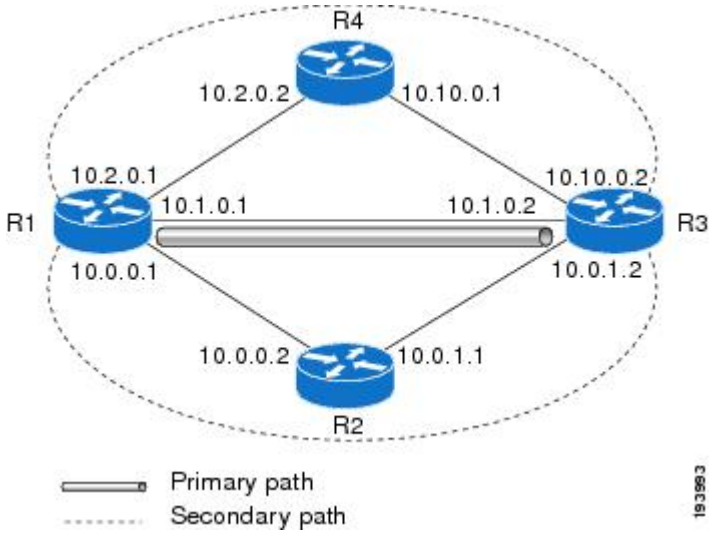
```
Router# show mpls traffic-eng tunnels tunnel500 protection

R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 52
 Fast Reroute Protection: None
 Path Protection: 0 Common Link(s), 0 Common Node(s)
  Primary lsp path:10.2.0.1   10.2.0.2
                   10.10.0.1  10.10.0.2
                   10.0.0.9
  Protect lsp path:10.0.0.1   10.0.2
                   10.0.1.1 10.0.1.2
                   10.0.0.9
  Path Protect Parameters:
   Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
  InLabel : -
  OutLabel : FastEthernet0/0/0, 16
  RSVP Signalling Info:
    Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 53
   RSVP Path Info:
    My Address: 10.0.0.1
    Explicit Route: 10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
   RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
R1#
```

Configuration Examples for MPLS Traffic Engineering (TE): Enhanced Path Protection

Creating a Path Option List: Example

The figure below shows the network topology for enhanced path protection.

p Network Topology for Enhanced Path Protection



The following example configures two explicit paths named **secondary1**and **secondary2**.

```
Router(config)# ip explicit-path name secondary1

Router(cfg-ip-expl-path)# index 1 next 10.0.0.2

Explicit Path name secondary1:
 1: next-address 10.0.0.2
Router(cfg-ip-expl-path)# index 2 next 10.0.1.2

Explicit Path name secondary1:
 1: next-address 10.0.0.2
 2: next-address 10.0.1.2
Router(cfg-ip-expl-path)# ip explicit-path name secondary2

Router(cfg-ip-expl-path)# index 1 next 10.2.0.2

Explicit Path name secondary2:
 1: next-address 10.2.0.2
Router(cfg-ip-expl-path)# index 2 next 10.10.0.2

Explicit Path name secondary2:
 1: next-address 10.2.0.2
 2: next-address 10.10.0.2
Router(cfg-ip-expl-path)# exit
```

In the following example a path option list of backup paths is created. You define the path option list by using the explicit paths.

```
Router(config)# mpls traffic-eng path-option list name pathlist-01

Router(cfg-pathoption-list)# path-option 10 explicit name secondary1

path-option 10 explicit name secondary1
Router(cfg-pathoption-list)# path-option 20 explicit name secondary2

path-option 10 explicit name secondary1
path-option 20 explicit name secondary2
Router(cfg-pathoption-list)# exit
```

Assigning a Path Option List to Protect a Primary Path Option: Example

In the following example, a traffic engineering tunnel is configured:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface tunnel 2

Router(config-if)# tunnel mpls traffic-eng path-option protect 10 list name secondary-list
```

The following **show running interface** command output shows that path protection has been configured. Tunnel 2 has path option 10 using path primary1 and protected by secondary-list.

```
Router# show running-config interface tunnel 2

Building configuration...
Current configuration : 296 bytes
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 103.103.103.103
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 explicit name primary1
 tunnel mpls traffic-eng path-option protect 10 list name secondary-list
```

Example Configuring Tunnels Before and After Path Protection

The **show mpls traffic-eng tunnels** command shows information about the primary (protected) path. The following sample output shows that path protection has been configured.

```
Router# show mpls traffic-eng tunnels tunnel500
Name: R1_t500   (Tunnel500)   Destination: 10.0.0.9
 Status:
  Admin: up  Oper: up  Path: valid  Signalling: connected
  path option 10, type explicit path344 (Basis for Setup, path weight 20)
  path option 20, type explicit path345
  Path Protection: 0 Common Link(s), 0 Common Node(s)
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
    State: explicit path option 10 is active
BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet1/0/0, 16
 RSVP Signalling Info:
   Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 43
  RSVP Path Info:
   My Address: 10.2.0.1
   Explicit Route: 10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
   Record Route: NONE
   Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
```

```
  RSVP Resv Info:
   Record Route: NONE
   Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  Shortest Unconstrained Path Info:
   Path Weight: 20 (TE)
   Explicit Route: 10.0.0.1  10.0.0.2  10.0.1.1  10.0.1.2
                   10.0.0.9
 History:
  Tunnel:
   Time since created: 18 minutes, 22 seconds
   Time since path change: 19 seconds
   Number of LSP IDs (Tun_Instances) used: 43
  Current LSP:
   Uptime: 22 seconds
   Selection: reoptimization
  Prior LSP:
   ID: path option 10 [27]
   Removal Trigger: reoptimization completed
```

The following **show mpls traffic-eng tunnels** command output shows information about the secondary path. Tunnel500 is protected. The protection path is used, and the primary path is down. The command output shows the IP explicit paths of the primary LSP and the secondary LSP.

```
Router# show mpls traffic-eng tunnels tunnel500 protection

R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 43
 Fast Reroute Protection: None
 Path Protection: 0 Common Link(s), 0 Common Node(s)
  Primary lsp path:10.2.0.1  10.2.0.2
                   10.10.0.1 10.10.0.2
                   10.0.0.9
  Protect lsp path:10.0.0.1  10.0.0.2
                   10.0.1.1  10.0.1.2
                   10.0.0.9
  Path Protect Parameters:
   Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
  InLabel : -
  OutLabel : FastEthernet0/0/0, 17
  RSVP Signalling Info:
   Src 10.1.1.1,  Dst 10.0.0.9,  Tun_Id 500,  Tun_Instance 44
  RSVP Path Info:
   My Address: 10.0.0.1
   Explicit Route: 10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
   Record Route: NONE
   Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
   Record Route: NONE
   Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
R1#
```

The following **shutdown**command shuts down the interface to use path protection:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet1/0/0
Router(config-if)# shutdown
Router(config-if)# end
Router#
```

The following **show mpls traffic-eng tunnels** command shows that the protection path is used, and the primary path is down:

```
Router# show mpls traffic-eng tunnels tunnel500
Name: R1_t500 (Tunnel500) Destination: 10.0.0.9
 Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path option 10, type explicit path344
  path option 20, type explicit path345
  Path Protection: Backup lsp in use.
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
  State: explicit path option 10 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet0/0/0, 17
 RSVP Signalling Info:
    Src 10.1.1.1,  Dst 10.0.0.9,  Tun_Id 500,  Tun_Instance 44
  RSVP Path Info:
    My Address: 10.0.0.1
    Explicit Route: 10.0.0.2 10.0.1.1 10.0.1.2 10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
 Shortest Unconstrained Path Info:
  Path Weight: 20 (TE)
  Explicit Route: 10.0.0.1  10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
 History:
  Tunnel:
    Time since created: 23 minutes, 28 seconds
    Time since path change: 50 seconds
    Number of LSP IDs (Tun_Instances) used: 44
 Current LSP:
  Uptime: 5 minutes, 24 seconds
  Selection:
 Prior LSP:
  ID: path option 10 [43]
  Removal Trigger: path error
  Last Error: PCALC:: Explicit path has unknown address, 10.2.0.1
R1#
```

The "up" value in the Oper field of the **show mpls traffic-eng tunnels protection** command shows that protection is enabled:

```
Router# show mpls traffic-eng tunnels tunnel500 protection

R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 44
 Fast Reroute Protection: None
 Path Protection: Backup lsp in use.
R1#
```

The **no shutdown** command in the following command sequence causes the interface to be up again and activates the primary path:

```
Router> enable

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet1/0/0
Router(config-if)# no shutdown
Router(config-if)# end
```

The following command output shows that path protection has been reestablished and the primary path is being used:

```
Router# show mpls traffic-eng tunnels tunnel500

Name: R1_t500 (Tunnel500) Destination: 10.0.0.9
 Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 10, type explicit path344 (Basis for Setup, path weight 20)
  path option 20, type explicit path345
  Path Protection: 0 Common Link(s), 0 Common Node(s)
  path protect option 10, type explicit path3441 (Basis for Protect, path weight 20)
  path protect option 20, type explicit path348
 Config Parameters:
  Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 100 bw-based
  auto-bw: disabled
 Active Path Option Parameters:
  State: explicit path option 10 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled
 InLabel : -
 OutLabel : FastEthernet1/0/0, 16
 RSVP Signalling Info:
   Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 52
  RSVP Path Info:
   My Address: 10.2.0.1
   Explicit Route: 10.2.0.2  10.10.0.1  10.10.0.2  10.0.0.9
   Record Route: NONE
   Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  RSVP Resv Info:
   Record Route: NONE
   Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
  Shortest Unconstrained Path Info:
   Path Weight: 20 (TE)
   Explicit Route: 10.0.0.1  10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
  History:
   Tunnel:
    Time since created: 25 minutes, 26 seconds
    Time since path change: 23 seconds
    Number of LSP IDs (Tun_Instances) used: 52
   Current LSP:
   Uptime: 26 seconds
    Selection: reoptimization
   Prior LSP:
    ID: path option 10 [44]
    Removal Trigger: reoptimization completed
R1#
```

Following is sample **show mpls traffic-eng tunnels** command output. Tunnel500 is protected. After a failure, the primary LSP is protected.

```
Router# show mpls traffic-eng tunnels tunnel500 protection

R1_t500
 LSP Head, Tunnel500, Admin: up, Oper: up
 Src 10.1.1.1, Dest 10.0.0.9, Instance 52
 Fast Reroute Protection: None
 Path Protection: 0 Common Link(s), 0 Common Node(s)
  Primary lsp path:10.2.0.1   10.2.0.2
                   10.10.0.1  10.10.0.2
                   10.0.0.9
  Protect lsp path:10.0.0.1   10.0.2
                   10.0.1.1 10.0.1.2
                   10.0.0.9
  Path Protect Parameters:
   Bandwidth: 100 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
  InLabel : -
  OutLabel : FastEthernet0/0/0, 16
  RSVP Signalling Info:
    Src 10.1.1.1, Dst 10.0.0.9, Tun_Id 500, Tun_Instance 53
   RSVP Path Info:
    My Address: 10.0.0.1
    Explicit Route: 10.0.0.2  10.0.1.1  10.0.1.2  10.0.0.9
    Record Route: NONE
    Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
   RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
R1#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS traffic engineering commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
RSVP commands	*Cisco IOS Quality of Service Solutions Command Reference*
IS-IS	• *Cisco IOS IP Routing Protocols Command Reference* • Configuring a Basic IS-IS Network
OSPF	• *Cisco IOS IP Routing Protocols Command Reference* • Configuring OSPF
ISSU	Cisco IOS XE In Service Software Upgrade Support
NSF/SSO	• Cisco Nonstop Forwarding • Stateful Switchover

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering Path Protection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 183: Feature Information for MPLS Traffic Engineering Path Protection

Feature Name	Releases	Feature Information
MPLS Traffic Engineering Path Protection	Cisco IOS XE Release 2.3	The MPLS Traffic Engineering (TE): Path Protection feature provides an end-to-end failure recovery mechanism (that is, full path protection) for MPLS TE tunnels. This feature was integrated into Cisco IOS XE Release 2.3. The following commands were introduced or modified: **show ip rsvp high-availability database**, **tunnel mpls traffic-eng path-option**, **tunnel mpls traffic-eng path-option protect**.
ISSU--MPLS Traffic Engineering (TE)--Path Protection	Cisco IOS XE Release 2.3	Cisco ISSU allows you to perform a Cisco IOS XE software upgrade or downgrade while the system continues to forward packets. This feature was integrated into Cisco IOS XE Release 2.3.
NSF/SSO--MPLS Traffic Engineering (TE)--Path Protection	Cisco IOS XE Release 2.3	Cisco NSF with SSO provides continuous packet forwarding, even during a network processor hardware or software failure. This feature was integrated into Cisco IOS XE Release 2.3.
MPLS TE--Enhanced Path Protection	Cisco IOS XE Release 3.5S	Enhanced path protection provides support of multiple backup path options per primary path option. This feature was integrated into Cisco IOS XE Release 3.5S. The following commands were added or modified: **mpls traffic-eng path-option list**, **show mpls traffic-eng path-option list** , **show mpls traffic-eng tunnels**, and **tunnel mpls traffic-eng path-option protect**.

Glossary

autotunnel mesh group --An autotunnel mesh group (referred to as a mesh group) is a set of connections between edge LSRs in a network.

backup tunnel --An MPLS TE tunnel used to protect other (primary) tunnels' traffic when a link or node failure occurs.

BGP --Border Gateway Protocol. An interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems).

Cisco Express Forwarding --A means for accelerating the forwarding of packets within a router, by storing route lookup.

Fast Reroute --Procedures that enable temporary routing around a failed link or node while a new LSP is being established at the headend.

graceful restart --A process for helping an RP restart after a node failure has occurred.

headend --The router that originates and maintains a given LSP. This is the first router in the LSP's path.

hop --Passage of a data packet between two network nodes (for example, between two routers).

interface --A network connection.

IS-IS --Intermediate System-to-Intermediate System. Link-state hierarchical routing protocol that calls for intermediate system (IS) routers to exchange routing information based on a single metric to determine network topology.

ISSU --In Service Software Upgrade. The ISSU process allows Cisco IOS XE software at the router level to be updated or otherwise modified while packet forwarding continues.

link --A point-to-point connection between adjacent nodes. There can be more than one link between adjacent nodes. A link is a network communications channel consisting of a circuit or transmission path and all related equipment between a sender and a receiver. Sometimes referred to as a line or a transmission link.

LSP --label switched path. A configured connection between two routers, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

MPLS --Multiprotocol Label Switching. Packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

NHOP --next hop. The next downstream node along an LSP's path.

NHOP backup tunnel --next-hop backup tunnel. The backup tunnel terminating at the LSP's next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link, and is used to protect primary LSPs that were using this link before the failure.

NNHOP --next-next hop. The node after the next downstream node along an LSP's path.

NNHOP backup tunnel --next-next-hop backup tunnel. The backup tunnel terminating at the LSP's next-next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link or node, and is used to protect primary LSPs that were using this link or node before the failure.

node --The endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network. Nodes can be processors, controllers, or workstations.

NSF --Cisco nonstop forwarding. Cisco NSF always runs with stateful switchover (SSO) and provides redundancy for Layer 3 traffic. NSF works with SSO to minimize the amount of time that a network is unavailable to its users following a switchover. The main purpose of NSF is to continue forwarding IP packets following a supervisor engine switchover.

OSPF --Open Shortest Path First. A link-state hierarchical Interior Gateway Protocol routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

primary LSP --The last LSP originally signaled over the protected interface before the failure. A primary LSP is signaled by configuring a primary path option.

primary tunnel --A tunnel whose LSP may be fast rerouted if there is a failure. Backup tunnels cannot be primary tunnels.

protected interface --An interface that has one or more backup tunnels associated with it.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RP --Route Processor. A generic term for the centralized control unit in a chassis.

RSVP --Resource Reservation Protocol. An IETF protocol used for signaling requests (setting up reservations) for Internet services by a customer before that customer is permitted to transmit data over that portion of the network.

secondary LSP --The LSP that is signaled to provide path protection. A secondary LSP protects a primary LSP.

secondary path option --Configuration of the path option that provides protection.

SRLG --Shared Risk Link Group. Sets of links that are likely to go down together (for example, because they have the same underlying fiber).

state --Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

tailend --The router upon which an LSP is terminated. This is the last router in the LSP's path.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

topology --The physical arrangement of network nodes and media within an enterprise networking structure.

tunnel --Secure communications path between two peers, such as two routers.

VoIP --Voice over IP. The capability of a router to carry voice traffic (for example, telephone calls and faxes) over an IP network. Cisco's voice support is implemented by using voice packet technology.

CHAPTER 104

MPLS Traffic Engineering BFD-triggered Fast Reroute

The MPLS Traffic Engineering: BFD-triggered Fast Reroute feature allows you to obtain link and node protection by using the Bidirectional Forwarding Detection (BFD) protocol to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators.

To obtain link and node protection by using the Resource Reservation Protocol (RSVP) with Hellos support, refer to the MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) process module. RSVP Hellos enable a router to detect when a neighboring node has gone down but its interface to that neighbor is still operational.

Prerequisites for MPLS Traffic Engineering BFD-triggered Fast Reroute

- Configure BFD. Refer to the *Bidirectional Forwarding Detection* process module.
- Enable MPLS TE on all relevant routers and interfaces.
- Configure MPLS TE tunnels.
- For additional prerequisites, refer to the MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) process module.

Restrictions for MPLS Traffic Engineering BFD-triggered Fast Reroute

- You cannot configure BFD and RSVP Hellos on the same interface.
- BFD may not be supported on some interfaces.
- For additional restrictions, refer to the MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection) process module.

Information About MPLS Traffic Engineering BFD-triggered Fast Reroute

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol Hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable.

Fast Reroute

Fast Reroute (FRR) is a mechanism for protecting Multiprotocol Label Switching (MPLS) traffic engineering (TE) label switched paths (LSPs) from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

Link Protection

Backup tunnels that bypass only a single link of the LSP's path provide link protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure.

Node Protection

FRR provides node protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called next-next-hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSP paths, thereby bypassing the next-hop node. They protect LSPs if a node along their path fails by enabling the node upstream of the failure to reroute the LSPs and their traffic around the failed node

to the next-next hop. FRR supports the use of RSVP Hellos to accelerate the detection of node failures. NNHOP backup tunnels also provide protection from link failures, because they bypass the failed link as well as the node.

Bandwidth Protection

NHOP and NNHOP backup tunnels can be used to provide bandwidth protection for rerouted LSPs. This is referred to as backup bandwidth. You can associate backup bandwidth with NHOP or NNHOP backup tunnels. This informs the router of the amount of backup bandwidth a particular backup tunnel can protect. When a router maps LSPs to backup tunnels, bandwidth protection ensures that an LSP uses a given backup tunnel only if there is sufficient backup bandwidth. The router selects which LSPs use which backup tunnels in order to provide maximum bandwidth protection. That is, the router determines the best way to map LSPs onto backup tunnels in order to maximize the number of LSPs that can be protected.

How to Configure MPLS Traffic Engineering BFD-triggered Fast Reroute

This section shows you how to add FRR protection to a network in which MPLS TE LSPs are configured.

The following sections describe how to use FRR to protect LSPs in your network from link or node failures. Each task is identified as either required or optional.

Note You can perform the configuration tasks in any order.

Note An NNHOP backup tunnel must *not* go via the NHOP backup tunnel.

Enabling BFD Support on the Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello bfd**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello bfd** **Example:** `Router(config)# ip rsvp signalling hello bfd`	Enables the BFD protocol on the router for MPLS TE link and node protection.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Enabling Fast Reroute on LSPs

LSPs can use backup tunnels only if the LSPs have been configured as fast reroutable. To enable FRR on the LSP, enter the following commands at the headend of each LSP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng fast-reroute** [**bw-protect**] [**node-protect**]
5. **exit**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number*	Enters interface configuration mode for the specified tunnel.

	Command or Action	Purpose
	Example: `Router(config)# interface tunnel 1000`	• The *number* argument is the number of the tunnel.
Step 4	**tunnel mpls traffic-eng fast-reroute** [**bw-protect**] [**node-protect**] **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect node-protect`	Enables an MPLS TE tunnel to use an established backup tunnel if there is a link or node failure. • The **bw-protect** keyword sets the "bandwidth protection desired" bit so that backup bandwidth protection is enabled. • The **node-protect** keyword sets the "node protection desired" bit so that backup bandwidth protection is enabled.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Creating a Backup Tunnel to the Next Hop or to the Next-Next Hop

To create a backup tunnel to the next hop or to the next-next hop, perform the following task.

Enter the commands on the node that will be the headend of the backup tunnel (that is, the node whose downstream link or node may fail). The node on which you enter the commands must be a supported platform. See the Finding Feature Information section.

Creating a backup tunnel is basically no different from creating any other tunnel.

Note When using the **exclude-address** command to specify the path for a backup tunnel, you must exclude an interface address to avoid a link (for creating an NHOP backup tunnel), or a router-ID address to avoid a node (for creating an NNHOP backup tunnel).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**

7. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}}[**lockdown**]
8. **exit**
9. **ip explicit-path name** *name*
10. **exclude-address** *address*
11. **exit**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Creates a new tunnel interface and enters interface configuration mode. • The *number* argument is the number of the tunnel.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface. • The *type* and *number* arguments name the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface. **Note** The **ip unnumbered loopback 0** command gives the tunnel interface an IP address that is the same as that of interface loopback 0. This command is not effective until loopback 0 has been configured with an IP address.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 10.3.3.3`	Specifies the destination for a tunnel interface. • The *ip-address*argument specifies the IP address of the device, expressed in dotted decimal notation, where the tunnel will terminate. That address should be the router ID of the device that is the NHOP or NNHOP of LSPs to be protected.

	Command or Action	Purpose
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets encapsulation mode of the tunnel to MPLS TE.
Step 7	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}}[**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 10 explicit name avoid-protected-link`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. • The *number* argument is the preference for this path option. When you configure multiple path options, lower numbered options are preferred. Valid values are from 1 to 1000. • The **dynamic** keyword indicates that the path of the label switched path (LSP) is dynamically calculated. • The **explicit** keyword indicates that the path of the LSP is an IP explicit path. • The **name** *path-name* keyword and argument are the path name of the IP explicit path that the tunnel uses with this option. • The **identifier** *path-number*keyword and argument pair names the path number of the IP explicit path that the tunnel uses with this option. The range is from 1 to 65535. • The **lockdown** keyword specifies that The LSP cannot be reoptimized. **Note** A dynamic path is used if an explicit path is currently unavailable.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enter global configuration mode.
Step 9	**ip explicit-path name** *name* **Example:** `Router(config)# ip explicit-path name avoid-protected-link`	Enters IP explicit path mode for IP explicit paths to create the named path. • The *name* argument is the name of the explicit path.
Step 10	**exclude-address** *address* **Example:** `Router(cfg-ip-expl-path)# exclude-address 10.3.3.3`	Excludes an address from an explicit-path. • The *address* argument specifies the IP address of the link to be protected for link protection. For node protection, it specifies the router ID of the node to be protected.

	Command or Action	Purpose
		Note Backup tunnel paths can be dynamic or explicit and they do not have to use an excluded address. Because backup tunnels must avoid the protected link or node, it is convenient to use an excluded address.
Step 11	**exit** **Example:** `Router(cfg-ip-expl-path))# exit`	Exits IP explicit path configuration mode and returns to global configuration mode.
Step 12	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Assigning Backup Tunnels to a Protected Interface

To assign one or more backup tunnels to a protected interface, perform the following task.

Enter the commands on the node that will be the headend of the backup tunnel (that is, the node whose downstream link or node may fail). The node on which you enter the commands must be a supported platform. See the Finding Feature Information section.

Note You must configure the interface to have an IP address and to enable the MPLS TE tunnel feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type* slot/subs*lot* / *port*[**.** *subinterface*]
4. **mpls traffic-eng backup-path tunnel** *tunnel-id*
5. **exit**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* slot/sub*slot* / *port*[**.** *subinterface*] **Example:** `Router(config)# interface Gigabitethernet 2/1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the chassis slot number. Refer to the appropriate hardware manual for slot information. For SIPs, refer to the platform-specific SPA hardware installation guide or the corresponding "Identifying Slots and Subslots for SIPs and SPAs" topic in the platform-specific SPA software configuration guide. • The **/** *subslot* keyword and argument pair is the secondary slot number on a SIP where a SPA is installed. The slash (/) is required. Refer to the platform-specific SPA hardware installation guide and the corresponding "Specifying the Interface Address on a SPA" topic in the platform-specific SPA software configuration guide for subslot information. • The **/** *port* keyword and argument pair is the port or interface number. The slash (/) is required. Refer to the appropriate hardware manual for port information. For SPAs, refer to the corresponding "Specifying the Interface Address on a SPA" topics in the platform-specific SPA software configuration guide • The **.** *subinterface-number*keyword and argument pair is the subinterface number in the range 1 to 4294967293. The number that precedes the period (.) must match the number to which this subinterface belongs.
Step 4	**mpls traffic-eng backup-path tunnel** *tunnel-id* **Example:** `Router(config-if)# mpls traffic-eng backup-path tunnel2`	Configures the physical interface to use for a backup tunnel in the event of a detected failure on that interface. • The *tunnel-id*argument is a string that identifies a backup tunnel to use if there is a link or node failure for LSPs going out the configured interface. **Note** You can enter this command multiple times to associate multiple backup tunnels with the same protected interface.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if))# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Enabling BFD on the Protected Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type* slot/subs*lot* / *port*[**.** *subinterface*]
4. **ip rsvp signalling hello bfd**
5. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
6. **exit**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* slot/subs*lot* / *port*[**.** *subinterface*] **Example:** `Router(config)# interface Gigabitethernet 2/1/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the chassis slot number. Refer to the appropriate hardware manual for slot information. For SIPs, refer to the platform-specific SPA hardware installation guide or the corresponding “Identifying

	Command or Action	Purpose
		Slots and Subslots for SIPs and SPAs" topic in the platform-specific SPA software configuration guide. • The **/** *subslot* keyword and argument pair is the secondary slot number on a SIP where a SPA is installed. The slash (/) is required. Refer to the platform-specific SPA hardware installation guide and the corresponding "Specifying the Interface Address on a SPA" topic in the platform-specific SPA software configuration guide for subslot information. • The **/** *port* keyword and argument pair is the port or interface number. The slash (/) is required. Refer to the appropriate hardware manual for port information. For SPAs, refer to the corresponding "Specifying the Interface Address on a SPA" topics in the platform-specific SPA software configuration guide • The **.** *subinterface-number*keyword and argument pair is the subinterface number in the range 1 to 4294967293. The number that precedes the period (.) must match the number to which this subinterface belongs.
Step 4	**ip rsvp signalling hello bfd** **Example:** `Router(config-if)# ip rsvp signalling hello bfd`	Enables the BFD protocol on an interface for MPLS TE link and node protection.
Step 5	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Router(config-if)# bfd interval 100 min_rx 100 multiplier 4`	Sets the BFD session parameters for an interface. • The **interval** *milliseconds* keyword and argument pair specifies the rate at which BFD control packets will be sent to BFD peers. The configurable time period for the milliseconds argument is from 50 to 999. • The **min_rx** *millisecond* keyword and argument pair specifies the rate at which BFD control packets will be expected to be received from BFD peers. The configurable time period for the milliseconds argument is from 1 to 999. • The **multiplier** i*nterval-multiplier*keyword and argument pair specifies the number of consecutive BFD control packets that must be missed from a BFD peer before BFD declares that the peer is unavailable and the Layer 3 BFD peer is informed of the failure. The configurable value range for the multiplier-value argument is from 3 to 50.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-if))# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Associating Backup Bandwidth and Pool Type with a Backup Tunnel

To associate backup bandwidth with a backup tunnel and designate the type of LSP that can use a backup tunnel, enter the following tasks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng backup-bw** {*bandwidth* | [**sub-pool** {*bandwidth* | **Unlimited**}] [**global-pool** {*bandwidth* | **Unlimited**}]} [**any** {*bandwidth* | **Unlimited**}]
5. **exit**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 2`	Enters interface configuration mode for the specified tunnel. • The *number* argument is the number of the tunnel.

	Command or Action	Purpose
Step 4	**tunnel mpls traffic-eng backup-bw** {*bandwidth* | [**sub-pool** {*bandwidth* | **Unlimited**}] [**global-pool** {*bandwidth* | **Unlimited**}]} [**any** {*bandwidth* | **Unlimited**}] **Example:** `Router(config-if)# tunnel mpls traffic-eng backup-bw sub-pool 1000`	Associates bandwidth with a backup tunnel and designates whether LSPs that allocate bandwidth from the specified pool can use the tunnel.
Step 5	**exit** **Example:** `Router(config-if))# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Backup Bandwidth Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel mpls traffic-eng fast-reroute** [**bw-protect**]
5. **exit**
6. **mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:**	Enters interface configuration mode for the specified tunnel. • The *number* argument is the number of the tunnel.

	Command or Action	Purpose
	`Router(config)# interface tunnel 2`	
Step 4	**tunnel mpls traffic-eng fast-reroute** [**bw-protect**] **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect`	Enables an MPLS TE tunnel to use an established backup tunnel in the event of a link or node failure. • The **bw-protect** keyword gives an LSP priority for using backup tunnels with bandwidth protection.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 6	**mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw** **Example:** `Router(config)# mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw`	Changes the backup protection preemption algorithm from minimize the number of LSPs that are demoted to minimize the amount of bandwidth that is wasted.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying That Fast Reroute Is Operational

SUMMARY STEPS

1. **show mpls traffic-eng tunnels brief**
2. **show ip rsvp sender detail**
3. **show mpls traffic-eng fast-reroute database**
4. **show mpls traffic-eng tunnels backup**
5. **show mpls traffic-eng fast-reroute database**
6. **show ip rsvp reservation detail**
7. **show ip rsvp hello**
8. **show ip rsvp interface detail**
9. **show ip rsvp hello bfd nbr**
10. **show ip rsvp hello bfd nbr detail**
11. **show ip rsvp hello bfd nbr summary**

DETAILED STEPS

Step 1 **show mpls traffic-eng tunnels brief**

Use this command to verify that backup tunnels are up:

Example:

```
Router# show mpls traffic-eng tunnels brief

Signalling Summary:
    LSP Tunnels Process:           running
    RSVP Process:                  running
    Forwarding:                    enabled
    Periodic reoptimization:       every 3600 seconds, next in 1706 seconds
TUNNEL NAME                      DESTINATION      UP IF     DOWN IF   STATE/PROT
Router_t1                        10.112.0.12      -         Gi4/0/1   up/up
Router_t2                        10.112.0.12      -         unknown   up/down
Router_t3                        10.112.0.12      -         unknown   admin-down
Router_t1000                     10.110.0.10      -         unknown   up/down
Router_t2000                     10.110.0.10      -         Gi4/0/1   up/up
Displayed 5 (of 5) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Step 2 **show ip rsvp sender detail**

Use this command to verify that LSPs are protected by the appropriate backup tunnels.

Following is sample output from the **show ip rsvp sender detail** command when the command is entered at the router acting as the point of local repair (PLR) before a failure:

Example:

```
Router# show ip rsvp sender detail

PATH:
 Tun Dest:   10.10.0.6  Tun ID: 100  Ext Tun ID: 10.10.0.1
 Tun Sender: 10.10.0.1  LSP ID: 31
 Path refreshes:
  arriving: from PHOP 10.10.7.1 on Et0/0 every 30000 msecs
 Session Attr:
  Setup Prio: 7, Holding Prio: 7
  Flags: (0x7) Local Prot desired, Label Recording, SE Style
  session Name: R1_t100
 ERO: (incoming)
  10.10.7.2 (Strict IPv4 Prefix, 8 bytes, /32)
  10.10.0.6 (Strict IPv4 Prefix, 8 bytes, /32)
 RRO:
   10.10.7.1/32, Flags:0x0 (No Local Protection)
   10.10.4.1/32, Flags:0x9 (Local Prot Avail/to NNHOP) !Available to NNHOP
   10.10.1.1/32, Flags:0x0 (No Local Protection)
 Traffic params - Rate: 10K bits/sec, Max. burst: 1K bytes
   Min Policed Unit: 0 bytes, Max Pkt Size 4294967295 bytes
 Fast-Reroute Backup info:
   Inbound  FRR: Not active
   Outbound FRR: No backup tunnel selected
 Path ID handle: 50000416.
 Incoming policy: Accepted. Policy source(s): MPLS/TE
 Status: Proxy-terminated
```

Step 3 **show mpls traffic-eng fast-reroute database**

Enter the **clear ip rsvp hello instance counters**command to verify the following:

- MPLS TE FRR Node Protection has been enabled.
- A certain type of LSP can use a backup tunnel.

The following command output displays the LSPs that are protected:

Example:

```
Router# show mpls traffic-eng fast-reroute database
Tunnel head end item frr information:
Protected tunnel              In-label Out intf/label   FRR intf/label   Status
Tunnel500                     Tun hd   AT4/0.100:Untagg Tu501:20         ready
Prefix item frr information:
Prefix              Tunnel     In-label Out intf/label    FRR intf/label  Status
10.0.0.8/32         Tu500      18       AT4/0.100:Pop ta  Tu501:20        ready
10.0.8.8/32         Tu500      19       AT4/0.100:Untagg  Tu501:20        ready
10.8.9.0/24         Tu500      22       AT4/0.100:Untagg  Tu501:20        ready
LSP midpoint item frr information:
LSP identifier     In-label Out     intf/label     FRR intf/label     Status
```

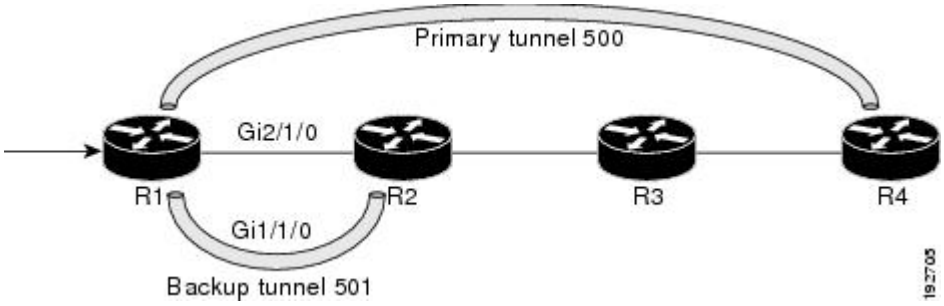
If Label Distribution Protocol (LDP) is not enabled, separate prefix items are not shown because all prefixes then use a single rewrite. To confirm that a particular IP prefix is FRR protected, even though it is not shown in this display, enter it within the **show mpls forwarding-table** *ip-address* **detail** command. The final line of the display will tell whether that prefix is protected:

Example:

```
Router# show mpls forwarding-table 10.0.0.11 32 detail

Local    Outgoing    Prefix         Bytes tag   Outgoing              Next Hop
tag      tag or VC   or Tunnel Id   switched    interface
Tun hd   Untagged    10.0.0.11/32   48 5/0      Gi5/0    point2point
          MAC/Encaps=4/8, MTU=1520, Tag Stack{22}
          48D18847 00016000
          No output feature configured
          Fast Reroute Protection via (Tu0, outgoing label 12304)
```

The following command output displays the LSPs that are protected when the FRR primary tunnel is over a Gigabit Ethernet interface and the backup tunnel is over a Gigabit Ethernet interface. As shown in the figure below, interface Gigabit Ethernet 2/1/0 is protected by backup tunnel 501.

Figure 178: Protected LSPs



The figure above shows the following:

- Primary tunnel 500--Path is R1 via Gigabit Ethernet2/1/0 to R2 to R3 to R4.
- FRR backup tunnel 501--Path is R1 via Gigabit Ethernet1/1/0 to R2.
- Interface Gigabit Ethernet1/1/0--Protected by backup tunnel 501.

Example:

```
Router# show mpls traffic-eng fast-reroute database
Tunnel head end item frr information:
Protected tunnel In-label Out intf/label FRR intf/label Status
Tunnel500 Tun hd AT4/0.100:Untagg Tu501:20 ready
Prefix item frr information:
Prefix Tunnel In-label Out intf/label FRR intf/label Status
10.0.0.8/32 Tu500 18 AT4/0.100:Pop ta Tu501:20 ready
10.0.8.8/32 Tu500 19 AT4/0.100:Untagg Tu501:20 ready
10.8.9.0/24 Tu500 22 AT4/0.100:Untagg Tu501:20 ready
LSP midpoint item frr information:
LSP identifier In-label Out intf/label FRR intf/label Status
```

The following command output displays the LSPs that are protected when the FRR backup tunnel is over a Gigabit Ethernet interface.

Example:

```
Router# show mpls traffic-eng fast-reroute database

Tunnel head end item frr information:
Protected tunnel In-label Out intf/label FRR intf/label Status
Tunnel500 Tun hd PO2/0:Untagged Tu501:20 ready
Prefix item frr information:
Prefix Tunnel In-label Out intf/label FRR intf/label Status
10.0.0.8/32 Tu500 18 PO2/0:Pop tag Tu501:20 ready
10.0.8.8/32 Tu500 19 PO2/0:Untagged Tu501:20 ready
10.8.9.0/24 Tu500 22 PO2/0:Untagged Tu501:20 ready
LSP midpoint item frr information:
LSP identifier In-label Out intf/label FRR intf/label Status
```

Step 4 **show mpls traffic-eng tunnels backup**

For backup tunnels to be operational, the LSP must be reroutable. At the headend of the LSP, enter the **show run interface tunnel** *tunnel-number* command. The output should include the **tunnel mpls traffic-eng fast-reroute** command. If it does not, enter this command for the tunnel.

On the router where the backup tunnels originate, enter the **show mpls traffic-eng tunnels backup**command. Following is sample command output:

Example:

```
Router# show mpls traffic-eng tunnels backup
Router_t578
  LSP Head, Tunnel578, Admin: up, Oper: up
  Src 10.55.55.55, Dest 10.88.88.88, Instance 1
  Fast Reroute Backup Provided:
    Protected i/fs: PO1/0, PO1/1, PO3/3
    Protected lsps: 1
    Backup BW: any pool unlimited; inuse: 100 kbps
Router_t5710
  LSP Head, Tunnel5710, Admin: admin-down, Oper: down
  Src 10.55.55.55, Dest 10.7.7.7, Instance 0
  Fast Reroute Backup Provided:
    Protected i/fs: PO1/1
    Protected lsps: 0
    Backup BW: any pool unlimited; inuse: 0 kbps
Router_t5711
  LSP Head, Tunnel5711, Admin: up, Oper: up
  Src 10.55.55.55, Dest 10.7.7.7, Instance 1
  Fast Reroute Backup Provided:
    Protected i/fs: PO1/0
```

```
Protected lsps: 2
Backup BW: any pool unlimited; inuse: 6010 kbps
```

The command output will allow you to verify the following:

- Backup tunnel exists--Verify that there is a backup tunnel that terminates at this LSP's NHOP or NNHOP. Look for the LSP's NHOP or NNHOP in the Dest field.
- Backup tunnel is up--To verify that the backup tunnel is up, look for "Up" in the Oper field.
- Backup tunnel is associated with the LSP's interface--Verify that the interface for the LSP is allowed to use this backup tunnel. Look for the LSP's output interface in the protected i/fs field list.
- Backup tunnel has sufficient bandwidth--If you restricted the amount of bandwidth a backup tunnel can hold, verify that the backup tunnel has sufficient bandwidth to hold the LSPs that would use this backup tunnel if there is a failure. The bandwidth of an LSP is defined by the line **tunnel mpls traffic-eng bandwidth** at the headend of the LSP. To determine the available bandwidth on a backup tunnel, look at the "cfg" and "inuse" fields. If there is insufficient backup bandwidth to accommodate the LSPs that would use this backup tunnel in the event of a failure, create an additional backup tunnel or increase the backup bandwidth of the existing tunnel by using the **tunnel mpls traffic-eng bandwidth** command.

Note In order to determine how much bandwidth is sufficient, offline capacity planning may be required.

Backup tunnel has appropriate bandwidth type--If you restricted the type of LSPs (subpool or global pool) that can use this backup tunnel, verify that the LSP is the appropriate type for the backup tunnel. The type of the LSP is defined by the line **tunnel mpls traffic-eng bandwidth** at the headend of this LSP. If this line contains the word "sub pool", then it uses subpool bandwidth; otherwise, it uses global pool bandwidth. Verify that the type matches the type the backup tunnel can hold by looking in the output of the **tunnel mpls traffic-eng bandwidth**command.

If none of the verification actions described succeed, enable debug by entering the **debug ip rsvp fast-reroute** command and the **debug mpls traffic-eng fast-reroute** command on the router that is the headend of the backup tunnel. Then do the following:

a. Enter the **shutdown** command for the primary tunnel.

b. Enter the **no shutdown** command for the primary tunnel.

c. View the debug output.

Step 5 **show mpls traffic-eng fast-reroute database**

Enter the **clear ip rsvp hello instance counters** command to verify the following:

- MPLS TE FRR node protection has been enabled.
- A certain type of LSP can use a backup tunnel.

The following command output displays the LSPs that are protected:

Example:

```
Router# show mpls traffic-eng fast-reroute database
Tunnel head end item frr information:
Protected Tunnel   In-label   intf/label       FRR intf/label      Status
Tunnel10           Tun        Gi0/1/0:Untagged   Tu0:12304           ready
Prefix item frr information:
Prefix        Tunnel  In-label    Out intf/label  FRR intf/label   Status
```

```
10.0.0.11/32  Tu110   Tun hd      Gi0/1/0:Untagged  Tu0:12304        ready
LSP midpoint frr information:
LSP identifier       In-label   Out intf/label   FRR intf/label    Status
10.0.0.12 1 [459]    16         Gi0/1/1:17         Tu2000:19         ready
```

Note If Label Distribution Protocol (LDP) is not enabled, separate prefix items are not shown because all prefixes then use a single rewrite. To confirm that a particular IP prefix is FRR protected, even though it is not shown in this display, enter it within the **show mpls forwarding-table** *ip-address* **detail** command. The final line of the display will tell whether that prefix is protected.

Example:

```
Router# show mpls forwarding-table 10.0.0.11 32 detail

Local    Outgoing    Prefix         Bytes tag    Outgoing       Next Hop
tag      tag or VC   or Tunnel Id   switched     interface
Tun hd   Untagged    10.0.0.11/32   48 Gi0/1/0     point2point
          MAC/Encaps=4/8, MTU=1520, Tag Stack{22}
          48D18847 00016000
          No output feature configured
          Fast Reroute Protection via (Tu0, outgoing label 12304)
```

Step 6 **show ip rsvp reservation detail**

Following is sample output from the **show ip rsvp reservation detail** command entered at the headend of a primary LSP. Entering the command at the headend of the primary LSP shows, among other things, the status of FRR (that is, local protection) at each hop this LSP traverses. The per-hop information is collected in the Record Route Object (RRO) that travels with the Resv message from the tail to the head.

Example:

```
Router# show ip rsvp reservation detail
Reservation:
  Tun Dest:   10.1.1.1  Tun ID: 1  Ext Tun ID: 10.1.1.1
  Tun Sender: 10.1.1.1  LSP ID: 104
  Next Hop: 10.1.1.2 on Gi1/0/2
  Label: 18 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  Average Bitrate is 0 bits/sec, Maximum Burst is 1K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  RRO:
    10.1.1.1/32, Flags:0x1 (Local Prot Avail/to NHOP)
      Label subobject: Flags 0x1, C-Type 1, Label 18
    10.1.1.1/32, Flags:0x0 (Local Prot Avail/In Use/Has BW/to NHOP)
      Label subobject: Flags 0x1, C-Type 1, Label 16
    10.1.1.2/32, Flags:0x0 (No Local Protection)
      Label subobject: Flags 0x1, C-Type 1, Label 0
  Resv ID handle: CD000404.
  Policy: Accepted. Policy source(s): MPLS/TE
```

Notice the following about the primary LSP:

- It has protection that uses an NHOP backup tunnel at its first hop.
- It has protection and is actively using an NHOP backup tunnel at its second hop.
- It has no local protection at its third hop.

The RRO display shows the following information for each hop:

- Whether local protection is available (that is, whether the LSP has selected a backup tunnel)

- Whether local protection is in use (that is, whether the LSP is using its selected backup tunnel)
- Whether the selected backup tunnel is an NHOP or NNHOP backup tunnel
- Whether the backup tunnel used at this hop provides bandwidth protection

Step 7 **show ip rsvp hello**

Use this command to display hello status and statistics for FRR, reroute (hello state timer), and graceful restart. Following is sample output:

Example:

```
Router# show ip rsvp hello

Hello:
 RSVP Hello for Fast-Reroute/Reroute: Enabled
  Statistics: Disabled
 BFD for Fast-Reroute/Reroute: Enabled
 RSVP Hello for Graceful Restart: Disabled
```

Step 8 **show ip rsvp interface detail**

Use this command to display the interface configuration for Hello. Following is sample output:

Example:

```
Router# show ip rsvp interface detail

Gi2/1/1:
 RSVP: Enabled
 Interface State: Up
 Bandwidth:
  Curr allocated: 0 bits/sec
  Max. allowed (total): 0 bits/sec
  Max. allowed (per flow): 0 bits/sec
  Max. allowed for LSP tunnels using sub-pools (pool 1): 0 bits/sec
  Set aside by policy (total): 0 bits/sec
 Signalling:
  DSCP value used in RSVP msgs: 0x3F
  Number of refresh intervals to enforce blockade state: 4
 Authentication: disabled
  Key chain: <none>
  Type:  md5
  Window size: 1
  Challenge:  disabled
 FRR Extension:
  Backup Path: Configured (or "Not Configured")
 BFD Extension:
  State: Disabled
  Interval: Not Configured
 RSVP Hello Extension:
  State: Disabled
  Refresh Interval: FRR: 200  , Reroute: 2000
  Missed Acks:      FRR: 4    , Reroute: 4
  DSCP in HELLOs:   FRR: 0x30 , Reroute: 0x30
```

Step 9 **show ip rsvp hello bfd nbr**

Use this command to display information about all MPLS traffic engineering link and node protected neighbors that use the BFD protocol. Following is sample output. The command output is the same as the **show ip rsvp hello bfd nbr summary** command output.

Example:

```
Router# show ip rsvp hello bfd nbr

Client  Neighbor    I/F     State   LostCnt   LSPs
FRR     10.0.0.6    Gi2/1/1  Up      0         1
```

Step 10 **show ip rsvp hello bfd nbr detail**

Use this command to display detailed information about all MPLS traffic engineering link and node protected neighbors that use the BFD protocol:

Example:

```
Router# show ip rsvp hello bfd nbr detail

 Hello Client Neighbors
 Remote addr 10.0.0.6, Local addr  10.0.0.7
  Type: Active
  I/F: Gi2/1/1
  State: Up (for 00:09:41)
  Clients: FRR
  LSPs protecting: 1 (frr: 1, hst upstream: 0 hst downstream: 0)
  Communication with neighbor lost: 0
```

Step 11 **show ip rsvp hello bfd nbr summary**

Use this command to display summarized information about all MPLS traffic engineering link and node protected neighbors that use the BFD protocol. The command output is the same as the **show ip rsvp hello bfd nbr summary** command output.

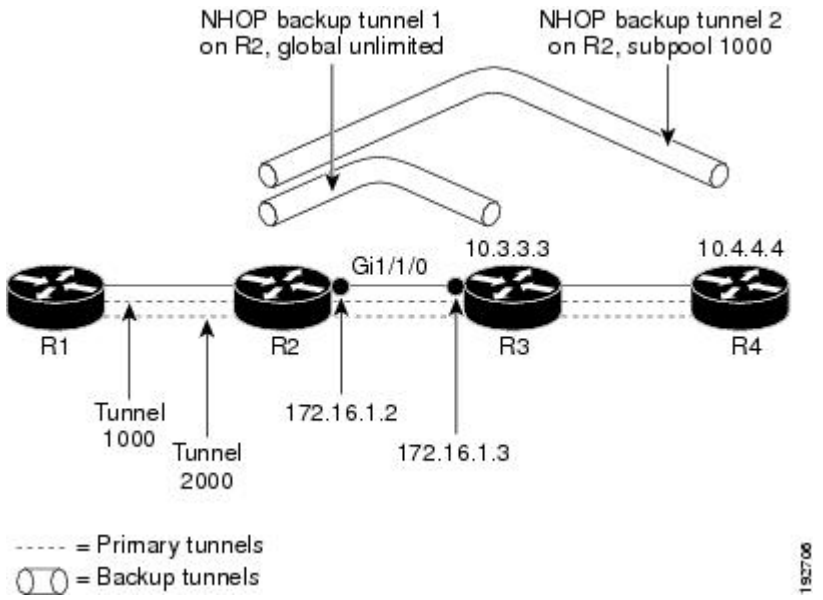
Example:

```
Router# show ip rsvp hello bfd nbr summary

Client  Neighbor    I/F     State  LostCnt  LSPs
FRR     10.0.0.6   Gi2/1/1  Up     0        1
```

Configuration Examples for MPLS Traffic Engineering BFD-triggered Fast Reroute

The examples in this section are based on the backup tunnels shown in the figure below.

Figure 179: Backup Tunnels



Example Enabling BFD Support on the Router

The following example enables the BFD protocol on the router:

```
Router(config)# ip rsvp signalling hello bfd
```

Example Enabling Fast Reroute on LSPs

On router R1 in the figure above, enter interface configuration mode for each tunnel to be protected (Tunnel 1000 and Tunnel 2000). Enable these tunnels to use a backup tunnel in case of a link or node failure along their paths.

Tunnel 1000 will use ten units of bandwidth from the subpool.

Tunnel 2000 will use five units of bandwidth from the global pool. The “bandwidth protection desired” bit and the “node protection desired bit” have been set by specifying **bw-prot** and **node-prot**, respectively, in the **tunnel mpls traffic-eng fast-reroute** command.

```
Router(config)# interface tunnel 1000
Router(config-if)# tunnel mpls traffic-eng fast-reroute
Router(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 10
Router(config)# interface tunnel 2000
Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect node-protect
Router(config-if)# tunnel mpls traffic-eng bandwidth 5
```

Example Creating a Backup Tunnel to the Next Hop

On router R2 in the figure above, create an NHOP backup tunnel to R3. This backup tunnel should avoid using the link 10.1.1.2.

```
Router(config)# ip explicit-path name avoid-protected-link
Router(cfg-ip-expl-path)# exclude-address 10.1.1.2

Explicit Path name avoid-protected-link:
____1: exclude-address 10.1.1.2
Router(cfg-ip_expl-path)# exit

Router(config)# interface tunnel 1

Router(config-if)# ip unnumbered loopback 0

Router(config-if)# tunnel destination 10.3.3.3
Router(config-if)# tunnel mode mpls traffic-eng

Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit avoid-protected-link
```

Example Creating an NNHOP Backup Tunnel

On router R2 in the figure above, create an NNHOP backup tunnel to R4. This backup tunnel should avoid R3.

```
Router(config)# ip explicit-path name avoid-protected-node

Router(cfg-ip-expl-path)# exclude-address 10.3.3.3

Explicit Path name avoid-protected-node:
____1: exclude-address 10.3.3.3
Router(cfg-ip_expl-path)# end

Router(config)# interface tunnel2

Router(config-if)# ip unnumbered loopback0

Router(config-if)# tunnel destination 10.4.4.4

Router(config-if)# tunnel mode mpls traffic-eng0

Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit avoid-protected-node
```

Example Assigning Backup Tunnels to a Protected Interface

On router R2 in the figure above, both backup tunnels are associated with interface Gigabit Ethernet 0/1/0:

```
Router(config)# interface Gi0/1/0

Router(config-if)# mpls traffic-eng backup-path tunnel 1

Router(config-if)# mpls traffic-eng backup-path tunnel 2
```

Example Enabling BFD on the Protected Interface

In the figure above, BFD is enabled on interface Gigabit Ethernet 2/1/1:

```
Router(config)# interface Gi2/1/1

Router(config-if)# ip rsvp signalling hello bfd

Router(config-if)# bfd interval 100 min_rx 100 multiplier 4
```

Example Associating Backup Bandwidth and Pool Type with Backup Tunnels

In the figure above, backup tunnel 1 is to be used only by LSPs that take their bandwidth from the global pool. It does not provide bandwidth protection. Backup tunnel 2 is to be used only by LSPs that take their bandwidth from the subpool. Backup tunnel 2 provides bandwidth protection for up to 1000 units.

```
Router(config)# interface tunnel 1

Router(config-if)# tunnel mpls traffic-eng backup-bw global-pool Unlimited

Router(config)# interface tunnel 2

Router(config-if)# tunnel mpls traffic-eng backup-bw sub-pool 1000
```

Example Configuring Backup Bandwidth Protection

Note This global configuration is required only to change the backup protection preemption algorithm from minimize the number of LSPs that are demoted to minimize the amount of bandwidth that is wasted.

```
Router(config-if)# tunnel mpls traffic-eng fast-reroute bw-protect
Router(config)# mpls traffic-eng fast-reroute backup-prot-preemption optimize-bw
```

Additional References

Related Documents

Related Topic	Document Title
Link and node protection	MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection)
Multiprotocol Label Switching commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Bidirectional Forwarding Direction configuration information	"Bidirectional Forwarding Detection" chapter in the *Cisco IOS IP Routing Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering BFD-triggered Fast Reroute

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 184: Feature Information for MPLS Traffic Engineering: BFD-triggered Fast Reroute

Feature Name	Releases	Feature Information
MPLS Traffic Engineering: BFD-triggered Fast Reroute	Cisco IOS XE Release 2.3	The MPLS Traffic Engineering: BFD-triggered Fast Reroute feature allows you to obtain link and node protection by using the Bidirectional Forwarding Detection (BFD) protocol to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. In Cisco IOS XE Release 2.3, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature: **clear ip rsvp hello bfd**, **ip rsvp signalling hello bfd** (configuration), **ip rsvp signalling hello bfd** (interface), **show ip rsvp hello**, **show ip rsvp hello bfd nbr**, **show ip rsvp hello bfd nbr detail**, **show ip rsvp hello bfd nbr summary**, and **show ip rsvp interface detail**.

Glossary

backup bandwidth --The usage of NHOP and NNHOP backup tunnels to provide bandwidth protection for rerouted LSPs.

backup tunnel --An MPLS TE tunnel used to protect other (primary) tunnels' traffic when a link or node failure occurs.

bandwidth --The available traffic capacity of a link.

fast reroute --Procedures that enable temporary routing around a failed link or node while a new LSP is being established at the headend.

global pool --The total bandwidth allocated to an MPLS traffic engineering link or node.

headend --The router that originates and maintains a given LSP. This is the first router in the LSP's path.

hop --Passage of a data packet between two network nodes (for example, between two routers).

instance --A Hello instance implements the RSVP Hello extensions for a given router interface address and remote IP address. Active Hello instances periodically send Hello Request messages, expecting Hello ACK messages in response. If the expected Ack message is not received, the active Hello instance declares that the neighbor (remote IP address) is unreachable (that is, it is lost). This can cause LSPs crossing this neighbor to be fast rerouted.

interface --A network connection.

link --A point-to-point connection between adjacent nodes. There can be more than one link between adjacent nodes. A network communications channel consisting of a circuit or transmission path and all related equipment between a sender and a receiver. Sometimes referred to as a line or a transmission link.

LSP --label-switched path. A configured connection between two routers, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

MPLS --Multiprotocol Label Switching. Packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

NHOP --next hop. The next downstream node along an LSP's path.

NHOP backup tunne l--next-hop backup tunnel. Backup tunnel terminating at the LSP's next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link, and is used to protect primary LSPs that were using this link before the failure.

NNHOP --next-next hop. The node after the next downstream node along an LSP's path.

NNHOP backup tunnel --next-next-hop backup tunnel. Backup tunnel terminating at the LSP's next-next hop beyond the point of failure, and originating at the hop immediately upstream of the point of failure. It bypasses a failed link or node, and is used to protect primary LSPs that were using this link or node before the failure.

node --Endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network. Computers on a network, or any endpoint or a junction common to two or more lines in a network. Nodes can be processors, controllers, or workstations.

primary LSP --The last LSP originally signaled over the protected interface before the failure. The LSP before the failure.

primary tunnel --Tunnel whose LSP may be fast rerouted if there is a failure. Backup tunnels cannot be primary tunnels.

protected interface --An interface that has one or more backup tunnels associated with it.

redundancy --The duplication of devices, services, or connections so that, in the event of a failure, the redundant devices, services, or connections can perform the work of those that failed.

RSVP --Resource Reservation Protocol. An IETF protocol used for signaling requests (setting up reservations) for Internet services by a customer before that customer is permitted to transmit data over that portion of the network.

state --Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

subpool --The more restrictive bandwidth in an MPLS traffic engineering link or node. The subpool is a portion of the link or node's overall global pool bandwidth.

tailend --The router upon which an LSP is terminated. This is the last router in the LSP's path.

tunnel --Secure communications path between two peers, such as two routers.

CHAPTER 105

MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

The MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion feature provides a means to exclude a link or node from the path for a Multiprotocol Label Switching (MPLS) TE label switched path (LSP).

The feature is enabled through the **ip explicit-path** command that allows you to create an IP explicit path and enter a configuration submode for specifying the path. The feature adds to the submode commands the **exclude-address** command for specifying addresses to exclude from the path.

If the excluded address for an MPLS TE LSP identifies a flooded link, the constraint-based shortest path first (CSPF) routing algorithm does not consider that link when computing paths for the LSP. If the excluded address specifies a flooded MPLS TE router ID, the CSPF routing algorithm does not allow paths for the LSP to traverse the node identified by the router ID.

Prerequisites for MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

Your network must support the following Cisco IOS XE features in order to support IP explicit address exclusion:

- MPLS
- IP Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)

Restrictions for MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

MPLS TE will accept an IP explicit path comprised of either all excluded addresses configured by the **exclude-address** command or all included addresses configured by the **next-address**command, but not a combination of both.

Information About MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

MPLS Traffic Engineering

MPLS is an Internet Engineering Task Force (IETF)-specified framework that provides for the efficient designation, routing, forwarding, and switching of traffic flows through the network.MPLS is a method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

Traffic engineering (TE) is the process of adjusting bandwidth allocations to ensure that enough is left for high-priority traffic.

In MPLS TE, the upstream router creates a network tunnel for a particular traffic stream, then fixes the bandwidth available for that tunnel.

Cisco Express Forwarding

Cisco Express Forwarding is an advanced, Layer 3 switching technology inside a router. It defines the fastest method by which a Cisco router forwards packets from ingress to egress interfaces. The **ip cef** command enables Cisco Express Forwarding globally, and the **ip route-cache cef**command enables Cisco Express Forwarding on an interface.

How to Configure MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

Configuring IP Explicit Address Exclusion

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip explicit-path** {**name** *path-name* | **identifier** *number*} [**enable** | **disable**]
4. **exclude-address** *ip-address*

5. **exit**
6. **exit**
7. **show ip explicit-path**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip explicit-path** {**name** *path-name* | **identifier** *number*} [**enable** | **disable**] **Example:** `Router(config)# ip explicit-path name OmitR12`	Specifies the name or number of the explicit path, and enables the path, and enters explicit-path configuration mode.
Step 4	**exclude-address** *ip-address* **Example:** `Router(cfg-ip-expl-path)# exclude-address 10.12.12.12`	Excludes the specified link or node from consideration by the constraint-based SPF. • The *ip-address* is a link address or the router ID for a node.
Step 5	**exit** **Example:** `Router(cfg-ip-expl-path)# exit`	Exits from explicit-path configuration mode, and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits from global configuration mode, and returns to privileged EXEC mode.
Step 7	**show ip explicit-path** **Example:** `Router# show ip explicit-path`	Displays information about configured IP explicit paths.

Configuring an MPLS Traffic Engineering Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered loopback0**
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | **ID** *path-number*}} [*lockdown*]
9. **exit**
10. **exit**
11. **show mpls traffic eng tunnels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel11`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered loopback0** **Example:** `Router(config-if)# ip unnumbered loopback0`	Assigns the tunnel interface an IP address. • An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 10.11.11.11`	Specifies the destination for a tunnel. • The destination of the tunnel must be the MPLS traffic engineering router ID of the destination device.

	Command or Action	Purpose
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 100`	Configures the bandwidth for the MPLS traffic engineering tunnel.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | **ID** *path-number*}} [*lockdown*] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 2 dynamic`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. • A dynamic path is used if an explicit path is unavailable. **Note** To configure a path option that specifies an exclude address, specify the **explicit** keyword (not the **dynamic** keyword) and specify an IP explicit path configured according to the steps in the "Configuring IP Explicit Address Exclusion, on page 2246" section.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits from interface configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.
Step 11	**show mpls traffic eng tunnels** **Example:** `Router# show mpls traffic eng tunnels`	Shows information about tunnels, including the current tunnel path if a tunnel is operational. • By viewing the command output, you can determine the path that was used to build a tunnel. If you entered the **exclude-address** command, the specified link or node should not be listed.

Configuration Examples for MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

Example Configuring IP Explicit Address Exclusion

The following example shows how to configure an MPLS TE tunnel with two path options: a preferred explicit path with an excluded address and a backup dynamic path.

Configure the IP explicit path named OmitR12, which excludes the router with router ID 10.12.12.12:

```
ip explicit-path name OmitR12
exclude-address 10.12.12.12
   Explicit Path name OmitR12:
   1: exclude-address 10.12.12.12
exit
```

To verify the configuration of the explicit path, use the **show ip explicit-path** command.

```
show ip explicit-paths name OmitR12
PATH OmitR12 (loose source route, path complete, generation 3)
 1: exclude-address 10.12.12.12
```

Note You must know the router IDs for LSRs (nodes) in the network; in this example, that 10.12.12.12 is a router ID. Otherwise, it will not be apparent whether the specified address is the IP address of a link or a router ID.

Example Configuring an MPLS Traffic Engineering Tunnel

The following example configures Tunnel11 with its two options, where the preferred path option is the IP explicit path OmitR2:

```
interface tunnel11
ip unnumbered loopback0
tunnel destination 10.11.11.11
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng path-option 1 explicit name OmitR12
tunnel mpls traffic-eng path-option 2 dynamic
```

Note There are additional commands for configuring properties for TE tunnels such as bandwidth and priority. For descriptions of those commands, refer to the Cisco IOS Multiprotocol Label Switching Command Reference.

Additional References

Related Documents

Related Topic	Document Title
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
MPLS configuration information	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 185: Feature Information for MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion

Feature Name	Releases	Feature Configuration Information
MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion	Cisco IOS XE Release 2.3	The MPLS Traffic Engineering (TE)--IP Explicit Address Exclusion feature provides a means to exclude a link or node from the path for Multiprotocol Label Switching (MPLS) TE label switched path (LSP). This feature was integrated into Cisco IOS XE Release 2.3. The following command was introduced by this feature: **exclude-address**.

Glossary

Cisco Express Forwarding --A means for accelerating the forwarding of packets within a router, by storing route lookup information in several data structures instead of in a route cache.

IP explicit path --A list of IP addresses, each representing a node or link in the explicit path.

link --Network communications channel consisting of a circuit or transmission path and all related equipment between a sender and a receiver. Sometimes referred to as a line or a transmission link.

MPLS --Multiprotocol Label Switching. Switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

node --Endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network.

CHAPTER 106

MPLS Traffic Engineering Shared Risk Link Groups

The MPLS Traffic Engineering: Shared Risk Link Groups feature enhances backup tunnel path selection so that a backup tunnel avoids using links that are in the same Shared Risk Link Group (SRLG) as interfaces the backup tunnel is protecting.

SRLGs refer to situations where links in a network share a common fiber (or a common physical attribute). If one link fails, other links in the group may fail too. Links in the group have a shared risk.

Prerequisites for MPLS Traffic Engineering Shared Risk Link Groups

- You must configure Fast Reroutable tunnels.
- You must enable the autotunnel backup.

Restrictions for MPLS Traffic Engineering Shared Risk Link Groups

- The backup tunnel must be within a single area.
- Manually created backup tunnels do not automatically avoid SRLGs of protected interfaces.
- A primary tunnel cannot be specified to avoid links belonging to specified SRLGS.

Information About MPLS Traffic Engineering Shared Risk Link Groups

MPLS Traffic Engineering Brief Overview

Multiprotocol Label Switching (MPLS) is an Internet Engineering Task Force (IETF)-specified framework that provides for the efficient designation, routing, forwarding, and switching of traffic flows through the network.

Traffic engineering (TE) is the process of adjusting bandwidth allocations to ensure that enough is left for high-priority traffic.

In MPLS TE, the upstream router creates a network tunnel for a particular traffic stream, then fixes the bandwidth available for that tunnel.

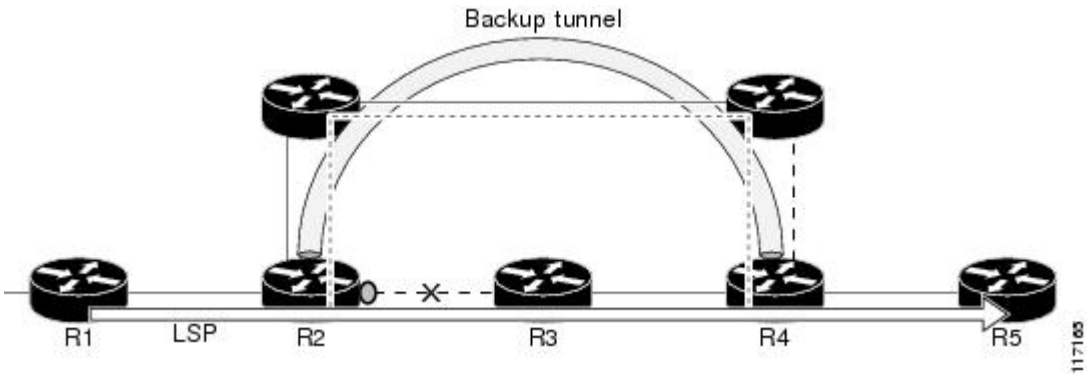
MPLS Traffic Engineering Shared Risk Link Groups

SRLGs refer to situations where links in a network share a common fiber (or a common physical attribute). If one link fails, other links in the group may fail too. Links in the group have a shared risk.

Backup tunnels should avoid using links in the same SRLG as interfaces they are protecting. Otherwise, when the protected link fails the backup tunnel fails too.

The figure below shows a primary label-switched path (LSP) from router R1 to router R5. The LSP protects against the failure of the R2-R3 link at R2 via a backup tunnel to R4. If the R2-R3 link fails, link protection reroutes the LSP along the backup tunnel. However, the R2-R3 link and one of the backup tunnel links are in the same SRLG. So if the R2-R3 link fails, the backup tunnel may fail too.

Figure 180: Backup Tunnel in the Same SRLG as the Interface It Is Protecting



The MPLS TE SRLG feature enhances backup tunnel path selection so a backup tunnel can avoid using links that are in the same SRLG as the interfaces it is protecting.

There are two ways for a backup tunnel to avoid the SRLGs of its protected interface:

- The router does not create the backup tunnel unless it avoids SRLGs of the protected interface.
- The router *tries* to avoid SRLGs of the protected interface, but if that is not possible the router creates the backup tunnel anyway. In this case there are two explicit paths. The first explicit path *tries* to avoid

the SRLGs of the protected interface. If that does not work, the backup tunnel uses the second path (which ignores SRLGs).

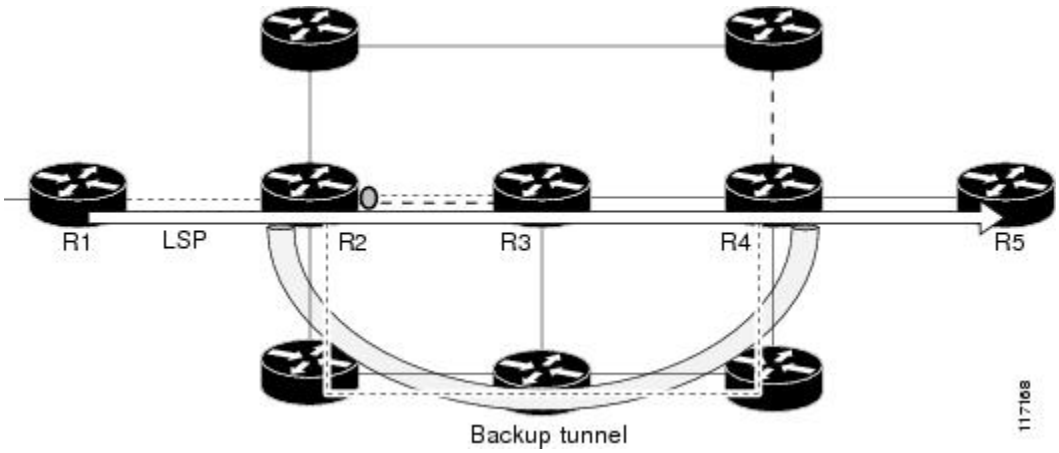
Note Only backup tunnels that routers create automatically (called autotunnel backup) can avoid SRLGs of protected interfaces.

To activate the MPLS TE SRLG feature, you must do the following:

- Configure the SRLG membership of each link that has a shared risk with another link.
- Configure the routers to automatically create backup tunnels that avoid SRLGs of the protected interfaces.

Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) flood the SRLG membership information (including other TE link attributes such as bandwidth availability and affinity) so that all routers in the network have the SRLG information for each link. With this topology information, routers can compute backup tunnel paths that exclude links having SRLGs in common with their protected interfaces. As shown in the figure below, the backup tunnel avoids the link between R2 and R3, which shares an SRLG with the protected interface.

Figure 181: Backup Tunnel That Avoids SRLG of Protected Interface

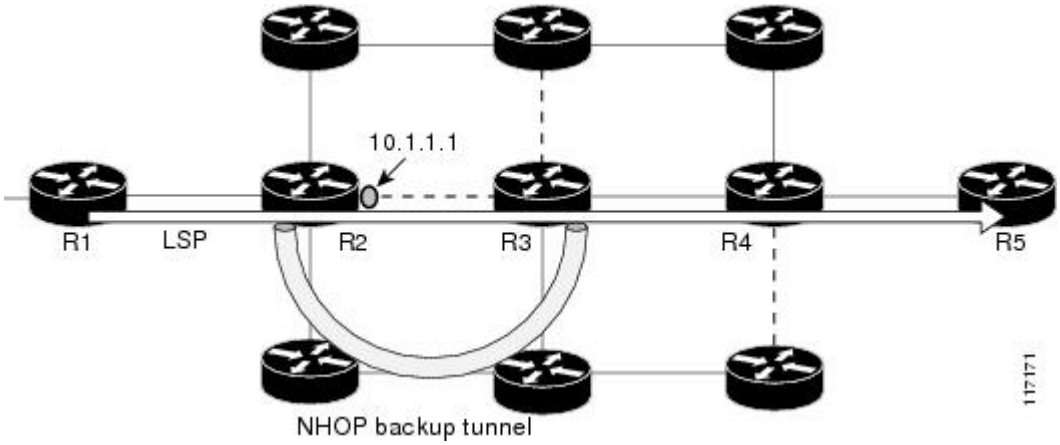


Fast Reroute Protection for MPLS TE SRLGs

Fast Reroute (FRR) protects MPLS TE LSPs from link and node failures by locally repairing the LSPs at the point of failure. This protection allows data to continue to flow on LSPs while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

Backup tunnels that bypass only a single link of the LSP's path provide link protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure. The figure below illustrates an NHOP backup tunnel.

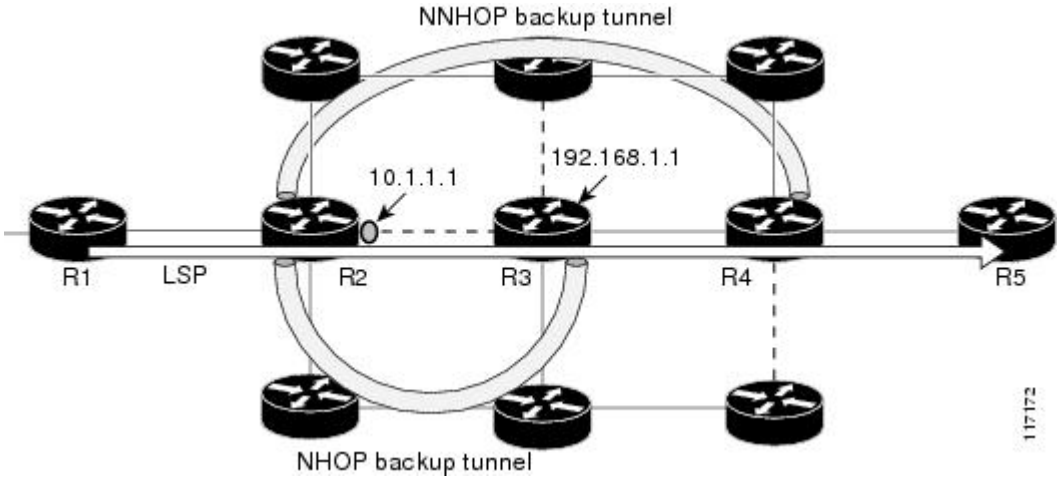
Figure 182: NHOP Backup Tunnel



FRR provides node protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called next-next-hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSP paths, thereby bypassing the next-hop node. They protect LSPs if a node along their path fails by enabling the node upstream of the failure to reroute the LSPs and their traffic around the failed node to the next-next hop. FRR supports the use of Resource Reservation Protocol (RSVP) hellos to accelerate the detection of node failures. NNHOP backup tunnels also provide protection from link failures, because they bypass the failed link and the node.

The figure below illustrates an NNHOP backup tunnel.

Figure 183: NNHOP Backup Tunnel



Autotunnel Backup for MPLS TE SRLGs

Autotunnel backup is the ability of routers to create backup tunnels automatically. Therefore, you do not need to preconfigure each backup tunnel and then assign the backup tunnel to the protected interface. Only automatically created backup tunnels can avoid SRLGs or their protected interfaces.

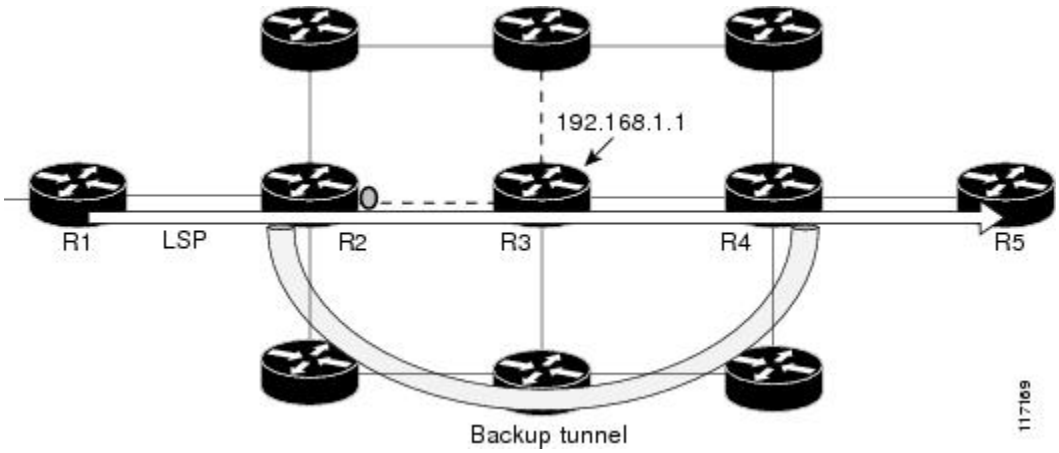
For information about backup tunnels, see the Fast Reroute Protection for MPLS TE SRLGs, on page 2255.

For detailed information about autotunnel backup and how you can change the default command values, see MPLS Traffic Engineering (TE)--AutoTunnel Primary and Backup.

To globally activate the autotunnel backup feature, enter the **mpls traffic-eng auto-tunnel backup** command.

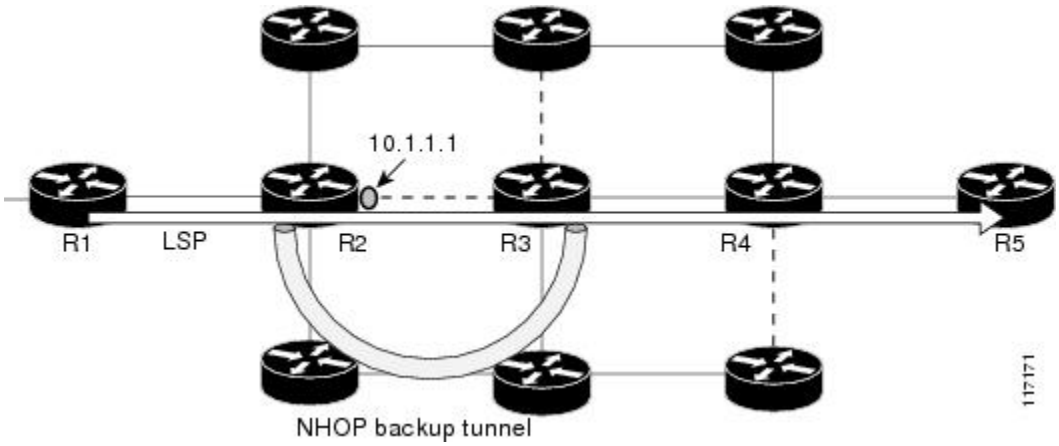
The figure below illustrates an NNHOP automatically generated backup tunnel that excludes the router 192.168.1.1 and terminates at router R4. The backup tunnel must avoid touching any links of 192.168.1.1.

Figure 184: Autotunnel Backup for NNHOP



The figure below illustrates an NHOP automatically generated backup tunnel that terminates at router R3 and avoids the link 10.1.1.1, not the entire node.

Figure 185: Autotunnel Backup for NHOP



Note NNHOP excludes the router ID (the entire router must be excluded; that is, no link of the router can be included in the backup tunnel's path). NHOP excludes only the link when the backup tunnel's path is computed.

How to Configure MPLS Traffic Engineering Shared Risk Link Groups

Configuring MPLS TE SRLG Membership of Each Link That Has a Shared Risk with Another Link

Perform the following task to configure MPLS TE SRLG membership of each link that has a shared risk with another link. Configuring SRLG membership enhances backup tunnel path selection so that a backup tunnel avoids using links that are in the same SRLG as interfaces the backup tunnel is protecting.

Enter the commands on the physical interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls traffic-eng srlg** [*number*] [
5. **mpls traffic-eng srlg end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router(config)# interface pos 1/1/1`	Specifies an interface and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information. The slash (/) is required.

	Command or Action	Purpose
Step 4	**mpls traffic-eng srlg** [*number*] [**Example:** `Router(config-if)# mpls traffic-eng srlg 5`	Configures the SRLG membership of a link (interface). • The *number* argument is an SRLG identifier. Valid values are 0 to 4,294,967,295. **Note** To make the link a member of multiple SRLGs, enter the **mpls traffic-eng srlg** command multiple times.
Step 5	**mpls traffic-eng srlg end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuring the Routers That Automatically Create Backup Tunnels to Avoid MPLS TE SRLGs

Perform the following task to configure routers that automatically create backup tunnels to avoid MPLS TE SRLGs of their protected interfaces. Backup tunnels provide link protection by rerouting traffic to the next hop bypassing failed links or in this instance by avoiding SRLGs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel backup srlg exclude** [**force** | **preferred**]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng auto-tunnel backup srlg exclude** [**force**	**preferred**] **Example:**	Specifies that autocreated backup tunnels should avoid SRLGs of its protected interface. • The **force** keyword forces the backup tunnel to avoid SRLGs of its protected interface or interfaces.

	Command or Action	Purpose
	`Router(config)# mpls traffic-eng auto-tunnel backup srlg exclude force`	• The **preferred** keyword causes the backup tunnel to *try* to avoid SRLGs of its protected interface or interfaces, but the backup tunnel can be created if SRLGs cannot be avoided.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Verifying the MPLS Traffic Engineering Shared Risk Link Groups Configuration

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **show mpls traffic-eng link-management interfaces***interface slot/port*
4. **show mpls traffic-eng topology**
5. **show mpls traffic-eng topology srlg**
6. **show mpls traffic-eng topology brief**
7. **show mpls traffic-eng link-management advertisements**
8. **show ip rsvp fast-reroute**
9. **mpls traffic-eng auto-tunnel backup srlg exclude force**
10. **show ip explicit-paths**
11. **show mpls traffic-eng tunnels tunnel** *num*
12. **mpls traffic-eng auto-tunnel backup srlg exclude preferred**
13. **show ip explicit-paths**
14. **show ip rsvp fast-reroute**
15. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password, if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show running-config**

Use the following commands to configure the SRLG membership of the interface pos 1/3/1 and to verify that the configuration is as expected. For example:

Example:

```
Router# configure terminal
Router(config)# interface pos 1/3/1
Router(config-if)# mpls traffic-eng srlg 1
Router(config-if)# mpls traffic-eng srlg 2
Router(config-if)# end
Router# show running-config

interface POS 1/3/1
 ip address 10.0.0.33 255.255.255.255
 no ip directed-broadcast
 ip router isis
 encapsulation ppp
 no ip mroute-cache
 mpls traffic-eng tunnels
 mpls traffic-eng backup-path Tunnel5000
 mpls traffic-eng srlg 1
 mpls traffic-eng srlg 2
 tag-switching ip
 crc 32
 clock source internal
 pos ais-shut
 pos report rdool
 pos report lais
 pos report lrdi
 pos report pais
 pos report prdi
 pos report sd-ber
 isis circuit-type level-2-only
 ip rsvp bandwidth 20000 20000 sub-pool 5000
```

This verifies that the Packet over SONET (POS) interface pos 1/3/1 is associated that SRLG 1 and SRLG 2.

Step 3 **show mpls traffic-eng link-management interfaces***interface slot/port*

Use this command to show the SRLG membership configured on interface pos 1/3/1. For example:

Example:

```
Router# show mpls traffic-eng link-management interfaces pos 1/3/1
System Information::
    Links Count:         11
Link ID:: PO1/3/1 (10.0.0.33)
    Link Status:
      SRLGs:               1 2
      Physical Bandwidth:  2488000 kbits/sec
      Max Res Global BW:   20000 kbits/sec (reserved:0% in, 0% out)
      Max Res Sub BW:      5000 kbits/sec (reserved:0% in, 0% out)
      MPLS TE Link State:  MPLS TE on, RSVP on, admin-up, flooded
      Inbound Admission:   allow-all
      Outbound Admission:  allow-if-room
      Admin. Weight:       10 (IGP)
      IGP Neighbor Count:  1
      IGP Neighbor:        ID 0000.0000.0004.00, IP 10.0.0.34 (Up)
    Flooding Status for each configured area [1]:
      IGP Area[1]: isis  level-2: flooded
```

Step 4 **show mpls traffic-eng topology**

Use this command to show the SRLG link membership flooded via the Interior Gateway Protocol (IGP). For example:

Example:

```
Router# show mpls traffic-eng topology

My_System_id:0000.0000.0003.00 (isis  level-2)
Signalling error holddown:10 sec Global Link Generation 9
IGP Id:0000.0000.0003.00, MPLS TE Id:10.0.3.1 Router Node  (isis
level-2)
      link[0]:Point-to-Point, Nbr IGP Id:0000.0000.0004.00,
nbr_node_id:2, gen:9
          frag_id 0, Intf Address:10.0.0.33, Nbr Intf Address:10.0.0.34
          TE metric:10, IGP metric:10, attribute_flags:0x0
          SRLGs:1 2
          physical_bw:2488000 (kbps), max_reservable_bw_global:20000
(kbps)
          max_reservable_bw_sub:5000 (kbps)
                                Global Pool       Sub Pool
               Total Allocated  Reservable       Reservable
               BW (kbps)        BW (kbps)        BW (kbps)
               ---------------  -----------      ----------
        bw[0]:           0           20000            5000
        bw[1]:           0           20000            5000
        bw[2]:           0           20000            5000
        bw[3]:           0           20000            5000
        bw[4]:           0           20000            5000
        bw[5]:           0           20000            5000
```

Step 5 **show mpls traffic-eng topology srlg**

Use this command to display all the links in the network that are members of a given SRLG. For example:

Example:

```
Router# show mpls traffic-eng topology srlg
MPLS TE Id:0000.0000.0003.00 (isis  level-2)
  SRLG:1
     10.0.0.33
  SRLG:2
     10.0.0.33
```

The following command shows that there are two links in SRLG 1:

Example:

```
Router# show mpls traffic-eng topology srlg
MPLS TE Id:0000.0000.0003.00 (isis  level-2)
  SRLG:1
     10.0.0.33
     10.0.0.49
```

Step 6 **show mpls traffic-eng topology brief**

Use this command to display brief topology information:

Example:

```
Router# show mpls traffic-eng topology brief
My_System_id:0000.0000.0003.00 (isis  level-2)
Signalling error holddown:10 sec Global Link Generation 9
IGP Id:0000.0000.0003.00, MPLS TE Id:10.0.3.1 Router Node  (isis
level-2)
      link[0]:Point-to-Point, Nbr IGP Id:0000.0000.0004.00,
nbr_node_id:2, gen:9
          frag_id 0, Intf Address:10.0.0.33, Nbr Intf Address:10.0.0.34
```

```
          TE metric:10, IGP metric:10, attribute_flags:0x0
          SRLGs:1 2
```

Step 7 **show mpls traffic-eng link-management advertisements**

Use this command to show local link information that MPLS TE link management is currently flooding into the global TE topology. For example:

Example:

```
Router# show mpls traffic-eng link-management advertisements

Flooding Status:      ready
Configured Areas:     1
IGP Area[1] ID::  isis  level-2
  System Information::
    Flooding Protocol:    ISIS
  Header Information::
    IGP System ID:        0000.0000.0003.00
    MPLS TE Router ID:    10.0.3.1
    Flooded Links:        2
  Link ID::  0
    Link Subnet Type:     Point-to-Point
    Link IP Address:      10.0.0.49
    IGP Neighbor:         ID 0000.0000.0007.00, IP 10.0.0.50
    TE metric:            80000
    IGP metric:           80000
    SRLGs:                None
    Physical Bandwidth:   622000 kbits/sec
    Res. Global BW:       20000 kbits/sec
    Res. Sub BW:          5000 kbits/sec
    Downstream::
                                Global Pool   Sub Pool
                                -----------   --------------
      Reservable Bandwidth[0]:  20000         5000 kbits/sec
      Reservable Bandwidth[1]:  20000         5000 kbits/sec
      Reservable Bandwidth[2]:  20000         5000 kbits/sec
      Reservable Bandwidth[3]:  20000         5000 kbits/sec
      Reservable Bandwidth[4]:  20000         5000 kbits/sec
      Reservable Bandwidth[5]:  20000         5000 kbits/sec
      Reservable Bandwidth[6]:  20000         5000 kbits/sec
      Reservable Bandwidth[7]:  20000         5000 kbits/sec
    Attribute Flags:      0x00000000
  Link ID::  1
    Link Subnet Type:     Point-to-Point
    Link IP Address:      10.0.0.33
    IGP Neighbor:         ID 0000.0000.0004.00, IP 10.0.0.34
    TE metric:            10
    IGP metric:           10
    SRLGs:                1
    Physical Bandwidth:   2488000 kbits/sec
    Res. Global BW:       20000 kbits/sec
    Res. Sub BW:          5000 kbits/sec
    Downstream::
                                Global Pool   Sub Pool
                                -----------   --------------
      Reservable Bandwidth[0]:  20000         5000 kbits/sec
      Reservable Bandwidth[1]:  20000         5000 kbits/sec
      Reservable Bandwidth[2]:  20000         5000 kbits/sec
      Reservable Bandwidth[3]:  20000         5000 kbits/sec
      Reservable Bandwidth[4]:  20000         5000 kbits/sec
      Reservable Bandwidth[5]:  20000         5000 kbits/sec
      Reservable Bandwidth[6]:  20000         5000 kbits/sec
```

```
      Reservable Bandwidth[7]:  20000         5000 kbits/sec
    Attribute Flags:      0x00000000
```

Step 8 **show ip rsvp fast-reroute**

Use this command to show that the primary tunnel is going over Pos1/3/1 on R3, on which SLRG 1 is configured. For example:

Example:

```
Router# show ip rsvp fast-reroute
Primary    Protect      BW      Backup
Tunnel     I/F        BPS:Type  Tunnel:Label  State   Level   Type
-------    -------    --------  ------------  -----   -----   ----
R3-PRP_t0  PO1/3/1 0:G  None      None          None    None    None
```

Step 9 **mpls traffic-eng auto-tunnel backup srlg exclude force**

Use the following commands to configure autotunnel backup with the **force** keyword. For example:

Example:

```
Router# configure terminal
Router(config)# mpls traffic-eng auto-tunnel backup
Router(config)# mpls traffic-eng auto-tunnel backup srlg exclude force
Router(config)# exit
```

Step 10 **show ip explicit-paths**

Use the following command to verify that the **force** keyword is configured with the pos1/3/1 link excluded from the IP explicit path. For example:

Example:

```
Router# show ip explicit-paths

PATH __dynamic_tunnel65436 (loose source route, path complete,
generation 24, status non-configured)
    1:exclude-address 10.0.0.33
    2:exclude-srlg    10.0.0.33
```

Step 11 **show mpls traffic-eng tunnels tunnel** *num*

Use the following command to show that autotunnel backup is configured but is down because the headend router does not have any other path to signal and it cannot use pos1/2/1 because it belongs in the same SRLG; that is, SRLG 1. For example:

Example:

```
Router# show mpls traffic-eng tunnels tunnel 65436
Name:R3-PRP_t65436                       (Tunnel65436) Destination:
10.0.4.1
  Status:
    Admin:up         Oper:down   Path:not valid   Signalling:Down
    path option 1, type explicit __dynamic_tunnel65436
  Config Parameters:
    Bandwidth:0        kbps (Global)  Priority:7  7   Affinity:
0x0/0xFFFF
    Metric Type:TE (default)
    AutoRoute: disabled  LockDown:disabled  Loadshare:0
bw-based
    auto-bw:disabled
```

```
  Shortest Unconstrained Path Info:
    Path Weight:10 (TE)
    Explicit Route:10.0.0.34 10.0.4.1
  History:
    Tunnel:
      Time since created:5 minutes, 29 seconds
    Path Option 1:
      Last Error:PCALC::No path to destination, 0000.0000.0004.00
```

Step 12 **mpls traffic-eng auto-tunnel backup srlg exclude preferred**

The following commands configure autotunnel backup with the **preferred** keyword. For example:

Example:

```
Router# configure terminal
Router(config)# mpls traffic-eng auto-tunnel backup
Router(config)# mpls traffic-eng auto-tunnel backup srlg exclude preferred
Router(config)# exit
```

Step 13 **show ip explicit-paths**

The following command shows two explicit paths. The first path avoids the SRLGs of the protected interface. The second path does not avoid the SRLGs. For example:

Example:

```
Router# show ip explicit-paths

PATH __dynamic_tunnel65436 (loose source route, path complete,
generation 30, status non-configured)
    1:exclude-address 10.0.0.33
    2:exclude-srlg    10.0.0.33
PATH __dynamic_tunnel65436_pathopt2 (loose source route, path complete,
generation 33, status non-configured)
    1:exclude-address 10.0.0.33
```

Step 14 **show ip rsvp fast-reroute**

The following command shows that the primary tunnel is protected with autotunnel backup using the second path option (see Step 10) that does not avoid the SRLGs. For example:

Example:

```
Router# show ip rsvp fast-reroute
Primary    Protect      BW      Backup
Tunnel     I/F        BPS:Type  Tunnel:Label  State   Level   Type
-------    -------    --------  ------------  -----   -----   ----
R3-PRP_t0  PO1/3/1 0:G  0:G       Tu65436:0     Ready   any-unl nhop
```

The following command shows the path options for the tunnel Tu65436:

Example:

```
Router# show mpls traffic-eng tunnels tunnel 65436
Name:R3-PRP_t65436                       (Tunnel65436) Destination:
10.0.4.1
  Status:
    Admin:up         Oper:up     Path:valid       Signalling:connected
    path option 2, type explicit __dynamic_tunnel65436_pathopt2 (Basis
for Setup, path weight 80020)
    path option 1, type explicit __dynamic_tunnel65436
```

```
  Config Parameters:
    Bandwidth:0        kbps (Global)  Priority:7  7   Affinity:
0x0/0xFFFF
    Metric Type:TE (default)
    AutoRoute: disabled  LockDown:disabled  Loadshare:0
bw-based
    auto-bw:disabled
  Active Path Option Parameters:
    State:explicit path option 2 is active
    BandwidthOverride:disabled  LockDown:disabled  Verbatim:disabled
InLabel  : -
  OutLabel :POS1/2/1, 23
  RSVP Signalling Info:
       Src 10.0.3.1, Dst 10.0.4.1, Tun_Id 65436, Tun_Instance 3
    RSVP Path Info:
      My Address:10.0.3.1
      Explicit Route:10.0.0.50 10.0.0.66 10.0.0.113 10.0.4.1
      Record   Route:  NONE
      Tspec:ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
    RSVP Resv Info:
      Record   Route:  NONE
      Fspec:ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
  Shortest Unconstrained Path Info:
    Path Weight:10 (TE)
    Explicit Route:10.0.0.34 10.0.4.1
```

Step 15 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS Traffic Engineering Shared Risk Link Groups

Configuring the SRLG Membership of Each Link That Has a Shared Risk with Another Link Example

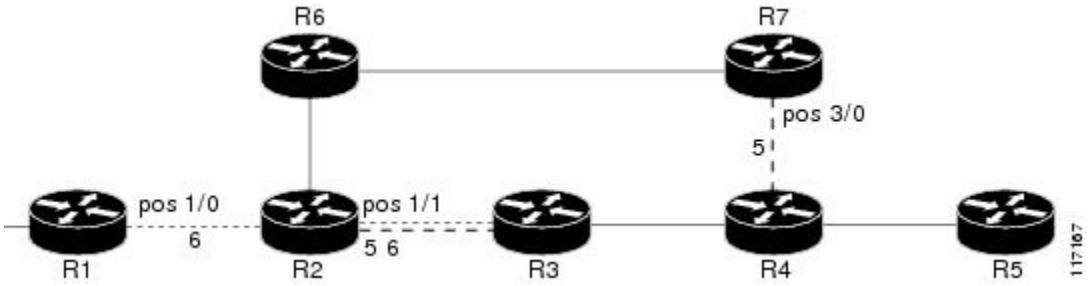
The following example shows how to specify that the SRLG membership of each link has a shared risk with another link.

As shown in the figure below and in the following commands:

- link R2-R3 = SRLG5
- link R2-R3 = SRLG6
- link R7-R4 = SRLG5
- link R1-R2 = SRLG6

```
Router1# configure terminal
Router1# interface pos 1/0
Router1(config-if)# mpls traffic-eng srlg 6

Router2# configure terminal
Router2# interface pos 1/1
Router2(config-if)# mpls traffic-eng srlg 5
Router2(config-if)# mpls traffic-eng srlg 6
Router7# configure terminal
Router7# interface pos 3/0
Router7(config-if)# mpls traffic-eng srlg 5
```

Figure 186: SRLG Membership



Configuring the Routers That Automatically Create Backup Tunnels to Avoid SRLGs Example

The following example shows how to specify that automatically created backup tunnels are forced to avoid SRLGs of their protected interfaces:

```
Router# configure terminal
Router(config)# mpls traffic-eng auto-tunnel backup
Router(config)# mpls traffic-eng auto-tunnel backup srlg exclude force
```

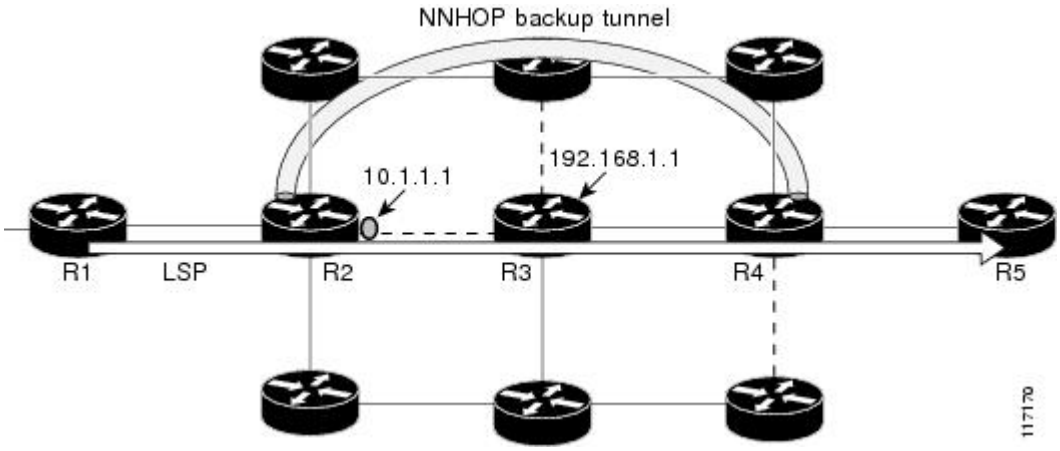
The figure below illustrates the automatically created NNHOP backup tunnel that would be created to avoid SRLGs of the protected interface if the following conditions exist:

The exclude address is 192.168.1.1.

The link at R2 has an IP address of 10.1.1.1.

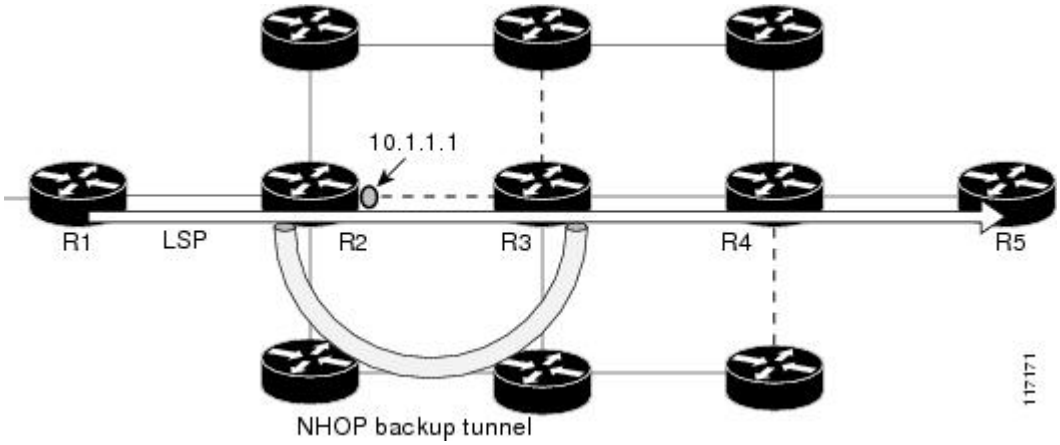
The backup tunnel's explicit path avoids links that have a membership in the same SRLG as the link whose IP address is 10.1.1.1.

Figure 187: srlg exclude force--NNHOP Autobackup Tunnel



The figure below illustrates the automatically created NHOP backup tunnel that would be created.

Figure 188: srlg exclude force--NHOP Autobackup Tunnel



Additional References

Related Documents

Related Topic	Document Title
Fast Reroute	MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection)
IS-IS	Integrated IS-IS Routing Protocol Overview
OSPF	Configuring OSPF
Autotunnel backups	MPLS Traffic Engineering AutoTunnel Primary and Backup

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
draft-ietf-isis-gmpls-extensions-16.txt	*IS-IS Extensions in Support of Generalized MPLS*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering Shared Risk Link Groups

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 186: Feature Information for MPLS Traffic Engineering Shared Risk Link Groups

Feature Name	Releases	Feature Information
MPLS Traffic Engineering: Shared Risk Link Groups	12.0(28)S 12.0(29)S 12.2(33)SRA 12.2(33)SXH 12.4(20)T Cisco IOS XE Release 3.5S	The MPLS Traffic Engineering: Shared Risk Link Groups feature enhances backup tunnel path selection so that a backup tunnel avoids using links that are in the same Shared Risk Link Group (SRLG) as interfaces the backup tunnel is protecting. SRLGs refer to situations where links in a network share a common fiber (or a common physical attribute). If one link fails, other links in the group may fail too. Links in the group have a shared risk. This document contains information about and instructions for configuring the MPLS Traffic Engineering Shared Risk Link Groups feature In 12.0(28)S, this feature was introduced. In 12.0(29)S, support was added for Open Shortest Path First (OSPF). In 12.2(33)SRA, this feature was integrated into a Cisco IOS 12.2SRA release In 12.2(33)SXH, this feature was integrated into a Cisco IOS 12.2SXH release. In 12.4(20)T, this feature was integrated into a Cisco IOS 12.4T release. In Cisco IOS XE Release 3.5S, this feature was integrated into Cisco IOS XE Release 3.5S. The following commands were introduced or modified: **mpls traffic-eng auto-tunnel backup srlg exclude**, **mpls traffic-eng srlg**, **show ip explicit-paths**, **show mpls traffic-eng link-management advertisements**, **show mpls traffic-eng link-management interfaces**, and **show mpls traffic-eng topology**.

Glossary

Fast Reroute --A mechanism for protecting MPLS traffic engineering (TE) LSPs from link and node failure by locally repairing the LSPs at the point of failure. This protection allows data to continue to flow on them while their headend routers attempt to establish end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

hop --Passage of a data packet between two network nodes (for example, between two routers).

IGP --Interior Gateway Protocol. An Internet protocol used to exchange routing information within an autonomous system.

interface --A network connection.

IP address --A 32-bit address assigned to hosts using TCP/IP. An IP address belongs to one of five classes (A, B, C, D, or E) and is written as four octets separated by periods (dotted decimal format). Each address consists of a network number, an optional subnetwork number, and a host number. The network and subnetwork numbers together are used for routing, and the host number is used to address an individual host within the

network or subnetwork. A subnet mask is used to extract network and subnetwork information from the IP address.

IP explicit path --A list of IP addresses, each representing a node or link in the explicit path.

IS-IS --Intermediate System-to-Intermediate System. OSI link-state hierarchal routing protocol based on DECnet Phase V routing, where intermediate system (IS) routers exchange routing information based on a single metric to determine the network topology.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

link --A point-to-point connection between adjacent nodes.

LSP --label-switched path. A path that is followed by a labeled packet over several hops, starting at an ingress LSR and ending at an egress LSR.

LSR --label switching router. A Layer 3 router that forwards a packet based on the value of a label encapsulated in the packet.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets. ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

node --An endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network.

OSPF --Open Shortest Path First. A link-state hierarchical Interior Gateway Protocol (IGP) routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

router ID --Something by which a router originating a packet can be uniquely distinguished from all other routers; for example, an IP address from one of the router's interfaces.

traffic engineering --The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel --A secure communication path between two peers, such as two routers. A traffic engineering tunnel is a label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.

CHAPTER 107

MPLS Traffic Engineering Inter-AS TE

The MPLS Traffic Engineering: Inter-AS TE feature provides Autonomous System Boundary Router (ASBR) node protection, loose path reoptimization, stateful switchover (SSO) recovery of label-switched paths (LSPs) that include loose hops, ASBR forced link flooding, Cisco IOS Resource Reservation Protocol (RSVP) local policy extensions for interautonomous system (Inter-AS), and per-neighbor keys:

- ASBR node protection--Protects interarea and Inter-AS TE label-switched paths (LSPs) from the failure of an Area Border Router (ABR) or ASBR.
- Loose path reoptimization--Allows a Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnel's LSPs to traverse hops that are not in the tunnel headend router's topology database (that is, they are not in the same Open Shortest Path First (OSPF) area, Intermediate System-to-Intermediate System (IS-IS) level, or autonomous system as the tunnel's headend router).
- Loose hop recovery--Supports SSO recovery of LSPs that include loose hops.
- ASBR forced link flooding--Helps an LSP cross a boundary into another domain when information in the other domain is not available to the headend router.
- Cisco IOS RSVP local policy extensions for Inter-AS--Allows network administrators to create controlled policies for TE tunnels that function across multiple autonomous systems.
- Per-neighbor keys--Allows cryptographic authentication to be accomplished on a per-neighbor basis.

Prerequisites for MPLS Traffic Engineering Inter-AS TE

- Enable MPLS.
- Configure TE on routers.

- Ensure that your network supports the following Cisco features:
  - MPLS
  - Cisco Express Forwarding
  - IS-IS or OSPF
- For loose path reoptimization, know how to configure the following:
  - IP explicit paths for MPLS TE tunnels
  - Loose hops
  - Interarea and Inter-AS tunnels

Restrictions for MPLS Traffic Engineering Inter-AS TE

Loose Path Reoptimization

- Midpoint reoptimization is not supported.

ASBR Forced Link Flooding

- The TE metric and affinity attributes that are known at a headend router (and used as constraints when an LSP's path is computed) are not currently signaled. Consequently, explicit router (ERO) expansions do not consider these constraints.
- Each node in an autonomous system must have a unique router ID.
- The router ID configured on a link must not conflict with the router ID within the autonomous system.
- If a link is configured for forced link flooding, the link's neighbors are not learned by regular Interior Gateway Protocol (IGP) updates. If a link is already learned about neighbors by IGP on a link, you cannot configure the link as passive. Therefore, to configure a link for forced flooding, be sure that the node does not already have a neighbor on that link.

Information About MPLS Traffic Engineering Inter-AS TE

MPLS Traffic Engineering Tunnels

MPLS TE lets you build LSPs across your network that you then forward traffic down.

MPLS TE LSPs, also called TE tunnels, let the headend of a TE tunnel control the path its traffic takes to a particular destination. This method is more flexible than forwarding traffic based only on a destination address.

Interarea tunnels allow you to do the following:

- Build TE tunnels between areas (interarea tunnels)
- Build TE tunnels that start and end in the same area, on multiple areas on a router (intra-area tunnels)

Some tunnels are more important than others. For example, you may have tunnels carrying Voice over IP (VoIP) traffic and tunnels carrying data traffic that are competing for the same resources. Or you may simply

have some data tunnels that are more important than others. MPLS TE allows you to have some tunnels preempt others. Each tunnel has a priority, and more-important tunnels take precedence over less-important tunnels.

Multiarea Network Design

You can establish MPLS TE tunnels that span multiple IGP areas and levels. The tunnel headend routers and tailend routers do not have to be in the same area. The IGP can be either IS-IS or OSPF.

To configure an interarea tunnel, use the **next-address loose** command to specify on the headend router a loosely routed explicit path of the LSP that identifies each ABR the LSP should traverse. The headend router and the ABRs along the specified explicit path expand the loose hops, each computing the path segment to the next ABR or tunnel destination.

Fast Reroute

MPLS Fast Reroute (FRR) is a fast recovery local protection technique that protects TE LSPs from link, shared risk link group (SRLG), and node failure. One or more TE LSPs (called backup LSPs) are preestablished to protect against the failure of a link, node, or SRLG. If there is a failure, each protected TE LSP traversing the failed resource is rerouted onto the appropriate backup tunnels.

The backup tunnel must meet the following requirements:

- It should not pass through the element it protects.
- It should intersect with a primary tunnel at a minimum of two nodes: point of local repair (PLR) and merge point (MP). The PLR should be the headend LSR of the backup tunnel, and the MP should be the tailend LSR of the backup tunnel. The PLR is where FRR is triggered when a link, node, or SRLG failure occurs.
- FRR protection can be performed for an Inter-AS tunnel only if the backup tunnel's merge point can route packets to the PLR's backup tunnel's egress interface. You can configure a static route or you can configure Border Gateway Protocol (BGP) to export the backup tunnel's egress interface to other autonomous systems.
- If the preferred link is a passive link, you must assign an administrative-weight for it. To assign an administrative weight, use the **mpls traffic-eng administrative-weight** command in interface configuration mode.
- Each router must be configured with the **mpls traffic-eng reoptimize events link-up** command in global configuration mode.

ASBR Node Protection

A TE LSP that traverses an ASBR needs a special protection mechanism (ASBR node protection) because the MP and PLR will be in different autonomous systems that have different IGPs.

A PLR ensures that the backup tunnel intersects with the primary tunnel at the MP by examining the Record Route Object (RRO) of the primary tunnel to see if any addresses specified in the RRO match the destination of the backup tunnel.

Addresses specified in RRO IPv4 and IPv6 subobjects can be node-IDs and interface addresses. The traffic engineering RFC 3209 specifies that you can use a router address or interface address, but recommends using

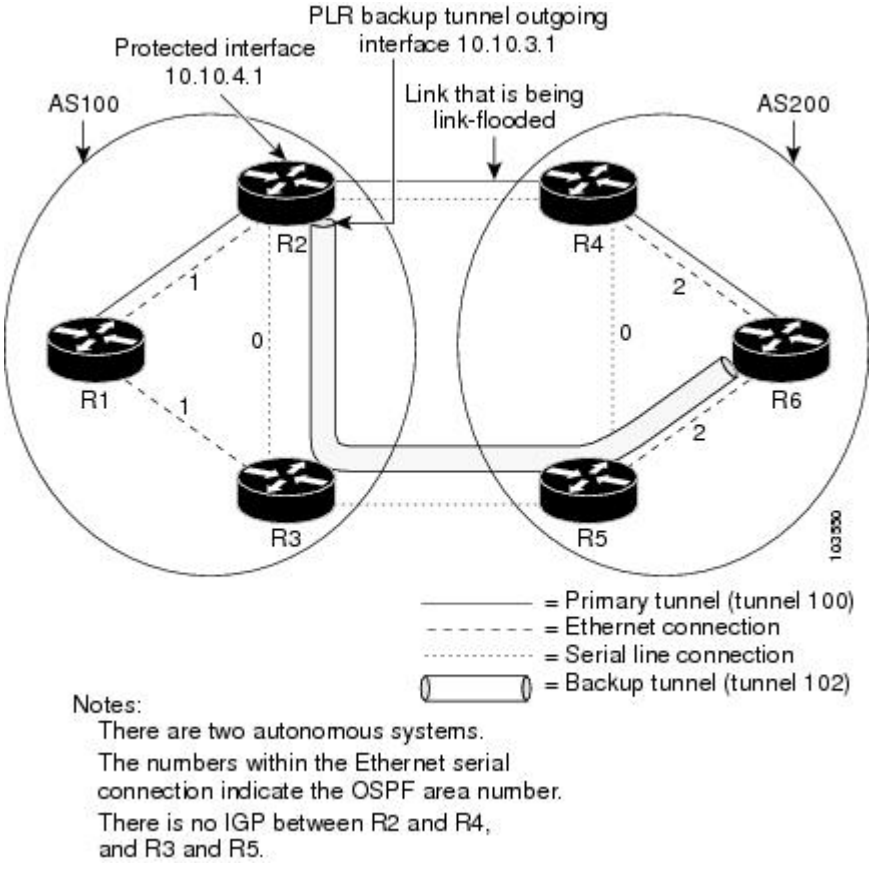
the interface address of outgoing path messages. Therefore, in the figure below router R2 is more likely to specify interface addresses in the RRO objects carried in the resv messages of the primary tunnel (T1) and the backup tunnel.

Node IDs allow the PLR to select a suitable backup tunnel by comparing node IDs in the resv RRO to the backup tunnel's destination.

RSVP messages that must be routed and forwarded to the appropriate peer (for example, an resv message) require a route from the MP back to the PLR for the RSVP messages to be delivered. The MP needs a route to the PLR backup tunnel's outgoing interface for the resv message to be delivered. Therefore, you must configure a static route from the MP to the PLR. For the configuration procedure, see the Configuring a Static Route from the MP to the PLR, on page 2285.

The figure below illustrates ASBR node protection. Router R4 is node-protected with a backup tunnel from R2-R3-R5-R6.

Figure 189: ASBR Node Protection



In this configuration, IP addresses are as follows:

- R1--Loopback0 10.10.0.1
  - Ethernet 0--IP address of 10.10.1.1 is connected to R2 Ethernet 0
  - Ethernet 1--IP address of 10.10.2.1 is connected to R3 Ethernet 1
- R2--Loopback0 10.10.0.2
  - Ethernet 0--IP address of 10.10.1.2 is connected to R1 Ethernet 0

  - Ethernet 1--IP address of 10.10.3.1 is connected to R3 Ethernet 1
  - Serial 2--IP address of 10.10.4.1 is connected to R4 serial 2
- R3--Loopback0 10.10.0.3
  - Ethernet 0--IP address of 10.10.2.2 is connected to R1 Ethernet 1
  - Ethernet 1--IP address of 10.10.3.2 is connected to R2 Ethernet 1
  - Serial 2--IP address of 10.10.5.1 is connected to R5 serial 2
- R4--Loopback0 10.10.0.4
  - Ethernet 0--IP address of 10.10.7.1 is connected to R6 Ethernet 0
  - Ethernet 1--IP address of 10.10.6.1 is connected to R5 Ethernet 1
  - Serial 2--IP address of 10.10.4.2 is connected to R2 serial 2
- R5--Loopback0 10.10.0.5
  - Ethernet 0--IP address of 10.10.8.1 is connected to R6 Ethernet 0
  - Ethernet 1--IP address of 10.10.6.2 is connected to R4 Ethernet 1
  - Serial 2--IP address of 10.10.5.2 is connected to R3 serial 2
- R6--Loopback0 10.10.0.6
  - Ethernet 0--IP address of 10.10.7.2 is connected to R4 Ethernet 0
  - Ethernet 1--IP address of 10.10.8.2 is connected to R5 Ethernet 1

In the figure above, the following situations exist:

- Routers R1, R2, and R3 are in AS 100. The R1-R2 and R1-R3 links are in OSPF area 1.
- Routers R4, R5, and R6 are in AS200. The R4-R6 and R5-R6 links are in OSPF area 2.
- The link R2-R3 is in AS100, and link R4-R5 is in AS200. The links R2-R3 and R4-R5 are in OSPF area 0.
- The links R2-R4 and R3-R5 are not running an IGP because they cross the Inter-AS boundary between AS100 and AS200. Because they are not running IGP, you must configure an administrative weight for each passive interface for FRR to work. Use the **mpls traffic-eng administrative-weight** command in interface configuration mode.
- There is a primary tunnel, tunnel 100, from R1-R2-R4-R6.
- There is a backup tunnel, tunnel 102, from R2-R3-R5-R6.
- There is a TE tunnel, tunnel 101, from R6-R5-R3-R1 for returning data traffic for tunnel 100.
- There is a TE tunnel, tunnel 103, from R6-R5-R3-R2 for returning data traffic for tunnel 102.
- The explicit paths of all the tunnels use loose hops.
- The R2-R4 link is configured to be link flooded in both R2's and R4's IGP. The R3-R5 link is configured to be link flooded in both R3's and R5's IGP.

Router R2 needs to ensure the following:

- Backup tunnel intersects with the primary tunnel at the MP, and therefore has a valid MP address. In the figure above, R2 needs to determine that tunnel 100 and backup tunnel 102 share MP node R6.

- Backup tunnel satisfies the request of the primary LSP for bandwidth protection. For example, the amount of bandwidth guaranteed for the primary tunnel during a failure, and the type of protection (preferably protecting against a node failure rather than a link failure).

Node-IDs Signaling in RROs

ASBR node protection includes a node-ID flag (0x20), which is also called a node-ID subobject. When it is set, the flag indicates that the address specified in the RRO object in the resv message is the node-ID address. The node-ID address refers to the traffic engineering router ID.

A node must always use the same address in the RRO (that is, it must use IPv4 or IPv6, but not both).

To display all the hops, enter the following command on the headend router. Sample command output is as follows:

```
Router(config)# show ip rsvp reservations detail
Reservation:
 Tun Dest:   10.10.0.6  Tun ID: 100  Ext Tun ID: 10.10.0.1
 Tun Sender: 10.10.0.1  LSP ID: 31
 Next Hop: 10.10.1.2 on Ethernet0/0
 Label: 17 (outgoing)
 Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
 Average Bitrate is 10K bits/sec, Maximum Burst is 1K bytes
 Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
 RRO:
   10.10.0.2/32, Flags:0x29 (Local Prot Avail/to NNHOP, Is Node-id)
   10.10.4.1/32, Flags:0x9 (Local Prot Avail/to NNHOP)
     Label subobject: Flags 0x1, C-Type 1, Label 17
   10.10.0.4/32, Flags:0x20 (No Local Protection, Is Node-id)
   10.10.7.1/32, Flags:0x0 (No Local Protection)
     Label subobject: Flags 0x1, C-Type 1, Label 17
   10.10.0.6/32, Flags:0x20 (No Local Protection, Is Node-id)
   10.10.7.2/32, Flags:0x0 (No Local Protection)
     Label subobject: Flags 0x1, C-Type 1, Label 0
 Resv ID handle: 0100040E.
 Status:
 Policy: Accepted. Policy source(s): MPLS/TE
```

For a description of the fields, see the Cisco IOS Quality of Service Solutions Command Reference.

Addition of the Node-ID Subobject

When a fast reroutable LSP is signaled, the following actions occur:

- An LSR adds a node-ID subobject and an incoming label subobject in the resv message.
- If there is an RRO object in the path message, an LSR adds a node-ID subobject, an RRO IPv4 subobject that records the interface address, and an incoming label subobject in the resv message.

If you enable record-route on the headend LSR, the interface addresses for the LSP are included in the RRO object of the resv message.

To enable record-route, enter the following command with the **record-route** keyword:

```
tunnel mpls traffic-eng record-route
```

Processing of an RRO with Node-ID Subobjects

The node-ID subobject is added to the RECORD_ROUTE object before the label route subobject. If RECORD_ROUTE is turned on, the RRO object consists of the following in this order: node-ID, interface address, and label.

Merge Point Location

The destination of the backup tunnel is the node-ID of the MP. A PLR can find the MP and appropriate backup tunnel by comparing the destination address of the backup tunnel with the node-ID subobjects included in the resv RRO for the primary tunnel.

When both the IPv4 node-ID and IPv6 node-ID subobjects are present, a PLR can use either or both of them to find the MP address.

Determination of Backward Compatibility

To remain compatible with nodes that do not support RRO IPv4 or IPv6 node-ID subobjects, a node can ignore those objects. Those nodes cannot be the MP in a network with interarea or Inter-AS traffic engineering.

Loose Path Reoptimization

Interarea and Inter-AS LSPs

If the LSP of an MPLS TE tunnel traverses hops that are not in the headend router's topology database (that is, the hops are in a different OSPF area or IS-IS level), the LSP is called an *interarea TE LSP* .

If the LSP of the tunnel traverses hops that are in a different autonomous system (AS) from the tunnel's headend router, the LSP is called an *Inter-AS TE LSP* .

Interarea LSPs and Inter-AS TE LSPs can be signaled using loose hop subobjects in their EROs. The headend does not have "strict" knowledge of hops beyond its area, so the LSP's path is "loosely" specified at the headend. Downstream routers processing these loose hop subobjects (which do have the knowledge) are relied upon to expand them into strict hops.

Loose Hop Configuration

Beyond the headend area, configure hops as loose hops. Typically you specify only the ABRs and the tailend router of a tunnel, but any other combination is allowed.

Loose Hop Expansion

Loose hop expansion is the conversion of a single ERO loose hop subobject into one or more strict hop subobjects.

Interarea and Inter-AS TE LSPs can be signaled using loose hop subobjects in their EROs. When a router receives a path message containing an ERO that has a loose hop as the next address, the router typically expands the ERO by converting the single loose hop subobject into one or more strict hop subobjects. The router typically has the knowledge, in its topology database, of the best way to reach the loose hop and computes this path by using constraint-based shortest path first (CSPF). So the router substitutes this more specific information for the loose hop subobject found in the ERO. This process is called loose hop expansion or ERO expansion.

Loose hop expansions can occur at one or more hops along an LSP's path. This process is referred to as loose path reoptimization.

Tunnel Reoptimization Procedure

Tunnel reoptimization is the signaling of an LSP that is more optimal than the LSP a TE tunnel is currently using (for example, it may be shorter or may have a lower cost), and the switching over of the tunnel's data to use this new LSP.

The new more optimal TE LSP is always established and the data moved onto it before the original LSP is torn down (so it is called the "make before break" procedure). This ensures that no data packets are lost during the transition to the new LSP.

For tunnel reoptimization to function:

- Each router must be configured with the **mpls traffic-eng reoptimize events link-up** command.
- Each passive link must have an assigned administrative weight. To configure an administrative weight, use the **mpls traffic-eng administrative-weight** command in interface configuration mode.

The TE LSPs reoptimization process is triggered under the following circumstances:

- Periodically (based on a timer)
- User entered a command (**mpls traffic-eng reoptimize**) requesting reoptimization
- Network event, such as a link-up

Regardless of how reoptimization is triggered, the headend router reoptimizes a tunnel only if it can find a better path than the one the tunnel currently uses. If there is not a better path in the local topology database, no new LSP is signaled and reoptimization does not occur.

Prior to the addition of loose path reoptimization, interarea TE LSPs were not reoptimized if a better path became available in any area beyond the headend area. This is because the headend router was not capable of finding a better path when the better path existed in an area beyond its view (that is, it was not in its local topology database).

With the addition of loose path reoptimization, a tunnel's headend can reoptimize LSPs even if they span multiple areas, levels, or autonomous systems. This is done via the implementation of a query and response protocol defined in *draft-vasseur-mpls-loose-path-reopt-02.txt* . This draft defines a protocol whereby a tunnel's headend may query downstream routers to perform ERO expansion for this tunnel's LSP. These downstream routers respond in the affirmative if they can find a more optimal path than the one in use. (This is done via a new ERO expansion.) Having received an affirmative answer to its query, a headend signals a new LSP for the tunnel, and the new LSP benefits from a new ERO expansion along the better path.

Loose path reoptimization is on by default, and cannot be disabled. Whenever an LSP reoptimization is attempted but the headend fails to find a better path, if the LSP contains loose ERO subobjects, a query is sent downstream to determine whether downstream routers can find a better path. If an affirmative answer comes back, the LSP is reoptimized. That is, a new LSP is signaled (which will follow the better path), the tunnel's data packets are switched over to use this new LSP, and the original LSP is torn down.

For details on this query and response protocol, see *draft-vasseur-mpls-loose-path-reopt-02.txt.*

ASBR Forced Link Flooding

When you configure forced link flooding on an interface, the MPLS TE link management module advertises the link to all nodes. As a result of this advertisement, the TE topology database on all the nodes within the Inter-AS is updated with this information.

ASBR forced link flooding allows the links to be advertised even if IGP adjacencies are not running over these links. TE LSPs can traverse these links at the edge of a network between two nodes running BGP (or static routes) even if the exit ASBR is not listed in the IP explicit path. Therefore, a headend LSR can consider that link when it computes its TE LSP path.

Configuration of ASBR Forced Link Flooding

To activate ASBR forced link flooding, configure a link as passive and provide neighbor information (that is, the neighbor IGP ID and the neighbor TE ID).

Link Flooding

A passive link is configured on an interface of an ASBR. The link is flooded in the ASBR's IGP. All the links are flooded as point-to-point links.

Flooding notifications are also sent when there is a change to a link's property.

OSPF Flooding

OSPF floods opaque link-state advertisement (LSA) Type 10 link information.

If a multiaccess link has more than one neighbor, a Type 10 LSA is advertised for each neighbor. In the topology database, neighbors are represented by point-to-point neighbor relationships.

Link TLV

A link TLV describes a single link and contains multiple sub-TLVs.

An opaque LSA contains a single link TLV.

For each ASBR-to-ASBR link, an ASBR must flood an opaque LSA containing one link TLV that has the link's attributes.

A link TLV comprises the following sub-TLVs:

- Link type (1 octet)--(Required) Defines the type of the link. The link type of a passive interface always is 1 (point-to-point), even for a multiaccess subnetwork.
- Link ID (4 octets)--(Required) Identifies the other end of the link for a point-to-point link. Includes the system ID of the neighbor, requires static configuration for a multiaccess ASBR-to-ASBR link, and includes the system ID of the neighbor.
- Local interface IP address (4 octets)--Specifies the IP addresses of the neighbor's interface corresponding to this link.
- Remote interface IP address (4 octets)--Specifies the IP addresses of the neighbor's interface corresponding to this link. The remote interface IP address is set to the router ID of the next hop. There must be a static configuration for the ASBR-to-ASBR link.
- Traffic engineering metric (4 octets)
- Maximum bandwidth (4 octets)

- Maximum reservable bandwidth (4 octets)
- Unreserved bandwidth (32 octets)
- Administrative group (4 octets)

IS-IS TLV

In IS-IS, when autonomous system A1 floods its LSP, it includes the system ID and a pseudonode number.

If three autonomous systems are connected to a multiaccess network LAN, each link is considered to be a point-to-point link. The links are marked with the maximum metric value so that the inter-ASBR links are considered by CSPF and not by shortest path first (SPF).

TE uses the protocol TLV type 22, which has the following data structure:

- System ID and pseudonode number node (7 octets)
- Default metric (3 octets)
- Length of sub-TLVs (1 octet)
- Sub-TLVs (0 to 244 octets), where each sub-TLV consists of a sequence of the following: 1 octet for subtype, 1 octet for the length of the value field of the sub-TLV, and 0 to 242 octets for the value

The table below defines the sub-TLVs.

Table 187: Sub-TLVs

Sub-TLV	Length (Octets)	Name
3	4	Administrative group (color).
6	4	IPv4 address for the interface described by the main TLV.
8	4	IPv4 address for a neighboring router on this link. This will be set to the router ID of the next hop.
9	4	Maximum link bandwidth.
10	4	Reservable link bandwidth.
11	32	Unreserved bandwidth.
18	3	TE default metric.
250 to 254	--	Reserved for Cisco-specific extensions.
255	--	Reserved for future expansion.

Note The TE router ID is TLV type 134.

Topology Database

When the topology database module receives a link-state advertisement (LSA), the module scans the LSA to find the neighbors of the links. The ASBR link is part of the same LSA and is installed in the TE topology database like any other link.

During the CSPF operation, the TE headend module uses the TE topology database to find a path to the destination. Because the Inter-AS links are part of the TE topology database, the CSPF operation uses these links to compute the LSP path.

Link Flooding

The IGP floods information about a link in the following situations:

- When a link goes down
- When a link's configuration is changed (for example, when the link cost is modified)
- When it is time to periodically reflood the router's IGP information
- When link bandwidth changes significantly

Flooding is a little different in IS-IS and OSPF. In OSPF, only information about the link that has changed is flooded, because a Type 10 LSA contains a single link advertisement. In IS-IS, information about all links on a node is flooded even if only one has changed, because the Type 22 TLV contains a list of all links on the router.

How to Configure MPLS Traffic Engineering Inter-AS TE

Note There is no configuration procedure for loose path reoptimization.

Configuring Loose Hops

The section describes how to do the following so that there can be loose hops:

Configuring an Explicit Path on the Tunnel That Will Cross the Inter-AS Link

If you want a tunnel to span multiple networks, configure an explicit path on the tunnel that will cross the Inter-AS link by performing the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip explicit-path** {**name** *path-name* | **identifier** *number*} [**enable** | **disable**]
4. **next-address loose** *A.B.C.D*
5. **interface tunnel** *number*
6. **tunnel mpls traffic-eng fast-reroute**

7. **mpls traffic-eng reoptimize events link-up**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip explicit-path** {**name** *path-name* | **identifier** *number*} [**enable** | **disable**] **Example:** `Router(config)# ip explicit-path identifier 2 enable`	Enters the subcommand mode for IP explicit paths and creates or modifies the explicit path. This command places the router in IP explicit path configuration mode.
Step 4	**next-address loose** *A.B.C.D* **Example:** `Router(cfg-ip-expl-path)# next-address loose 10.10.0.2`	Specifies the next loose IP address in the explicit path. Each area border router (ABR) the path must traverse should be specified in a **next-address loose** command. This command places the router in global configuration mode.
Step 5	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 100`	Configures a tunnel interface. This command places the router in interface configuration mode.
Step 6	**tunnel mpls traffic-eng fast-reroute** **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute`	Enables an MPLS traffic engineering tunnel to use an established backup tunnel in the event of a link failure.
Step 7	**mpls traffic-eng reoptimize events link-up** **Example:** `Router(config)# mpls traffic-eng reoptimize events link-up`	Enables automatic reoptimization of MPLS traffic engineering when an interface becomes operational.

Configuring a Route to Reach the Remote ASBR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number*} **Example:** `Router(config)# ip route 10.10.0.1 255.255.255.255 tunnel 101`	Establishes static routes.

Configuring a Static Route from the MP to the PLR

To enable Fast Reroute protection that spans across different autonomous systems, configure a static route from the MP to the PLR by performing the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask ip-address outgoing-interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask ip-address outgoing-interface* **Example:** `Router(config)# ip route 10.10.3.1 255.255.255.255 10.0.0.0 FastEthernet0/0`	Establishes static routes. Refer to the appropriate hardware manual for interface information. **Note** Enter this command on the MP. The destination is the PLR.

Configuring ASBR Forced Link Flooding

This section describes how to do the following so that you can configure ASBR forced link flooding:

Configuring the Inter-AS Link as a Passive Interface Between Two ASBRs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **ip address** *ip-address mask* [**secondary**]
5. **mpls traffic-eng passive-interface nbr-te-id** *te-router-id* [**nbr-if-addr** *if-addr*] [**nbr-igp-id** {**isis** *sysid* | **ospf** *sysid*}]
6. **mpls traffic-eng administrative-weight** *weight*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router(config)# interface serial 2/0`	Specifies an interface and enters interface configuration mode. Refer to the appropriate hardware manual for interface information.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.10.4.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**mpls traffic-eng passive-interface nbr-te-id** *te-router-id* [**nbr-if-addr** *if-addr*] [**nbr-igp-id** {**isis** *sysid* | **ospf** *sysid*}] **Example:** `Router(config-if)# mpls traffic-eng passive-interface nbr-te-id 10.10.11.12 nbr-igp-id ospf 10.10.15.18`	Configures a link as a passive interface between two ASBRs. **Note** For an RSVP Hello configuration on the Inter-AS link, all fields are required.
Step 6	**mpls traffic-eng administrative-weight** *weight* **Example:** `Router(config-if)# mpls traffic-eng administrative-weight 20`	Overrides the Interior Gateway Protocol (IGP) administrative weight (cost) of the link and assigns a specific weight for the link.

Creating LSPs Traversing the ASBRs

To create LSPs traversing the ASBRs, perform the following steps.

Note Perform Steps 3 through 7 for the primary LSP and then for the backup LSP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip explicit path** *name* **enable**
4. **next-address loose** *A.B.C.D*
5. **interface tunnel** *number*
6. **tunnel mpls traffic-eng fast-reroute**
7. **tunnel mpls traffic-eng path-option***number* {**dynamic** | **explicit** | {**name** *path-name* | *path-number*}} [**lockdown**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip explicit path** *name* **enable** **Example:** `Router(config)# ip explicit path route1 enable`	Specifies the name of the explicit path and enables the path.
Step 4	**next-address loose** *A.B.C.D* **Example:** `Router(config)# next-address loose 10.10.10.2`	Configures a loose hop.
Step 5	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 100`	Configures a tunnel interface and enters interface configuration mode.
Step 6	**tunnel mpls traffic-eng fast-reroute** **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute`	Enables an MPLS traffic engineering tunnel to use an established backup tunnel in the event of a link failure.
Step 7	**tunnel mpls traffic-eng path-option***number* {**dynamic** | **explicit** | {**name** *path-name* | *path-number*}} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 route1`	Configures a path option for an MPLS traffic engineering tunnel.

Configuring Multiple Neighbors on a Link

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls traffic-eng passive-interface** [**nbr-te-id**] [*router-id* | *te-id*] [**nbr-igp-id**] [**isis** *sysid* | **ospf** *sysid*]
5. **mpls traffic-eng administrative-weight** *weight*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router(config)# interface serial 2/0`	Specifies an interface and enters interface configuration mode. Refer to the appropriate hardware manual for interface information.
Step 4	**mpls traffic-eng passive-interface** [**nbr-te-id**] [*router-id* | *te-id*] [**nbr-igp-id**] [**isis** *sysid* | **ospf** *sysid*] **Example:** `Router(config-if)# mpls traffic-eng passive-interface nbr-te-id 10.10.0.4 nbr-igp-id ospf 10.10.0.4`	Configures a link as a passive link.
Step 5	**mpls traffic-eng administrative-weight** *weight* **Example:** `Router(config-if)# mpls traffic-eng administrative-weight 20`	Overrides the Interior Gateway Protocol (IGP) administrative weight (cost) of the link and assigns a specific weight for the link.

Troubleshooting Tips

The following debug commands are useful for troubleshooting issues with MPLS Traffic Engineering: Inter-AS TE.

Debugging Headend of TE LSPs

```
debug mpls traffic-eng path lookup
debug mpls traffic-eng path verify
debug mpls traffic-eng path spf
```

Debugging Head and Midpoint (Link-Related Debugs)

```
debug mpls traffic-eng link-management igp-neighbors
debug mpls traffic-eng link-management advertisements
debug mpls traffic-eng link-management bandwidth-allocation
debug mpls traffic-eng link-management routing
```

Verifying the Inter-AS TE Configuration

To verify the Inter-AS TE configuration, perform the following steps.

Note Perform Step 1 for Fast Reroute ready, and Step 2 for Fast Reroute active.

SUMMARY STEPS

1. **show ip rsvp sender detail**
2. **show ip rsvp sender detail**
3. **show mpls traffic-eng link-management advertisements**

DETAILED STEPS

Step 1 **show ip rsvp sender detail**

Use this command to display the MP sender display for the primary tunnel when Fast Reroute is ready.

Example:

```
Router# show ip rsvp sender detail
PATH:
 Tun Dest:   10.10.0.6  Tun ID: 100  Ext Tun ID: 10.10.0.1
 Tun Sender: 10.10.0.1  LSP ID: 31
 Path refreshes:
  arriving: from PHOP 10.10.7.1 on Et0/0 every 30000 msecs
 Session Attr:
  Setup Prio: 7, Holding Prio: 7
  Flags: (0x7) Local Prot desired, Label Recording, SE Style
  session Name: R1_t100
 ERO: (incoming)
  10.10.7.2 (Strict IPv4 Prefix, 8 bytes, /32)
  10.10.0.6 (Strict IPv4 Prefix, 8 bytes, /32)
 RRO:
   10.10.7.1/32, Flags:0x0 (No Local Protection)
   10.10.4.1/32, Flags:0x9 (Local Prot Avail/to NNHOP) !Available to NNHOP
   10.10.1.1/32, Flags:0x0 (No Local Protection)
 Traffic params - Rate: 10K bits/sec, Max. burst: 1K bytes
   Min Policed Unit: 0 bytes, Max Pkt Size 4294967295 bytes
 Fast-Reroute Backup info:
   Inbound  FRR: Not active
   Outbound FRR: No backup tunnel selected
 Path ID handle: 50000416.
 Incoming policy: Accepted. Policy source(s): MPLS/TE
 Status: Proxy-terminated
```

Step 2 **show ip rsvp sender detail**

Use this command to display the MP sender display when the primary tunnel is Fast Reroute active:

Example:

```
Router# show ip rsvp sender detail
PATH:
 Tun Dest:   10.10.0.6  Tun ID: 100  Ext Tun ID: 10.10.0.1
```

```
Tun Sender: 10.10.0.1  LSP ID: 31
Path refreshes:
 arriving: from PHOP 10.10.3.1 on Et1/0 every 30000 msecs
Session Attr:
 Setup Prio: 7, Holding Prio: 7
 Flags: (0x7) Local Prot desired, Label Recording, SE Style
 Session Name: R1_t100
ERO: (incoming)
 10.10.0.4 (Strict IPv4 Prefix, 8 bytes, /32)
 10.10.0.6 (Loose IPv4 Prefix, 8 bytes, /32)
RRO:
 10.10.3.1/32, Flags:0xB (Local Prot Avail/In Use/to NNHOP) !Ready
 10.10.1.1/32, Flags:0x0 (No Local Protection)
Traffic params - Rate: 10K bits/sec, Max. burst: 1K bytes
 Min Policed Unit: 0 bytes, Max Pkt Size 4294967295 bytes
Fast-Reroute Backup info:
 Inbound  FRR: Active
  Orig Input I/F: Et0/0
  Orig PHOP: 10.10.7.1
  Now using Bkup Filterspec w/ sender: 10.10.3.1 LSP ID: 31
 Outbound FRR: No backup tunnel selected
Path ID handle: 50000416.
Incoming policy: Accepted. Policy source(s): MPLS/TE
Status: Proxy-terminated
```

Step 3 **show mpls traffic-eng link-management advertisements**

Use this command to display the influence of a passive link. On R2, the passive link to R4 is in the Link ID:: 1 section.

Example:

```
Router# show mpls traffic-eng link-management advertisements

Flooding Status: ready
Configured Areas: 2
IGP Area[1] ID:: ospf 1 area 0
 System Information::
  Flooding Protocol: OSPF
 Header Information::
  IGP System ID: 10.10.0.2
  MPLS TE Router ID: 10.10.0.2
  Flooded Links: 2
 Link ID:: 1
  Link Subnet Type: Point-to-Point
  Link IP Address: 10.10.4.1
  IGP Neighbor: ID 0-0-0-0-0-0-0-0, IP 10.10.0.4
  Physical Bandwidth: 1544 kbits/sec
  Res. Global BW: 1158 kbits/sec
  Res. Sub BW: 0 kbits/sec
  Downstream::
                            Global Pool Sub Pool
                            ----------- ---------
   Reservable Bandwidth[0]: 1158        0 kbits/sec
   Reservable Bandwidth[1]: 1158        0 kbits/sec
   Reservable Bandwidth[2]: 1158        0 kbits/sec
   Reservable Bandwidth[3]: 1158        0 kbits/sec
   Reservable Bandwidth[4]: 1158        0 kbits/sec
   Reservable Bandwidth[5]: 1158        0 kbits/sec
   Reservable Bandwidth[6]: 1158        0 kbits/sec
   Reservable Bandwidth[7]: 1148        0 kbits/sec
  Attribute Flags: 0x00000000
 IGP Area[1] ID:: ospf 1 area 1
  System Information::
   Flooding Protocol: OSPF
```

```
Header Information::
 IGP System ID: 10.10.0.2
 MPLS TE Router ID: 10.10.0.2
 Flooded Links: 2
Link ID:: 1
 Link Subnet Type: Point-to-Point
 Link IP Address: 10.10.4.1
 IGP Neighbor: ID 0-0-0-0-0-0-0, IP 10.10.0.4
 Physical Bandwidth: 1544 kbits/sec
 Res. Global BW: 1158 kbits/sec
 Res. Sub BW: 0 kbits/sec
 Downstream::
                          Global Pool Sub Pool
                          ----------- -----------
  Reservable Bandwidth[0]: 1158        0 kbits/sec
  Reservable Bandwidth[1]: 1158        0 kbits/sec
  Reservable Bandwidth[2]: 1158        0 kbits/sec
  Reservable Bandwidth[3]: 1158        0 kbits/sec
  Reservable Bandwidth[4]: 1158        0 kbits/sec
  Reservable Bandwidth[5]: 1158        0 kbits/sec
  Reservable Bandwidth[6]: 1158        0 kbits/sec
  Reservable Bandwidth[7]: 1148        0 kbits/sec
 Attribute Flags: 0x00000000
```

Configuration Examples for MPLS Traffic Engineering Inter-AS TE

Configuring Loose Hops Examples

Configuring an Explicit Path on the Tunnel That Will Cross the Inter-AS Link Example

The following commands configure a loose IP explicit path named route1 suitable for use as a path option with Inter-AS TE with the destination 10.10.10.6 that is to traverse ABRs 10.10.0.2 and 10.10.0.4. The tunnel headend and the specified ABRs will find a path from the source AS100 to the destination 10.10.0.6 in AS200. See the figure above.

```
Router(config)# ip explicit-path name route1 enable
Router(cfg-ip-expl-path)# next-address loose 10.10.0.2
Router(cfg-ip-expl-path)# next-address loose 10.10.0.4
Router(cfg-ip-expl-path)# next-address loose 10.10.0.6
```

Note that the explicit path for an interarea TE tunnel need not specify the destination router because the tunnel configuration specifies it in the tunnel destination command. The following commands configure an explicit path named path-without-tailend that would work equally well for the interarea tunnel created in the previous example:

```
Router(config)# ip explicit-path name path-without-tailend
Router(cfg-ip-expl-path)# next-address loose 10.10.0.2
Router(cfg-ip-expl-path)# next-address loose 10.10.0.4
```

Configuring a Route to Reach the Remote ASBR in the IP Routing Table Example

In the following example, packets for the ASBR whose router ID is 10.10.0.1 will be forwarded via tunnel 101:

```
Router> enable
Router# configure terminal
Router(config)# ip route 10.10.0.1 255.255.255.255 tunnel 101
```

Configuring a Static Route from the MP to the PLR Example

In the following example, a static route is configured from the MP to the PLR. The outgoing interface is tunnel 103.

```
Router> enable
Router# configure terminal
Router(config)# ip route 10.10.3.1 255.255.255.255 tunnel 103
```

Configuring ASBR Forced Link Flooding Examples

Configuring the Inter-AS Link as a Passive Interface Example

For this example, see the figure above.

Routers R2 and R4 have the following router IDs:

- Router R2--10.10.0.2
- Router R4--10.10.0.4

```
Router> enable
Router# configure terminal
Router(config)# interface serial 2/0
```

Configures OSPF on Router R2 When Its Neighbor Is Running OSPF Too

```
Router(config-if)# mpls traffic-end passive-interface nbr-te-id 10.10.0.4
```

Note Because both routers are running OSPF, the **nbr-igp-id** keyword is not specified.

Specifies That Both Router R2 and Its Neighbor Are Running OSPF (the nbr-igp-id Keyword Is Specified)

```
Router(config-if)# mpls traffic-eng passive-interface nbr-te-id 10.10.0.4 nbr-igp-id ospf
10.10.0.4
```

Configures IS-IS on Router R1

```
Router(config-if)# mpls traffic-eng passive-interface nbr-te-id 10.10.0.4 nbr-igp-id isis
40.0000.0002.0001.00
```

Configures the Neighbor IGP ID (nbr-igp-id) When There Is More than One Neighbor Specified on a Link

```
Router(config-if)# mpls traffic-end passive-interface nbr-te-id 10.10.0.4 nbr-igp-id ospf
10.10.0.4
Router(config-if)# mpls traffic-eng passive-interface nbr-te-id 10.10.0.7 nbr-igp-id ospf
10.10.0.7
```

Overrides the Interior Gateway Protocol (IGP) Administrative Weight of the Link and Assigns a Specific Weight

```
Router(config-if)# mpls traffic-eng administrative-weight 20
```

Note The ID is unique for each neighbor.

Configures a Link as a Passive Interface (Includes Global TE Commands)

```
interface serial 2/0
ip address 10.10.4.1.255.255.255.0
 mpls traffic-eng tunnels
 mpls traffic-eng administrative-weight 10
 mpls traffic-eng passive-interface nbr-te-id 10.10.0.4 nbr-igp-id ospf 10.10.0.4
 ip rsvp bandwidth 1000
 mpls traffic-eng administrative-weight 20
```

Creating LSPs Traversing the ASBRs Example

In the following example, a primary LSP is created:

```
Router> enable
Router# configure terminal
Router(config)# ip explicit path route1 enable
Router(config)# next-address loose 10.10.0.2
Router(config)# next-address loose 10.10.0.4
Router(config)# next-address loose 10.10.0.6
Router(config)# interface tunnel 100
Router(config-if)# tunnel mpls traffic-eng fast reroute
Router(config-if)# tunnel mpls traffic-eng path-option 1 route1
```

In the following example, a backup LSP is created:

```
Router> enable
Router# configure terminal
Router(config)# ip explicit path backpath1 enable
Router(config)# next-address loose 10.10.0.3
Router(config)# next-address loose 10.10.0.5
Router(config)# next-address loose 10.10.0.6
```

```
Router(config)# interface tunnel 102
Router(config)# mpls traffic-eng backup path tunnel 102
Router(config-if)# tunnel mpls traffic-eng path-option 1 backpath1
```

Configuring Multiple Neighbors on a Link Example

In the following example, there is more than one neighbor on a link:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 2/0
Router(config-if)# mpls traffic-eng passive-interface nbr-te-id 10.10.0.4 nbr-igp-id ospf
10.10.0.4
Router(config-if)# mpls traffic-eng administrative-weight 20
```

Additional References

Related Documents

Related Topic	Document Title
MPLS traffic engineering commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Multiprotocol Label Switching Command Reference*
Fast Reroute	MPLS TE: Link and Node Protection, with RSVP Hellos Support (with Fast Tunnel Interface Down Detection)
Link flooding and node protection	MPLS Traffic Engineering: Interarea Tunnels
IS-IS configuration tasks	Configuring a Basic IS-IS Network
OSPF configuration tasks	Configuring OSPF
IS-IS and OSPF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing Protocols Command Reference*
RSVP	RSVP Message Authentication

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3209	Extensions to RSVP for LSP Tunnels
draft-ietf-mpls-rsvp-lsp-fastreroute-02.txt	*Fast Reroute Extensions to RSVP-TE for LSP Tunnels*
draft-vasseur-mpls-loose-path-reopt-02.txt	*Reoptimization of an Explicitly Loosely Routed MPLS TE Path*
draft-vasseur-mpls-inter-as-te-00.txt	*MPLS Inter-AS Traffic Engineering*
draft-ietf-mpls-soft-preemption-00.txt	*MPLS Traffic Engineering Soft Preemption*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering Inter-AS TE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 188: Feature Information for MPLS Traffic Engineering: Inter-AS TE

Feature Name	Releases	Feature Information
MPLS Traffic Engineering: Inter-AS TE	12.0(29)S 12.2(33)SRA 12.2(33)SRB 12.2(33)SXH 12.4(20)T Cisco IOS XE Release 3.5S	The MPLS Traffic Engineering: Inter-AS TE feature provides ASBR node protection, loose path reoptimization, SSO recovery of LSPs that include loose hops, ASBR forced link flooding, Cisco IOS RSVP local policy extensions for Inter-AS, and per-neighbor key capabilities. In 12.0(29)S, this feature was introduced. In 12.2(33)SRA, the **nbr-if-addr** keyword was added to the **mpls traffic-eng passive-interface** command. In 12.2(33)SRB, support was added for SSO recovery of LSPs that include loose hops. In 12.2(33)SXH, this feature was integrated into Cisco IOS Release 12.2(33)SXH. In 12.4(20)T, this feature was integrated into Cisco IOS Release 12.4(20)T. In Cisco IOS XE Release 3.5S, this feature was integrated into Cisco IOS XE Release 3.5S.

Glossary

ABR --Area Border Router. A routers connecting two areas.

adjacency --The MPLS TE Forwarding Adjacency feature allows a network administrator to handle a traffic engineering, label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network based on the Shortest Path First (SPF) algorithm. A forwarding adjacency can be created between routers regardless of their location in the network. The routers can be located multiple hops from each other.

area --A logical set of network segments (for example, one that is OSPF-based) and their attached devices. Areas usually are connected to other areas by routers, making up a single autonomous system. OSPF and IS-IS define their areas differently. OSPF area borders are marked by routers. Some interfaces are in one area, and other interfaces are in another area. With IS-IS, all the routers are completely within an area, and the area borders are on links, not on routers. The routers that connect the areas are level-2 routers, and routers that have no direct connectivity to another area are level-1 routers.

ASBR --Autonomous System Boundary Router. The router is located between an OSPF autonomous system and a non-OSPF network. ASBRs run both OSPF and another routing protocol, such as RIP. ASBRs must reside in a nonstub OSPF area.

autonomous system --A collection of networks under a common administration sharing a common routing strategy. Autonomous systems are subdivided by areas.

backup tunnel --An MPLS traffic engineering tunnel used to protect other (primary) tunnel's traffic when a link or node failure occurs.

BGP --Border Gateway Protocol. Interdomain routing protocol that replaces EGP. BGP exchanges reachability information with other BGP systems.

border router --A router at the edge of a provider network that interfaces to another provider's border router using extended BGP procedures.

Cisco Express Forwarding --A means for accelerating the forwarding of packets within a router, by storing route lookup information in several data structures instead of in a route cache.

Fast Reroute --A mechanism for protecting MPLS traffic engineering (TE) LSPs from link and node failure by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

flooding --A traffic-passing technique used by switches and bridges in which traffic received on an interface is sent out all the interfaces of that device except the interface on which the information was received originally.

forwarding adjacency --A traffic engineering link (or LSP) into an IS-IS or OSPF network.

headend --The router that originates and maintains a given LSP. This is the first router in the LSP's path.

hop --Passage of a data packet between two network nodes (for example, between two routers).

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

Inter-AS LSP --An MPLS traffic engineering label-switched path (LSP) that traverses hops that are not in the headend's topology database (that is, it is not in the same OSPF area, IS-IS area, or autonomous system as the headend).

interface --A network connection.

IP explicit path --A list of IP addresses, each representing a node or link in the explicit path.

IS-IS --Intermediate System-to-Intermediate System. OSI link-state hierarchal routing protocol based on DECnet Phase V routing, where intermediate system (IS) routers exchange routing information based on a single metric to determine the network topology.

link --A point-to-point connection between adjacent nodes.

LSA --link-state advertisement. A broadcast packet used by link-state protocols that contains information about neighbors and path costs. LSAs are used by the receiving routers to maintain their routing tables.

LSP --label-switched path. A configured connection between two routers, in which MPLS is used to carry packets. An LSP is a path created by the concatenation of one or more label-switched hops, allowing a packet to be forwarded by swapping labels from an MPLS node to another MPLS node.

midpoint --A transit router for a given LSP.

midpoint reoptimization --Ability of a midpoint to trigger a headend reoptimization.

MP --merge point. The LSR where one or more backup tunnels rejoin the path of the protected LSP, downstream of the potential failure. An LSR can be both an MP and a PLR simultaneously.

MPLS --Multiprotocol Label Switching. Packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

multicast --Single packets are copied by the network and sent to a specific subset of network addresses. These addresses are specified in the Destination address field. (Multicast is an efficient paradigm for transmitting the same data to multiple receivers, because of its concept of a Group address. This allows a group of receivers to listen to the single address.)

node --Endpoint of a network connection or a junction common to two or more lines in a network. Nodes can be interconnected by links, and serve as control points in the network.

OSPF --Open Shortest Path First. A link-state, hierarchical Interior Gateway Protocol routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

opaque LSA --If a router understands LSA Type 10 link information, the router continues flooding the link throughout the network.

passive link --When IGP is not running on the link between two ASBRs, traffic engineering informs the IGP to flood link information on behalf of that link (that is, it advertises that link).

PLR --point of local repair. The headend LSR of a backup tunnel.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RSVP --Resource Reservation Protocol. An IETF protocol used for signaling requests (setting up reservations) for Internet services by a customer before that customer is permitted to transmit data over that portion of the network.

SPF --shortest path first. A routing algorithm used as the basis for OSPF operations. When an SPF router is powered up, it initializes its routing-protocol data structures and then waits for indications from lower-layer protocols that its interfaces are functional.

SRLG --Shared Risk Link Group. Sets of links that are likely to go down together (for example, because they have the same underlying fiber).

tailend --The router upon which an LSP is terminated. This is the last router in the LSP's path.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

TLV --type, length, values. A block of information embedded in Cisco Discovery Protocol advertisements.

CHAPTER 108

Configuring MPLS Traffic Engineering over GRE Tunnel Support

The MPLS Traffic Engineering (TE) over Generic Routing Encapsulation (GRE) Tunnel Support feature enables applications to establish TE tunnels over virtual interfaces.

Prerequisites for Configuring MPLS TE over GRE Tunnel Support

Your network must support the following:

- Cisco Express Forwarding
- External data encryptors
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)
- IPsec that is enabled on the GRE nodes to implement GRE traffic encryption
- MPLS TE that is configured on the interface and on GRE tunnels
- MPLS TE tunnels

If GRE tunnels and TE tunnels coexist within the same routing domain, routing loops will occur. Create separate routing domains by either configuring GRE overlay with static routing for GRE packets or using two separate routing processes, one for the GRE overlay and another for TE tunnels.

Restrictions for Configuring MPLS TE Over GRE Tunnel Support

The following TE features are not supported over GRE tunnels, so they should not be configured for TE tunnels that may traverse GRE tunnels:

- The following TE features are not supported over GRE tunnels. They should not be configured for TE tunnels that may traverse GRE tunnels:
  - Autoroute destinations
  - Automatic bandwidth adjustment
  - Autotunnel primary one-hop tunnels
  - Diff-Serve Aware TE (DS-TE)
  - Explicit path options that identify excluded nodes
  - Interarea/autonomous systems MPLS TE
  - Point-to-multipoint TE
  - Shared Risk Link Groups (SRLGs)
  - Tunnel-Based Admission Control (TBAC)
- GRE tunnels do not support Cisco nonstop forwarding with stateful switchover (NSF with SSO). If a switchover occurs, traffic loss occurs for TE over GRE, and the TE tunnels are resignaled.
- Fast Reroute (FRR) is not supported.

Information About Configuring MPLS TE over GRE Tunnel Support

MPLS TE over GRE Tunnel Support Overview

MPLS TE tunnels provide transport for label switching data through an MPLS network using a path, which is constraint-based, and is not restricted to the IGP shortest cost path. The TE tunnels are usually established over physical links between adjacent routers. However, some applications require establishing TE tunnels over virtual interfaces such as GRE tunnels. Federal Information Processing Standard (FIPS) 140-2 compliance mandates that federal customers require traffic encryption throughout their network infrastructure, which is referred to as Type-I encryption level of security. Type-I encryption environments differentiate between encrypted and unencrypted networks. The encrypted network is the secure part of the network that is in a secure facility, where encryption is not required. The unencrypted network is the unsecured part of the network where traffic encryption is required.

Two common methods of traffic encryption are as follows:

- External crypto devices
- Cisco IOS IPsec, which is the encryption embedded into Cisco IOS software

External crypto devices operate in Layer 2 (L2), providing link layer encryption of ATM and SONET traffic. Due to the migration of L2 networks to IP network, there is an increasing adoption of IP crypto devices and IPsec. This transition requires that the traffic encryption happens at the IP layer. The IP-based forwarding of service traffic, such as IP or Layer 3 (L3)/L2 VPN MPLS traffic, is implemented only through GRE tunnels.

The following MPLS TE features are supported when enabled over GRE tunnel:

- MPLS TE over GRE (Tunnel establishment and data traffic)
- Metrics (admin weight)
- Attribute flag and affinities
- Explicit path
- BFD
- ECMP without Class Based Tunnel Selection (CBTS)

Benefits of MPLS TE over GRE Tunnel Support

The MPLS TE Over GRE Tunnel Support feature enables you to leverage MPLS segmentation capabilities, such as Layer 2 and Layer 3 VPN, on GRE tunnel transport. This feature enables you to deploy MPLS TE to implement explicit path forwarding, FRR, and bandwidth management of traffic over GRE tunnels. Also, this feature helps maintain the TE capabilities currently supported by ATM legacy networks.

How to Configure MPLS TE over GRE Tunnel Support

Configuring Resource Reservation Protocol Bandwidth

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bandwidth** *kbps*
5. **ip address** *ip-address mask*
6. **mpls traffic-eng tunnels**
7. **tunnel source** *type number*
8. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
9. **ip rsvp bandwidth**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 0`	Configures a tunnel interface and enters interface configuration mode for the specified tunnel interface.
Step 4	**bandwidth** *kbps* **Example:** `Router(config-if)# bandwidth 100000`	Sets the total bandwidth for a bandwidth pool.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.0.0 255.255.255.254`	Configures a primary IP address for an interface.
Step 6	**mpls traffic-eng tunnels** **Example:** `Router(config-if)# mpls traffic-eng tunnels`	Enables traffic engineering tunnel signaling on the interface.
Step 7	**tunnel source** *type number* **Example:** `Router(config-if)# tunnel source loopback 1`	Configures the source address for the tunnel interface.
Step 8	**tunnel destination** {*host-name* | *ip-address* | *ipv6-address*} **Example:** `Router(config-if)# tunnel destination 192.168.1.1`	Specifies the destination for a tunnel. • *ip-address*—IP address of the host destination expressed in dotted decimal notation.
Step 9	**ip rsvp bandwidth** **Example:** `Router(config-if)# ip rsvp bandwidth`	Enables Resource Reservation Protocol (RSVP) for IP on an interface.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Configuring an MPLS TE Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *tunnel number*
4. **ip unnumbered** *type number*
5. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
6. **mpls traffic-eng tunnels**
7. **tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*]
8. **tunnel mpls traffic-eng bandwidth** *kbps*
9. **tunnel mpls traffic-eng path-option** *number* **dynamic**
10. **tunnel mpls traffic-eng fast-reroute**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *tunnel number* **Example:** `Router(config)# interface tunnel 10`	Configures a tunnel interface and enters interface configuration mode for the specified tunnel interface.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Assigns an IP address to the tunnel interface. • An MPLS TE tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** {*host-name* | *ip-address* | *ipv6-address*} **Example:** `Router(config-if)# tunnel destination 192.168.2.2`	Specifies the destination for a tunnel. • *ip-address*—IP address of the host destination expressed in dotted decimal notation.
Step 6	**mpls traffic-eng tunnels** **Example:**	Enables traffic engineering tunnel signaling on the interface.

	Command or Action	Purpose
	`Router(config-if)# mpls traffic-eng tunnels`	
Step 7	**tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*] **Example:** `Router(config-if)# tunnel mpls traffic-eng priority 7 7`	Configures the setup and reservation priority for the tunnel.
Step 8	**tunnel mpls traffic-eng bandwidth** *kbps* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 10`	Configures the bandwidth required for the tunnel.
Step 9	**tunnel mpls traffic-eng path-option** *number* **dynamic** **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 10 dynamic`	Configures the path option for the tunnel.
Step 10	**tunnel mpls traffic-eng fast-reroute** **Example:** `Router(config-if)# tunnel mpls traffic-eng fast-reroute`	Enables an MPLS TE tunnel to use an established backup tunnel in the event of a link or node failure.
Step 11	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Configuring an MPLS TE Tunnel over GRE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *tunnel* **number**
4. **ip unnumbered loopback** *number*
5. **tunnel destination** *ip-address*
6. **tunnel mpls traffic-eng autoroute announce**
7. **tunnel mpls traffic-eng**
8. **tunnel mpls traffic-eng path-option** *number* **dynamic**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *tunnel* **number** **Example:** `Router(config)# interface tunnel 100`	Configures an interface type and enters interface configuration mode
Step 4	**ip unnumbered loopback** *number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Assigns an IP address to the tunnel interface. • An MPLS TE tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 10.255.1.2`	Specifies the destination for a tunnel. • *ip-address*—IP address of the host destination expressed in dotted decimal notation.
Step 6	**tunnel mpls traffic-eng autoroute announce** **Example:** `Router(config-if)# tunnel mpls traffic-eng autoroute announce`	Specifies that the IGP should use the tunnel in its enhanced shortest path first (SPF) calculation.
Step 7	**tunnel mpls traffic-eng** **Example:** `Router(config-if)# tunnel mpls traffic-eng`	Sets the encapsulation mode of the tunnel to MPLS TE.
Step 8	**tunnel mpls traffic-eng path-option** *number* **dynamic** **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 10 dynamic`	Configures a path option for the MPLS TE tunnel. • If you specify the **dynamic** keyword, the Cisco IOS software checks both the physical bandwidth of the interface and the available TE bandwidth to make sure that the requested amount of bandwidth does not exceed the physical bandwidth of any link.
Step 9	**end** **Example:**	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

Configuration Examples for MPLS TE Over GRE Tunnel Support

Example Configuring MPLS TE Over GRE Tunnel Support

The following example shows how to configure MPLS TE over a GRE tunnel between two routers: Router 1 and Router 2. The first loopback interface is used for router identification, and the other for reachability. One OSPF is used for TE and the other for reachability.

Router 1

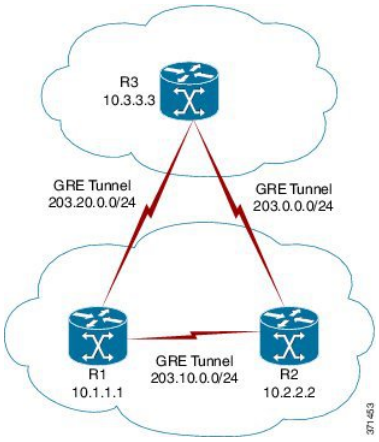
```
configure terminal
no logging console
mpls traffic-eng tunnels
interface Loopback 0
 ip address 172.16.1.1 255.255.255.255
 no shutdown
!
interface Loopback 1
 ip address 10.255.1.1 255.255.255.0
 no shutdown
!
interface gigabitethernet 1/1
 ip address 172.16.1.1 255.255.255.255
 ip rsvp bandwidth 100000
 no shutdown
!
router ospf 172
 router-id 172.16.1.1
 network 172.16.0.0 0.0.255.255 area 0
 mpls traffic-eng router-id Loopback 0
 mpls traffic-eng area 0
 no shutdown
!
router ospf 10
 router-id 10.255.1.1
 network 10.255.0.0 0.0.255.255 area 0
 no shutdown
!
interface Tunnel 10
bandwidth 20000
 ip address 172.16.0.1 255.255.255.252
 mpls traffic-eng tunnels
 keepalive 10 3
 tunnel source Loopback 1
 tunnel destination 10.255.1.2
 ip rsvp bandwidth 15000 sub-pool 5000
!
!
interface tunnel 100
ip unnumbered loopback 0
tunnel mode mpls traffic-eng
tunnel destination 192.168.10.10
```

```
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 10 dynamic
!
end
Router 2
configure terminal
no logging console
mpls traffic-eng tunnels
interface Loopback 0
 ip address 172.16.1.2 255.255.255.255
 no shutdown
!
interface Loopback 1
 ip address 10.255.1.2 255.255.255.255
 no shutdown
!
interface gigabitethernet 1/1
 ip address 10.255.0.2 255.255.255.252
 ip rsvp bandwidth 100000
 no shutdown
!
router ospf 172
 router-id 172.16.1.2
 network 172.16.0.0 0.0.255.255 area 0
 mpls traffic-eng router-id Loopback 0
 mpls traffic-eng area 0
 no shutdown
!
router ospf 10
 router-id 10.255.1.2
 network 10.255.0.0 0.0.255.255 area 0
 no shutdown
!
!
interface Tunnel0
bandwidth 20000
 ip address 172.16.0.2 255.255.255.252
 mpls traffic-eng tunnels
 keepalive 10 3
 tunnel source Loopback 1
 tunnel destination 10.255.1.1
 ip rsvp bandwidth 15000 sub-pool 5000
!
!
interface tunnel 100
ip unnumbered loopback 0
tunnel mode mpls traffic-eng
tunnel destination 172.16.1.1
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 10 dynamic
!
end
```

Example Configuring CBTS with MPLS over GRE

The following example shows how to configure Class-Based Tunnel Selection (CBTS) with MPLS Traffic Engineering (TE) over GRE.

Figure 190: The Network Structure of CBTS with MPLS over GRE



Configuration of the Midpoint Router (R1)

```
mpls traffic-eng tunnels
!
interface Tunnel 102
ip address 203.20.0.1 255.255.255.0
mpls ip
mpls traffic-eng tunnels
tunnel source GigabitEthernet 0/0/0
tunnel destination 192.168.0.1
tunnel key 22
tunnel checksum
ip rsvp bandwidth 500000
!
interface Tunnel 103
ip address 203.10.0.1 255.255.255.0
mpls ip
mpls traffic-eng tunnels
tunnel source GigabitEthernet 0/0/0
tunnel destination 192.168.10.1
tunnel key 33
tunnel checksum
ip rsvp bandwidth 500000
mpls traffic-eng tunnels
!
router ospf 1
router-id 10.1.1.1
network 10.1.1.1 0.0.0.0 area 1
network 203.20.0.1 0.0.0.0 area 1
network 203.10.0.1 0.0.0.0 area 1
mpls traffic-eng router-id Loopback 0
mpls traffic-eng area 1
```

Configuration of the Head Router (R2)

```
mpls traffic-eng tunnels
!
interface Tunnel 203
 ip address 203.0.0.1 255.255.255.0
 mpls ip
 mpls traffic-eng tunnels
```

```
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 192.168.10.1
 tunnel key 6
 tunnel checksum
 ip rsvp bandwidth 500000
!
interface Tunnel 211
 ip address 172.16.0.2 255.255.255.0
 mpls ip
 mpls traffic-eng tunnels
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 192.168.20.1
 tunnel key 22
 tunnel checksum
 ip rsvp bandwidth 500000
!
interface Tunnel 2300
 ip unnumbered Loopback 0
 tunnel mode mpls traffic-eng
 tunnel destination 10.3.3.3
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng autoroute metric relative -5
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 10 dynamic
 tunnel mpls traffic-eng exp-bundle master
 tunnel mpls traffic-eng exp-bundle member Tunnel 2301
 tunnel mpls traffic-eng exp-bundle member Tunnel 2302
!
interface Tunnel 2301
 ip unnumbered Loopback 0
 tunnel mode mpls traffic-eng
 tunnel destination 10.3.3.3
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng autoroute metric relative -5
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 10 explicit name TE2301
 tunnel mpls traffic-eng exp 6 7
!
interface Tunnel 2302
 ip unnumbered Loopback 0
 tunnel mode mpls traffic-eng
 tunnel destination 10.3.3.3
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng autoroute metric relative -5
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1000
 tunnel mpls traffic-eng path-option 10 explicit name TE2302
 tunnel mpls traffic-eng exp default
!
router ospf 1
 router-id 10.2.2.2
 network 10.2.2.2 0.0.0.0 area 1
 network 203.20.0.2 0.0.0.0 area 1
 network 172.16.0.2 0.0.0.0 area 1
 network 203.0.0.1 0.0.0.0 area 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 1
!
ip explicit-path name TE2301 enable
 next-address 203.0.0.2
ip explicit-path name TE2302 enable
 next-address 172.16.0.1
```

```
 next-address 172.26.0.2
```

Configuration of the Tail Router (R3)

```
mpls traffic-eng tunnels
!
interface Tunnel 302
 ip address 203.0.0.2 255.255.255.0
 mpls ip
 mpls traffic-eng tunnels
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 192.168.0.1
 tunnel key 6
 tunnel checksum
 ip rsvp bandwidth 500000
!
interface Tunnel 311
 ip address 172.26.0.2 255.255.255.0
 mpls ip
 mpls traffic-eng tunnels
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 192.168.20.1
 tunnel key 33
 tunnel checksum
 ip rsvp bandwidth 500000
 !
router ospf 1
 router-id 10.3.3.3
 network 10.3.3.3 0.0.0.0 area 1
 network 203.10.0.2 0.0.0.0 area 1
 network 172.26.0.2 0.0.0.0 area 1
 network 203.0.0.2 0.0.0.0 area 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 1
!
```

Additional References for MPLS TE Over GRE Tunnel Support

Related Documents

Related Topic	Document Title
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards

Standard	Title
FIPS 140-2	Security Requirements for Cryptographic Modules.

MIBs

MIB	MIBs Link
MPLS-TE-STD-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3812	MPLS TE Management Information Base (MIB)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS TE Over GRE Tunnel Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 189: Feature Information for MPLS TE over GRE Tunnel Support

Feature Name	Releases	Feature Information
MPLS TE over GRE Tunnel Support	Cisco IOS XE Release 3.3S 15.2(1)T Cisco IOS XE Release 3.12S Cisco IOS XE Release 3.16S	The MPLS TE over GRE Tunnel Support feature enables applications to establish traffic engineering tunnels over virtual interfaces. The following commands were introduced or modified: **mpls traffic-eng tunnels**, **tunnel mpls traffic-eng autoroute announce**. **tunnel mpls traffic-eng bandwidth**, **tunnel mpls traffic-eng fast-reroute**, **tunnel mpls traffic-eng path-option**, **tunnel mpls traffic-eng priority**. In Cisco IOS XE 3.12S release, CBTS support was added for GRE interface type on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE 3.16S release,CBTS support was added for GRE interface type on Cisco ISR4451/4431/4351 series Integrated Services Routers.

CHAPTER 109

MPLS Traffic Engineering—RSVP Graceful Restart

The MPLS Traffic Engineering—RSVP Graceful Restart feature allows a neighboring Route Processor (RP) to recover from disruption in control plane service (specifically, the Label Distribution Protocol (LDP) component) without losing its Multiprotocol Label Switching (MPLS) forwarding state. This feature has the following benefits:

- Graceful restart allows a node to recover state information from its neighbor when there is an RP failure or the device has undergone a stateful switchover (SSO).
- Graceful restart allows session information recovery with minimal disruption to the network.
- A node can perform a graceful restart to help a neighbor recover its state by keeping the label bindings and state information to provide a quick recovery of the failed node and not affect the traffic that is currently forwarded.

Prerequisites for MPLS TE—RSVP Graceful Restart

Perform the following tasks on routers before configuring the MPLS Traffic Engineering—RSVP Graceful Restart feature:

- Configure the Resource Reservation Protocol (RSVP).
- Enable MPLS.
- Configure traffic engineering (TE).
- Enable graceful restart.

Restrictions for MPLS TE—RSVP Graceful Restart

- Graceful restart supports node failure only.
- Cisco recommends that you configure interface hellos only if the neighbor router does not support node hellos.
- Unnumbered interfaces are not supported.
- You cannot configure an interface hello for graceful restart and an interface hello for Fast ReRoute or hello state timeout (HST) on the same interface.

Information About MPLS TE—RSVP Graceful Restart

Graceful Restart Operation

RSVP graceful restart allows RSVP TE enabled nodes to recover gracefully following a node failure in the network such that the RSVP state after the failure is restored as quickly as possible. The node failure may be completely transparent to other nodes in the network.

RSVP graceful restart preserves the label values and forwarding information and works with third-party or Cisco routers seamlessly.

RSVP graceful restart depends on RSVP hello messages to detect that a neighbor went down. Hello messages include Hello Request or Hello Acknowledgment (ACK) objects between two neighbors.

A node hello is transmitted when graceful restart is globally configured and the first LSP to the neighbor is created.

Interface hello is an optional configuration. If you configure the graceful restart Hello command on an interface, the interface hello is considered to be an additional hello instance with the neighbor.

The router transmits an interface hello for graceful restart when all of the following conditions are met:

- Graceful restart is configured globally.
- Graceful restart is configured on the interface.
- An LSP to the neighboring router is created and goes over the interface.

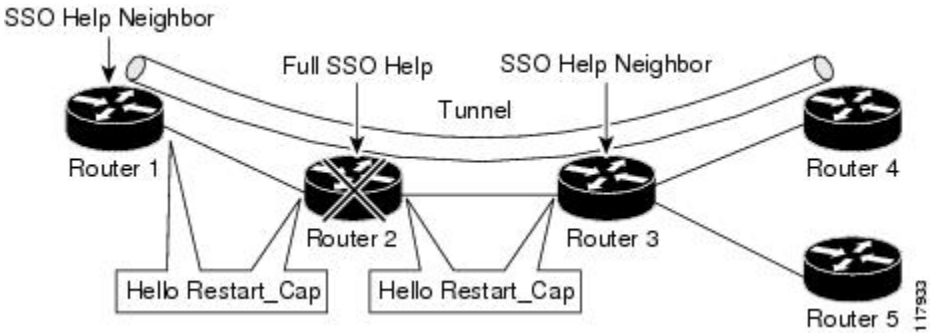
Cisco recommends that you use node hellos if the neighbor supports node hellos, and configure interface hellos only if the neighbor router does not support node hellos.

Interface hellos differ from node hellos. as follows:

- **Interface hello** —The source address in the IP header of the hello message has an IP address that matches the interface that the Hello message sent out. The destination address in the IP header is the interface address of the neighbor on the other side of the link. A TTL of 1 is used for per-interface hellos as it is destined for the directly-connected neighbor.
- **Node hello** —The source address in the IP header of the Hello message includes the TE router ID of the sending router. The destination address of the IP header has the router ID of the neighbor to which this message is sent. A TTL of more than 1 is used.

The figure below shows the graceful restart extension to these messages that an object called Restart_Cap, which tells neighbors that a node, may be capable of restarting if a failure occurs. The time-to-live (TTL) in these messages is set to 255 so that adjacencies can be maintained through alternate paths even if the link between two neighbors goes down.

Figure 191: How Graceful Restart Works



The Restart_Cap object has two values—the restart time, which is the sender's time to restart the RSVP_TE component and exchange hello messages after a failure; and the recovery time, which is the desired time that the sender wants the receiver to synchronize the RSVP and MPLS databases.

In the figure above, graceful restart is enabled on Router 1, Router 2, Router 3, and Router 4. For simplicity, assume that all routers are restart capable. A TE label switched path (LSP) is signaled from Router 1 to Router 4.

Router 2 and Router 3 exchange periodic graceful restart hello messages every 10000 ms (10 seconds), and so do Router 2 and Router 1 and Router 3 and Router 4. Assume that Router 2 advertises its restart time as 60000 ms (60 seconds) and its recovery time as 60000 ms (60 seconds) as shown in the following example:

```
23:33:36: Outgoing Hello:
23:33:36:   version:1 flags:0000 cksum:883C ttl:255 reserved:0 length:32
23:33:36:   HELLO               type HELLO REQUEST length 12:
23:33:36:   Src_Instance: 0x6EDA8BD7, Dst_Instance: 0x00000000
23:33:36:   RESTART_CAP         type 1 length 12:
23:33:36:   Restart_Time: 0x0000EA60
, Recovery_Time: 0x0000EA60
```

Note The restart and recovery time are shown in **bold** in the last entry.

Router 3 records this into its database. Also, both neighbors maintain the neighbor status as UP. However, Router 3's control plane fails at some point (for example, a Primary Route Processor failure). As a result, RSVP and TE lose their signaling information and states although data packets continue to be forwarded by the line cards.

When four ACK messages are missed from Router 2 (40 seconds), Router 3 declares communication with Router 2 lost "indicated by LOST" and starts the restart time to wait for the duration advertised in Router 2's restart time previously and recorded (60 seconds). Router 1 and Router 2 suppress all RSVP messages to Router 3 except hellos. Router 3 keeps sending the RSVP Path and Resv refresh messages to Router 4 and Router 5 so that they do not expire the state for the LSP; however, Router 3 suppresses these messages for Router 2.

Note A node restarts if it misses four ACKs or its hello src_instance (last source instance sent to its neighbor) changes so that its restart time = 0.

Before the restart time expires, Router 2 restarts and loads its configuration and graceful restart makes the configuration of Router 2 send the hello messages with a new source instance to all the data links attached. However, because Router 2 has lost the neighbor states, it does not know what destination instance it should use in those messages; therefore, all destination instances are set to 0.

When Router 3 sees the hello from Router 2, Router 3 stops the restart time for Router 2 and sends an ACK message back. When Router 3 sees a new source instance value in Router 2's hello message, Router 3 knows that Router 2 had a control plane failure. Router 2 gets Router 3's source instance value and uses it as the destination instance going forward.

Router 3 also checks the recovery time value in the hello message from Router 2. If the recovery time is 0, Router 3 knows that Router 2 was not able to preserve its forwarding information and Router 3 deletes all RSVP state that it had with Router 2.

If the recovery time is greater than 0, Router 1 sends Router 2 Path messages for each LSP that it had previously sent through Router 2. If these messages were previously refreshed in summary messages, they are sent individually during the recovery time. Each of these Path messages includes a Recovery_Label object containing the label value received from Router 2 before the failure.

When Router 3 receives a Path message from Router 2, Router 3 sends a Resv message upstream. However, Router 3 suppresses the Resv message until it receives a Path message.

How to Configure MPLS TE—RSVP Graceful Restart

Enabling Graceful Restart

Note It is optional that you configure graceful restart on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart mode help-neighbor**
4. **interface** *type number*
5. **ip rsvp signalling hello graceful-restart**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart mode help-neighbor** **Example:** `Router(config)# ip rsvp signalling hello graceful-restart mode help-neighbor`	Sets the number of DSCP hello messages on a neighboring router with restart capability.
Step 4	**interface** *type number* **Example:** `Router(config)# interface POS 1/0/0`	(Optional) Configures the interface type and number and enters interface configuration mode.
Step 5	**ip rsvp signalling hello graceful-restart** **Example:** `Router(config-if)# ip rsvp signalling hello graceful-restart`	(Optional) Enables RSVP TE graceful restart capability on a neighboring router.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Setting a DSCP Value

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart dscp** *num*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart dscp** *num* **Example:** `Router(config)# ip rsvp signalling hello graceful-restart dscp 30`	Sets the number of DSCP hello messages on a graceful restart-enabled router.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Setting a Hello Refresh Interval

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart refresh interval** *interval-value*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart refresh interval** *interval-value* **Example:**	Sets a hello refresh interval on a router with graceful restart enabled.

	Command or Action	Purpose
	`Router(config)# ip rsvp signalling hello graceful-restart refresh interval 5000`	
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Setting a Missed Refresh Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling hello graceful-restart refresh misses** *msg-count*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling hello graceful-restart refresh misses** *msg-count* **Example:** `Router(config)# ip rsvp signalling hello graceful-restart refresh misses 5`	Sets a refresh limit on a router with graceful restart enabled.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Verifying Graceful Restart Configuration

SUMMARY STEPS

1. **enable**
2. **show ip rsvp hello graceful-restart**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp hello graceful-restart** **Example:** `Router# show ip rsvp hello graceful-restart`	Displays information about the status of graceful restart and related parameters.
Step 3	**end** **Example:** `Router# end`	Exits to user EXEC mode.

Configuration Examples for MPLS TE—RSVP Graceful Restart

MPLS TE—RSVP Graceful Restart Example

In the following example, graceful restart is enabled, and related parameters, including a DSCP value, a refresh interval, and a missed refresh limit are set:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp signalling hello graceful-restart mode help-neighbor
Router(config)# ip rsvp signalling hello graceful-restart dscp 30
Router(config)# ip rsvp signalling hello graceful-restart refresh interval 10000
Router(config)# ip rsvp signalling hello graceful-restart refresh misses 4
Router(config)# end
```

The following example verifies the status of graceful restart and the configured parameters:

```
Router# show ip rsvp hello graceful-restart
Graceful Restart:Enabled (help-neighbor only)
  Refresh interval:10000 msecs
  Refresh misses:4
  DSCP:0x30
```

```
Advertised restart time:0 secs
Advertised recovery time:0 secs
Maximum wait for recovery:3600000 secs
```

Additional References

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference
Quality of service (QoS) classification	Classification Overview
QoS signalling	Signalling Overview
QoS congestion management	Congestion Management Overview
Stateful switchover	Stateful Switchover
MPLS Label Distribution Protocol	MPLS Label Distribution Protocol (LDP)
Information on stateful switchover, Cisco nonstop forwarding, graceful restart	NSF/SSO—MPLS TE and RSVP Graceful Restart
RSVP hello state timer	MPLS Traffic Engineering: RSVP Hello State Timer

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBS are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3209	*RSVP-TE: Extensions to RSVP for LSP Tunnels*

RFCs	Title
RFC 3473	*Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions*
RFC 3478	*Graceful Restart Mechanism for Label Distribution*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Traffic Engineering—RSVP Graceful Restart

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 190: Feature Information for MPLS Traffic Engineering—RSVP Graceful Restart

Feature Name	Releases	Feature Information
MPLS Traffic Engineering—RSVP Graceful Restart	12.0(29)S 12.2(33)SRE 12.4(20)T Cisco IOS XE Release 2.3	The MPLS TE—RSVP Graceful Restart feature allows a neighboring Route Processor (RP) to recover from disruption in control plane service (specifically, the Label Distribution Protocol (LDP) component) without losing its MPLS forwarding state. In Cisco IOS Release 12.0(29)S, this feature was introduced. In Cisco IOS Release 12.4(20)T, this feature was integrated. The following commands were introduced or modified: **ip rsvp signalling hello graceful-restart dscp**, **ip rsvp signalling hello graceful-restart mode help-neighbor**, **ip rsvp signalling hello graceful-restart refresh interval**, **ip rsvp signalling hello graceful-restart refresh misses**, **show ip rsvp counters**, **show ip rsvp counters state teardown**, **show ip rsvp hello**, **show ip rsvp hello client lsp detail**, **show ip rsvp hello client lsp summary**, **show ip rsvp hello client neighbor detail**, **show ip rsvp hello client neighbor summary**, **show ip rsvp hello graceful-restart**, **show ip rsvp hello instance detail**, **show ip rsvp hello instance summary**. In Cisco IOS Release 12.2(33)SRE, per node hellos allow interoperability with Cisco IOS Release 12.0S.

Glossary

autonomous system—A collection of networks that share the same routing protocol and that are under the same system administration.

ASBR—Autonomous System Boundary Router. A router that connects and exchanges information between two or more autonomous systems.

backup tunnel—An MPLS traffic engineering tunnel used to protect other (primary) tunnels' traffic when a link or node failure occurs.

DSCP—differentiated services code point. Six bits in the IP header, as defined by the IETF. These bits determine the class of service provided to the IP packet.

Fast Reroute—A mechanism for protecting MPLS traffic engineering (TE) LSPs from link and node failure by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

graceful restart—A process for helping a neighboring Route Processor restart after a node failure has occurred.

headend—The router that originates and maintains a given LSP. This is the first router in the LSP's path.

IGP—Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include IGRP, OSPF, and RIP.

instance—A mechanism that implements the RSVP hello extensions for a given router interface address and remote IP address. Active hello instances periodically send Hello Request messages, expecting Hello ACK messages in response. If the expected ACK message is not received, the active hello instance declares that

the neighbor (remote IP address) is unreachable (that is, it is lost). This can cause LSPs crossing this neighbor to be fast rerouted.

label—A short, fixed-length data identifier that tells switching nodes how to forward data (packets or cells).

LDP—Label Distribution Protocol. The protocol that supports MPLS hop-by-hop forwarding by distributing bindings between labels and network prefixes. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LSP—label switched path. A configured connection between two routers, in which MPLS is used to carry packets. A path created by the concatenation of one or more label switched hops, allowing a packet to be forwarded by swapping labels from an MPLS node to another MPLS node.

merge point—The tail of the backup tunnel.

MPLS—Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. MPLS enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels.

PLR—point of local repair. The headend of the backup tunnel.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

RP—Route processor. Processor module in routers that contains the CPU, system software, and most of the memory components that are used in the router. Sometimes called a supervisory processor.

state—Information that a router must maintain about each LSP. The information is used for rerouting tunnels.

tailend—The router upon which an LSP is terminated. This is the last router in the LSP's path.

TE—traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

topology—The physical arrangement of network nodes and media within an enterprise networking structure.

tunnel—Secure communications path between two peers, such as two routers.