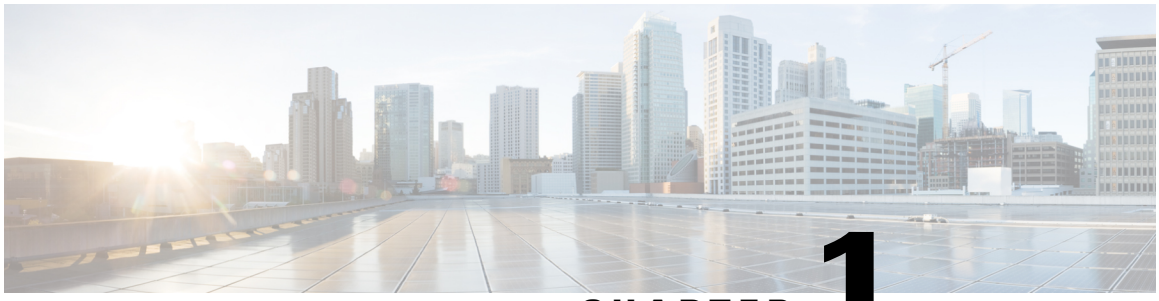
MPLS Basic Configuration Guide, Cisco IOS Release 15S

First Published: November 26, 2012

Last Modified: November 26, 2012

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



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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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.

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

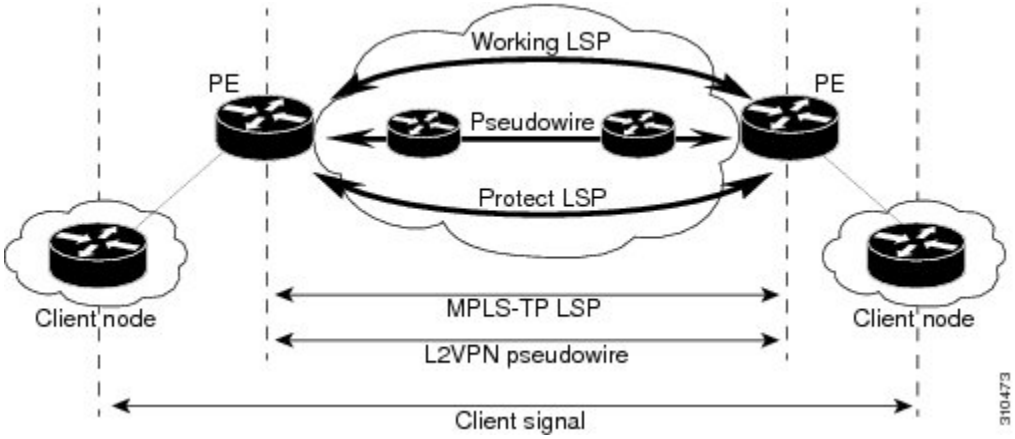
- Ping for static pseudowires over MPLS-TP tunnels is not supported.
- Pseudowire ping and traceroute functionality for multisegment pseudowires that have one or more static pseudowire segments is not supported.
- The following packet format is supported:
    - A labeled packet with Generic Associated 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 19, 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.

	Command or Action	Purpose
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.

	Command or Action	Purpose
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.

	Command or Action	Purpose
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:** `Device(config-pw-class)# preferred-path interface tunnel-tp2`	Specifies the tunnel to use as the preferred path.
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:** `Device# configure terminal`	Enters global configuration mode.
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.

	Command or Action	Purpose
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*]] **Example:** `Device(config-if)# tp destination 10.10.10.10`	Specifies the destination node of the tunnel.
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.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Device(config-if-working)# exit`	Exits working LSP interface configuration mode and returns to interface configuration mode.
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.

	Command or Action	Purpose
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.
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:** `Device# configure terminal`	Enters global configuration mode.
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

	Command or Action	Purpose
		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.

	Command or Action	Purpose
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.
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:** `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-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.

	Command or Action	Purpose
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:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 9	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.

	Command or Action	Purpose
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.
Step 17	Do one of the following: • **tlv** [*type-name*] *type-value length* [**dec** | **hexstr** | **str**] *value* • **tlv template** *template-name*	Specifies the TLV parameters or invokes a previously configured TLV template.

	Command or Action	Purpose
	Example: `Device(config-vfi-neighbor)# tlv statictemp 2 4 hexstr 1`	
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

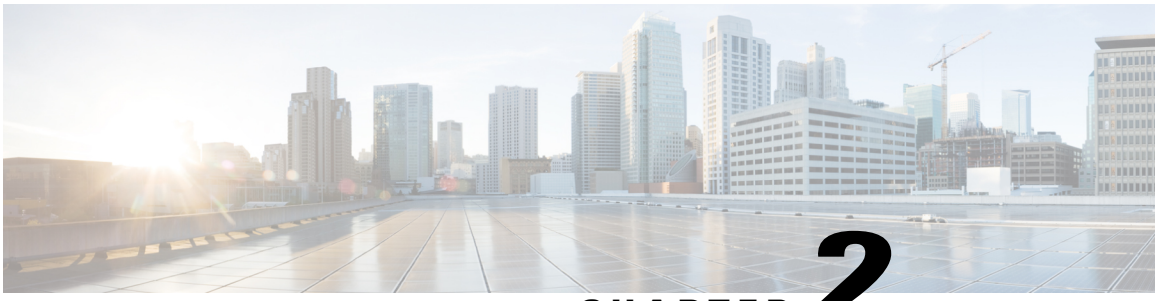
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

Table 1: Feature Information for MPLS Transport Profile

Feature Name	Releases	Feature Information
MPLS Transport Profile • Bidirectional MPLS-TP LSP • L2VPN Static to Dynamic PW Interconnection & PW Preferred Path for MPLS-TP Tunnels • MPLS TP: IP-less Configuration of MPLS TP Tunnels • MPLS-TP OAM: Continuity Check via BFD • MPLS-TP OAM: Fault Management • MPLS-TP OAM: GACH • MPLS-TP Path Protection • MPLS-TP OAM: Ping/Trace • MPLS-TP: PW Redundancy for Static PWs	Cisco IOS XE Release 3.5S	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 SONET and SDH TDM technologies to packet switching to support services with high bandwidth requirements, such as video. 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 l2transport static-oam**, **debug mpls tp**, **interface tunnel-tp interval local**, **interface logging (MPLS-TP)**, **medium p2p**, **mpls tp**, **mpls tp link**, **mpls tp lsp ping**, **notification static timeout refresh**, **pseudowire-static-oam class**, **pseudowire-tlv template**, **show mpls l2transport static-oam**, **show mpls tp status protocol**, **tlv**, **tlv template trace mpls tp**.

Feature Name	Releases	Feature Information
MPLS Transport Profile • MPLS-TP L2VPN Support for MPLS Transport Profile • MPLS-TP OAM: Continuity Check via BFD • MPLS-TP OAM: Fault Management • MPLS-TP OAM: GACH • MPLS-TP Path Protection • MPLS-TP OAM: Ping/Trace	Cisco IOS XE Release 3.10S	In Cisco IOS XE Release 3.10S, support was added for the Cisco ASR 1000 Router.



CHAPTER 2

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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.

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	Specifies static binding of labels to IPv4 prefixes. Bindings specified are installed automatically in the MPLS forwarding table as routing demands.

	Command or Action	Purpose
	Example: `Router(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55`	

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*]	Specifies a range of labels for use with MPLS Static Labels feature.

	Command or Action	Purpose
	Example: `Router(config)# mpls label range 200 100000 static 16 199`	(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 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 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 2: Feature Information for MPLS Static Labels

Feature Name	Releases	Feature Information
MPLS Static Labels	12.0(23)S 12.2(33)SRA 12.2(33)SXH	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 The following commands were introduced or modified: **debug mpls static binding**, **mpls label range**, **mpls static binding ipv4**, **mpls static crossconnect**, **show mpls label range**, **show mpls static binding ipv4**, **show mpls static crossconnect**

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.

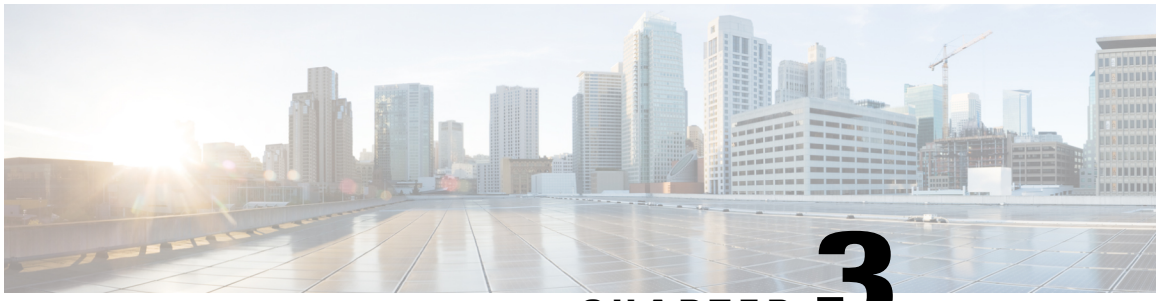
Routing Information Base --See RIB.

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

NetFlow MPLS Label Export

The NetFlow MPLS Label Export feature allows a label switch router (LSR) to collect and export Multiprotocol Label Switching (MPLS) labels allocated by the LSR when an adjacent router pushes that label on the top of the label stack of a transit packet. At the same time, the LSR collects the prefix associated with the MPLS label and the application that allocated the label. The router collects the information in a table called the MPLS Prefix/Application/Label (PAL) table and exports this data to a NetFlow collector as the label is allocated or, if so configured, periodically exports the full MPLS PAL table.

You can use this information to create a provider edge (PE)-to-PE matrix, which is useful for network traffic planning and billing. To realize this benefit, you must export the MPLS label information to a NetFlow collector for analysis. This feature also provides information that a NetFlow collector can use to create a Virtual Private Network (VPN) routing and forwarding instance (VRF)-to-PE and PE-to-VRF matrix.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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.

Prerequisites for NetFlow MPLS Label Export

The NetFlow MPLS Label Export feature requires the following:

- NetFlow configured on the LSR
- MPLS enabled on the LSR

If you are exporting data to a Cisco NetFlow collector, the following requirements apply:

- NetFlow Version 9 export format configured on the LSR
- NetFlow collector and analyzer that can use MPLS PAL records exported in NetFlow Version 9 format

Restrictions for NetFlow MPLS Label Export

The following restrictions apply to the NetFlow MPLS Label Export feature for Cisco IOS 12.2S releases and Cisco IOS Release 12.5(1):

- The MPLS PAL table does not support the export of information for the following:
  - IP Version 6 (IPv6) labels
  - IP Multicast labels
  - Quality of service (QoS) labels
  - Traffic engineering (TE) tunnel headend labels
- The ability to create a VRF-to-VRF traffic matrix is not supported.
- If one application deallocates a label and a second application soon reallocates the same label, the NetFlow collector might not be able to determine how many packets flowed while the label was owned by each application.
- In MPLS PAL table records, for labels allocated by VPNs, Border Gateway Protocol (BGP) IPv4, or BGP VPN Version 4 (VPNv4), the stored prefix can be either 0.0.0.0 or a route distinguisher (RD)-specific address:
  - If you do not configure the **mplsexportvpnv4prefixes** command, VPN prefixes are not tracked in the MPLS PAL table. These prefixes are displayed by the **showmplsflowmappings** command as 0.0.0.0.
  - If you configure the **mplsexportvpnv4prefixes**command, VPN prefixes are tracked and RD-specific addresses are displayed by the **showmplsflowmappings** command.

Information About NetFlow MPLS Label Export

The following sections contain useful information for understanding how to configure and use the NetFlow MPLS Label Export feature:

MPLS Label Information Gathering and Exporting

In a Cisco IOS 12.0S, 12.3T, or 12.4T release that supports the MPLS-Aware NetFlow feature, the mapping of the MPLS label to a prefix and an MPLS application is achieved through the use of the Label Forwarding Information Base (LFIB). You can display this information with the**showipcacheverboseflow**command. These releases do not support the NetFlow MPLS Label Export feature.

In a Cisco IOS 12.2(28)SB release or later release that supports the NetFlow MPLS Label Export feature, the mapping of the MPLS label to a destination prefix or Forwarding Equivalence Class (FEC) and to the MPLS application currently using the label is achieved through the use of an MPLS PAL table. Each supported MPLS application on the router where the NetFlow MPLS Label Export feature is configured registers its label values, prefixes, and owning applications as the labels are allocated. This label-tracking functionality operates on the Route Processor (RP) software.

The MPLS label information (label to prefix and application) mapping is exported to a NetFlow collector at the time when the label is allocated. You can configure periodic export of the full MPLS PAL table to a collector for further processing and analysis through the use of the**mplsexportinterval**command.

An *interval* argument to the**mplsexportinterval** command controls the time in minutes between full MPLS PAL table exports to the NetFlow collector. You can configure an interval in the range of 0 to 10080 (1 week) minutes:

- If you want to export MPLS PAL table information only when the label is allocated, then configure this command with a 0 time interval with the **mplsexportinterval0**command.
- If you want to trigger an immediate export of the full MPLS PAL table, reconfigure the command with an *interval* argument that is different from the interval that is configured. For example, if you have configured the **mplsexportinterval1440**command, reconfigure the command with any nonzero number except 1440.
- If you have a complex network that generates a large amount of traffic, configure a large interval between MPLS PAL table exports. You might want to configure an interval from 6 to 12 hours (360 and 720 minutes).

The *interval* argument that you specify is the least amount of time that passes before another export of the MPLS PAL table occurs. The system could delay the MPLS PAL table export for 10 minutes if the PAL export queue already contains a large number of entries. This could happen if the export occurred at a time when thousands of routes just came up, or if NetFlow did not have the time to clear the export queue from either a previous export of the full table or a previous time when thousands of routes came up in a brief period of time.

After you have entered the **mplsexportinterval** command, you can use the **showmplsflowmappings**command to display MPLS PAL table entries. To display information about the number of MPLS PAL records exported to the collector, use the **showipflowexportverbose**command.

Labels Allocated by VPNs BGP IPv4 or BGP VPNv4 in the MPLS PAL Table

If you want to see VPN prefix information, that is, labels allocated by VPN, BGP IPv4, or BGP VPNv4, you need to configure the **mplsexportvpnv4prefixes** command. If you do not configure the **mplsexportvpnv4prefixes** command, MPLS PAL stores labels allocated by these application as prefix 0.0.0.0.

After you configure the **mplsexportvpnv4prefixes**command, the VPN prefix and the associated RD are stored in the MPLS PAL table. VPN addresses are made unique by adding an RD to the front of the address. The RD removes any ambiguity when the same VPN prefix is used for more than one VRF.

Note To export VPN prefixes and associated RDs from the MPLS PAL table, the first time you configure the **mplsexportvpnv4prefixes**command you need to save the configuration and reboot the router or clear all routes from the table.

To display the VPN prefix entries in the MPLS PAL table, use the **showmplsflowmappings**command.

With the **mplsexportvpnv4prefixes**command configured, a line of the output might look like this:

```
Router# show mpls flow mappings
Label    Owner    Route-Distinguisher Prefix           Allocated
.
.
.
27       BGP      100:1               10.34.0.0         00:57:48
```

The format of the Route-Distinguisher field in the output depends on how the RD was configured. The RD can be configured in the as-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).

If you did not configure the **mplsexportvpnv4prefixes**command, a line of the output looks like this:

```
Router# show mpls flow mappings
.
.
.
Label    Owner    Route-Distinguisher Prefix           Allocated
21       BGP                          0.0.0.0           00:52:18
```

The Route-Distinguisher field is not populated and the Prefix is displayed as 0.0.0.0.

If the MPLS PAL table tracks a per-VRF aggregate label and you configured the **mplsexportvpnv4prefixes**command, the **showmplsflowmappings**command displays the RD associated with the per-VRF aggregate label, but the prefix for the per-VRF aggregate label is reported as 0.0.0.0. If the **mplsexportvpnv4prefixes**command is not configured, the per-VRF aggregate label is reported with no RD and prefix 0.0.0.0, and you cannot distinguish the per-VRF aggregate label from a normal BGP label.

MPLS PAL Table Record Export

In Cisco IOS Release 12.0S and later releases, the export of MPLS-Aware NetFlow cache records makes use of the NetFlow Version 9 export format data and template. The export of MPLS PAL table entries also uses the NetFlow Version 9 export format. MPLS PAL packets are exported as NetFlow options packets rather than NetFlow data packets. NetFlow options packets are defined in *Cisco Systems NetFlow Services Export Version 9,* Request for Comments (RFC) 3954.

The RP on the PE router learns and queues the MPLS PAL table records for export. The RP can combine large numbers of PAL table entries in a single Version 9 record and send the record to the NetFlow collector. The information exported by the RP contains instances of the following for each tracked label:

Label, allocating-application (Owner), Route-Distinguisher, Prefix, time stamp (Allocated)

Because the mapping may change as labels expire and are reused, each PAL record contains a time stamp indicating the system uptime at which the label was allocated.

NetFlow Export Template Format Used for MPLS PAL Entries

This is the NetFlow Version 9 export template format used for MPLS PAL entries:

MPLS label: 3 bytes

MPLS label application type: 1 byte

MPLS label IP prefix: 4 bytes

MPLS VPN prefix RD: 8 bytes

MPLS label allocation time: 4 bytes

MPLS Application Types Exported

The following MPLS application types are exported in the MPLS label application type field:

TE = 1

ATOM = 2

VPN = 3

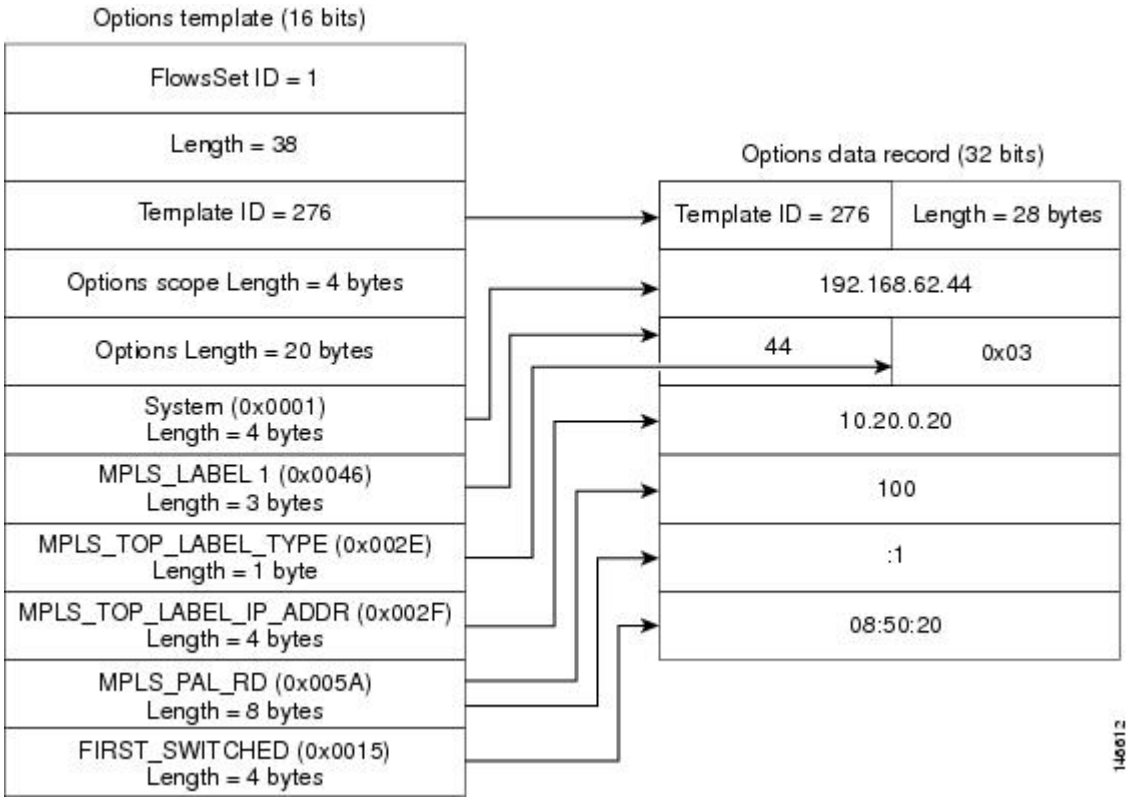
BGP = 4

LDP = 5

Options Template and Options Data Record for MPLS PAL Record Export

The figure below shows an example of the options template and options data record for MPLS PAL record export. This example shows that MPLS label 44 was allocated by a VPN 0x03 at 08:50:20 and is associated with the IP address 10.20.0.20 and with RD 100:1.

Figure 1: MPLS PAL Export Format Record



MPLS PAL and NetFlow Statistics Correlation on a NetFlow Collector

A NetFlow collector can gather the PAL NetFlow packets from a PE router and correlate the label mappings with the recent NetFlow records from adjacent provider core (P) routers.

For example, the MPLS PAL export packet contains MPLS label mappings over a period of time, as each label is allocated and reallocated on the PE router. The packet might contain the following information:

```
label 5, prefix 10.0.0.0,  type LDP, 12:00:00
label 4, prefix 10.10.0.0, type LDP, 13:00:00
label 5, prefix 10.9.0.0,  type VPN, 14:00:00
```

The NetFlow collector then receives a NetFlow packet from the adjacent P router indicating the following:

```
label 5, 123 packets, 9876 bytes, time 12:22:15.
```

The collector would match the time range known from the PAL packets with the line card (LC) packet time stamp. This would result in the correct mapping for label 5 at time 12:22:15, as follows:

```
label 5, application LDP, prefix 10.0.0.0.
```

The NetFlow collector needs to be able to handle relative differences in the time stamps caused by different reboot times of the P and PE routers.

To implement the offline label mapping checks in the NetFlow collector, the collector needs to maintain a history of label mappings obtained from the MPLS PAL NetFlow packets sent by the RP. If a label is deallocated and reallocated, the collector should track both the old and the new MPLS PAL information for the label.

Note On a rare occasion, the collector might not be able to accurately track how many packets flowed for a label that has been deallocated by one application and soon reallocated by another application.

MPLS Label Mapping on a Line Card

Label to prefix and application mapping is registered and exported from the router RP. This functionality does not occur on the line card. If you want to see the mapping for a particular label on a line card and the label of interest is tracked by the MPLS PAL table, then you can do the following:

- Enter the **showmplsforwarding** command on the line card.
- Enter the **showmplsflowmappings** on the RP.
- Compare the output of the two commands.

You might find the**|include**keyword to the commands useful in this case. For example, You could enter the**showmplsflowmappings|include777** command to see the information for any label with substring 777.

How to Configure NetFlow MPLS Label Export

Perform the following tasks to configure the NetFlow MPLS Label Export feature on an LSR. This feature provides the label, prefix, and application mapping through the MPLS PAL table that collects and exports the data to a NetFlow collector.

Configuring NetFlow MPLS Label Export and MPLS PAL Table Export

Perform this task to configure the NetFlow MPLS Label Export feature and MPLS PAL table export to a NetFlow collector. You can use the information generated for network traffic planning and billing.

The following task must be completed before MPLS labels are allocated by the router for the MPLS PAL table to be exported to a NetFlow collector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls export interval** *interval*
4. **end**
5. **copy running-config startup-config**
6. **exit**
7. Reboot the router.

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 export interval** *interval* **Example:** `Router(config)# mpls export interval 360` **Example:**	Configures a periodic time interval for the export of the entire MPLS PAL table to a NetFlow collector. • The *interval* argument specifies the time in minutes between full PAL table exports. The range of valid time intervals is 0 to 10,080 minutes. • We recommend that you select a time interval from 360 minutes (6 hours) to 1440 minutes (24 hours) depending on the size of your network and how often the NetFlow collector might be restarted. • If you enter an interval of 0, full PAL table exports are disabled. PAL information is exported only as labels are allocated. • If you need to restart your NetFlow collector and want to learn PAL information immediately, you can change the *interval* argument. When you change the time interval, the application exports the full PAL table. **Note** Allocated labels are tracked only after you enter the **mplsexportinterval** command. Any labels allocated before you enter this command are not tracked.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	Copies the modified configuration into router NVRAM, permanently saving the settings. The next time the router is reloaded or rebooted the NetFlow MPLS Label Export feature is already part of the configuration.
Step 6	**exit** **Example:** `Router# exit`	Exits to user EXEC mode.
Step 7	Reboot the router.	(Optional) Saves the configuration and reboots the router to ensure that the information collected by this feature is complete.

Displaying Information About the MPLS PAL Table

Perform this task to display information about the MPLS PAL table. The information displayed includes the label, the application that allocated the label, an RD and destination prefix associated with the label, and the time the label was allocated by the application.

SUMMARY STEPS

1. **enable**
2. **show mpls flow mappings**
3. **show ip flow export verbose | include PAL**
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 flow mappings**

Use this command to display entries in the MPLS PAL table. For example:

Example:

```
Router# show mpls flow mappings
Label    Owner    Route-Distinguisher Prefix           Allocated
18       LDP                          10.0.0.5        00:52:10
21       BGP                          0.0.0.0         00:52:18
22       BGP                          0.0.0.0         00:52:18
25       BGP                          0.0.0.0         00:51:44
26       LDP                          10.32.0.0       00:52:10
27       TE-MIDPT                     10.30.0.2       00:52:06
28       LDP                          10.33.0.0       00:52:10
29       LDP                          10.0.0.1        00:52:10
30       LDP                          10.0.0.3        00:52:10
```

In this example, the **mplsexportvpnv4prefixes** command was not configured. Therefore, the MPLS PAL functionality did not export an RD for the BGP application, and the associated prefix is exported as 0.0.0.0.

The following shows sample output from the **showmplsflowmappings**command if you previously entered the **mplsexportvpnv4prefixes** command:

Example:

```
Router# show mpls flow mappings
Label    Owner    Route-Distinguisher Prefix           Allocated
16       LDP                          10.0.0.3        00:58:03
17       LDP                          10.33.0.0       00:58:03
19       TE-MIDPT                     10.30.0.2       00:58:06
20       LDP                          10.0.0.5        00:58:03
23       LDP                          10.0.0.1        00:58:03
24       LDP                          10.32.0.0       00:58:03
27       BGP      100:1               10.34.0.0       00:57:48
31       BGP      100:1               10.0.0.9        00:58:21
32       BGP      100:1               10.3.3.0        00:58:21
```

Step 3 **show ip flow export verbose | include PAL**

Use this command to display the number of MPLS PAL records that were exported to the NetFlow collector. For example:

Example:

```
Router# show ip flow verbose | include PAL
6 MPLS PAL records exported
```

When you specify the **verbose** keyword and MPLS PAL records have been exported using NetFlow Version 9 data format, the command output contains an additional line that precedes the "*x* records exported in *y* UDP datagrams" line.

Step 4 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuring the Export of MPLS VPN Version 4 Label Information from the MPLS PAL Table to a NetFlow Collector

Perform the following task to configure the export of VPNv4 label information from the MPLS PAL table to a NetFlow collector.

This allows you to track VPN prefix information for MPLS labels allocated by VPNs, BGP IPv4, and BGP VPNv4. You can use the data analyzed by the collector to assist in network traffic planning and billing.

Before You Begin

A VRF must be configured on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls export interval** *interval*
4. **mpls export vpnv4 prefixes**
5. **end**
6. **copy running-config startup-config**
7. **exit**
8. Reboot the router.
9. **enable**
10. **show mpls flow mappings**

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 export interval** *interval* **Example:** `Router(config)# mpls export interval 1440`	Configures the collection and export of MPLS PAL information to a NetFlow collector. • The *interval* argument specifies the time in minutes between full PAL table exports. The range of valid time intervals is 0 to 10,080 minutes.

	Command or Action	Purpose
		• We recommend that you select a time interval of 6 hours (360 minutes) to 24 hours (1440 minutes) depending on the size of your network. • If you enter an interval of 0, full PAL table exports are disabled. PAL information is exported only as labels are allocated. • If you need to restart your NetFlow collector and want to learn PAL information immediately, you can change the *interval* argument. When you change the time interval, the application exports the full PAL table.
Step 4	**mpls export vpnv4 prefixes** **Example:** `Router(config)# mpls export vpnv4 prefixes`	Configures the tracking and export of VPNv4 label information from the MPLS PAL table to a NetFlow collector.
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 configuration into router NVRAM, permanently saving the settings. The next time the router is rebooted the tracking and export of VPNv4 label information from the MPLS PAL table to a NetFlow collector is already part of the configuration.
Step 7	**exit** **Example:** `Router# exit`	Exits to user EXEC mode.
Step 8	Reboot the router.	(Optional) Saves the configuration and reboots the router to ensure that the information collected by this feature is complete.
Step 9	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 10	**show mpls flow mappings** **Example:** `Router# show mpls flow mappings`	Displays MPLS PAL table entries that include VPNv4 prefixes and VPN RDs.

Configuration Examples for NetFlow MPLS Label Export

Configuring NetFlow MPLS Prefix Application Label Table Export Examples

The following examples show how to configure NetFlow MPLS PAL table export on a PE router.

This example shows how to configure the export of the full MPLS PAL table every 480 minutes (8 hours):

```
configure terminal
!
mpls export interval 480
end
copy running-config startup-config
exit
```

This example shows how to configure MPLS PAL information export only as the labels are allocated:

```
configure terminal
!
mpls export interval 0
end
copy running-config startup-config
exit
```

In this example, the full MPLS PAL table is not exported repeatedly.

Configuring the Export of MPLS VPNv4 Label Information from the MPLS PAL Table Example

The following example shows how to configure the export of MPLS VPNv4 label information from the MPLS PAL table:

```
configure terminal
!
mpls export interval 720
mpls export vpnv4 prefixes
end
copy running-config startup-config
exit
```

The full MPLS PAL table with MPLS VPNv4 label information is configured to export to the NetFlow collector every 720 minutes (12 hours).

Additional References

The following sections provide references related to the NetFlow MPLS Label Export feature.

Related Documents

Related Topic	Document Title
Tasks for configuring MPLS-aware NetFlow	Configuring MPLS-aware NetFlow
Overview of the NetFlow application and advanced NetFlow features and services	Cisco IOS NetFlow Overview
Tasks for configuring NetFlow to capture and export network traffic data	Configuring NetFlow and NetFlow Data Export
Tasks for configuring MPLS egress NetFlow accounting	Configuring MPLS Egress NetFlow Accounting
Detailed information about the fields available in Version 9 export format and about export format architecture	Cisco IOS NetFlow Version 9 Flow-Record Format

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 3954	Cisco Systems NetFlow Services Export Version 9
RFC 2547	BGP/MPLS VPNs

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

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Multiprotocol Label Switching Command Reference* at http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List* .

- **mpls export interval**
- **mpls export vpnv4 prefixes**
- **show ip flow export**
- **show mpls flow mappings**

Feature Information for NetFlow MPLS Label Export

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 3: Feature Information for NetFlow MPLS Label Export

Feature Name	Releases	Feature Information
NetFlow MPLS Label Export	12.2(28)SB 12.2(33)SRA	The NetFlow MPLS Label Export feature provides the label switch router (LSR) with the capability of collecting and exporting the top label in the MPLS label stack along with its prefix or Forwarding Equivalence Class (FEC) and the application allocating the label to a NetFlow collector for supported MPLS applications. In 12.2(28)SB, this feature was introduced. In 12.2(33)SRA, this feature was integrated into a 12.2SRA release.

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.

export packet --A type of packet built by a device (for example, a router) with NetFlow services enabled. The packet is addressed to another device (for example, the NetFlow Collection Engine). The packet contains NetFlow statistics. The other device processes the packet (parses, aggregates, and stores information on IP flows).

FEC --Forward Equivalency 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. A flow is another example

flow --A unidirectional stream of packets between a given source and destination--each of which is defined by a network-layer IP address and transport-layer source and destination port numbers. A unique flow is defined as the combination of the following key fields: source IP address, destination IP address, source port number, destination port number, Layer 3 protocol type, type of service (ToS), and input logical interface.

flowset --A collection of flow records that follow the packet header in an export packet. A flowset contains information that must be parsed and interpreted by the NetFlow Collection Engine. There are two different types of flowsets: template flowsets and data flowsets. An export packet contains one or more flowsets, and both template and data flowsets can be mixed in the same export packet.

IPv6 --IP Version 6. Replacement for IP Version 4 (IPv4). IPv6 includes support for flow ID in the packet header, which can be used to identify flows. Formerly called IPng (next generation).

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

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).

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.

LSR --label switch router. A router that forwards packets in a Multiprotocol Label Switching (MPLS) network by looking only at the fixed-length label.

MPLS --Multiprotocol Label Switching. A switching method in which IP traffic is forwarded 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.

NetFlow --A Cisco IOS application that provides statistics on packets flowing through the router. It is emerging as a primary network accounting and security technology.

NetFlow Collection Engine (formerly NetFlow FlowCollector)--A Cisco application that is used with NetFlow on Cisco routers and Catalyst series switches. The NetFlow Collection Engine collects packets from the router that is running NetFlow and decodes, aggregates, and stores them. You can generate reports on various aggregations that can be set up on the NetFlow Collection Engine.

NetFlow v9 --NetFlow export format Version 9. A flexible and extensible means of carrying NetFlow records from a network node to a collector. NetFlow Version 9 has definable record types and is self-describing for easier NetFlow Collection Engine configuration.

network byte order --Internet-standard ordering of the bytes corresponding to numeric values.

options data record --Special type of data record that is used in the NetFlow process. It is based on an options template and has a reserved template ID that provides information about the NetFlow process itself.

options template --A type of template record that the router uses to communicate the format of NetFlow-related data to the NetFlow collector.

P router --provider core or backbone router. A router that is part of a service provider's core or backbone network and is connected to the provider edge (PE) routers.

packet header --First part of an export packet. It provides basic information about the packet (such as the NetFlow version, number of records contained in the packet, and sequence numbering) so that lost packets can be detected.

PAL table --Prefix/Application/Label table. A data structure that collects and exports the prefix, application, and time stamp for a specific label.

PE router --provider edge router. A router that is part of a service provider's network connected to a customer edge (CE) router. All Virtual Private Network (VPN) processing occurs in the PE router.

RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 prefix.

There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format or it can be configured in the IP address:network number format (IP-address:nn).

RP --Route Processor. A 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 router. Sometimes called a Supervisory Processor.

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.

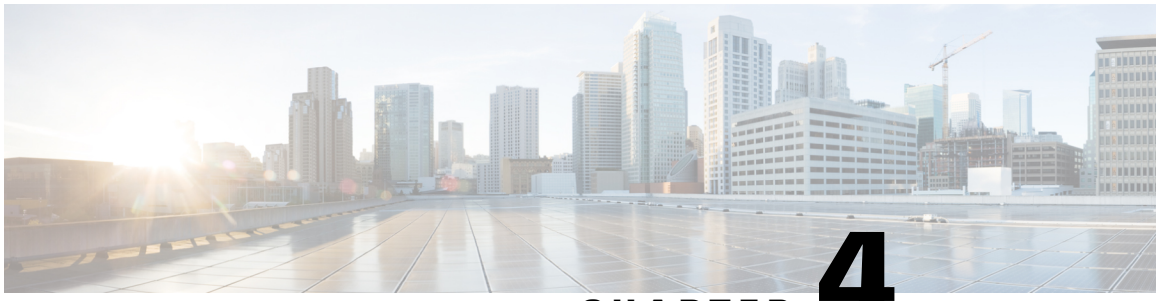
TE tunnel --traffic engineering tunnel. A label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path.

template flowset --A collection of template records that are grouped in an export packet.

template ID --A unique number that distinguishes a template record produced by an export device from other template records produced by the same export device. A NetFlow Collection Engine application can receive export packets from several devices. You should be aware that uniqueness is not guaranteed across export devices. Thus, you should configure the NetFlow Collection Engine to cache the address of the export device that produced the template ID in order to enforce uniqueness.

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.

VPNv4 prefix --IPv4 prefix preceded by an 8-byte route distinguisher. The VPN addresses are made unique by adding a route distinguisher to the front of the address.



CHAPTER 4

ATM PVC Bundle Enhancement MPLS EXP-Based PVC Selection

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This module describes the ATM PVC Bundle Enhancement MPLS EXP-Based PVC Selection feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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 Overview

The ATM PVC Bundle Enhancement MPLS EXP-Based PVC Selection feature is an extension to the IP to ATM Class of Service feature suite. The IP to ATM Class of Service feature suite, using virtual circuit (VC) support and bundle management, maps quality of service (QoS) characteristics between IP and ATM. It provides customers who have multiple VCs (with varying qualities of service to the same destination) the ability to build a QoS differentiated network.

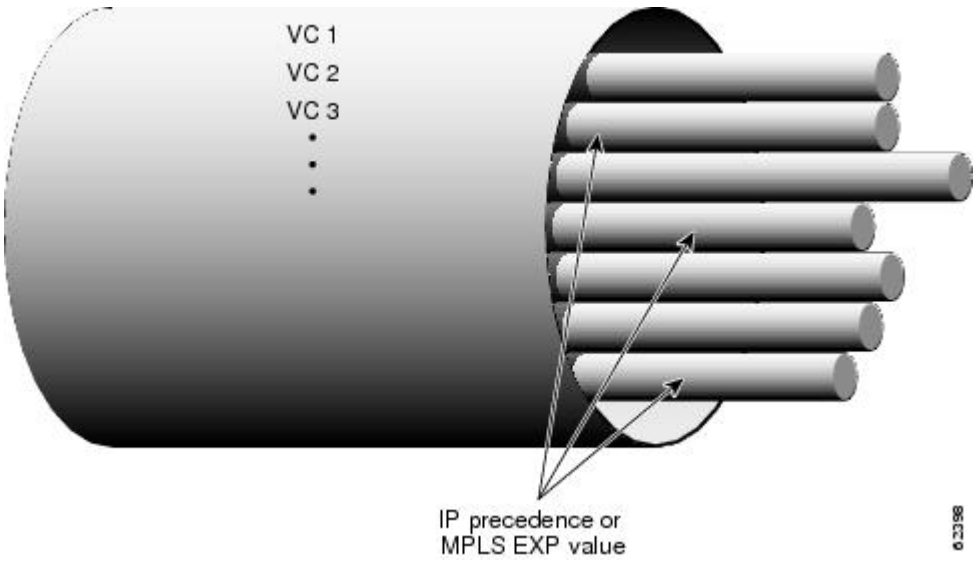
The IP to ATM Class of Service feature suite allowed customers to use IP precedence level as the selection criteria for packet forwarding. This new feature now gives customers the option of using the Multiprotocol Label Switching (MPLS) experimental (EXP) level as an additional selection criteria for packet forwarding.

Note If a selection criteria for packet forwarding is not selected (that is, if the packet is unlabeled), this new feature uses the IP precedence level as the default selection criteria.

VC Bundle Support and Bundle Management

ATM VC bundle management allows you to configure multiple VCs that have different QoS characteristics between any pair of ATM-connected routers. As shown in the figure below, these VCs are grouped in a bundle and are referred to as bundle members.

Figure 2: ATM VC Bundle



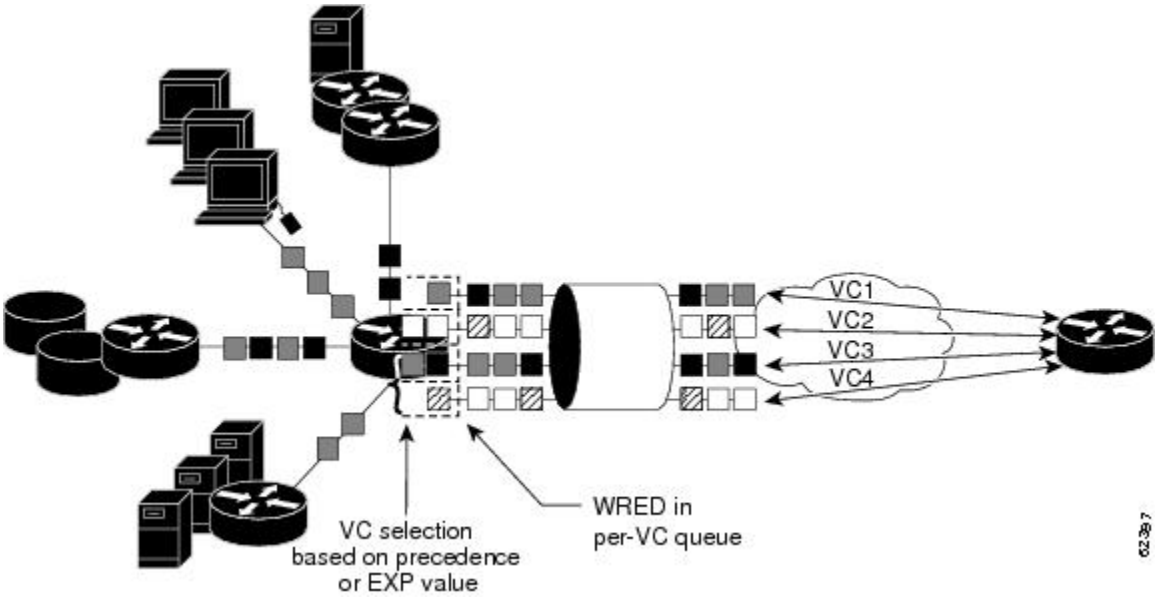
ATM VC bundle management allows you to define an ATM VC bundle and add VCs to it. Each VC of a bundle has its own ATM traffic class and ATM traffic parameters. You can apply attributes and characteristics to discrete VC bundle members, or you can apply them collectively at the bundle level.

Using VC bundles, you can create differentiated service by flexibly distributing MPLS EXP levels over the different VC bundle members. You can map a single MPLS EXP level, or a range of these levels, to each

discrete VC in the bundle, thereby enabling individual VCs in the bundle to carry packets marked with different MPLS EXP levels. You can use Weighted Random Early Detection (WRED) or distributed WRED (dWRED) to further differentiate service across traffic that has different MPLS EXP levels.

To determine which VC in the bundle to use to forward a packet to its destination, the ATM VC bundle management software matches MPLS EXP levels between packets and VCs (see the figure below). IP traffic is sent to the next hop address for the bundle because all VCs in a bundle share the same destination, but the VC used to carry a packet depends on the value set for that packet in the MPLS EXP level of the type of service (ToS) byte of its header. The ATM VC bundle management software matches the MPLS EXP level of the packet to the MPLS EXP levels assigned to a VC, sending the packet out on the appropriate VC. Moreover, the ATM VC bundle management software allows you to configure how traffic will be redirected when the VC to which the packet was initially directed goes down. The figure below illustrates how the ATM VC bundle management software determines which permanent virtual circuit (PVC) bundle member to use to carry a packet and how WRED (or dWRED) is used to differentiate traffic on the same VC.

Figure 3: ATM VC Bundle PVC Selection for Packet Transfer



The support of multiple parallel ATM VCs allows you to create stronger service differentiation at the IP layer. For instance, you might want to configure the network to provide IP traffic belonging to real-time class of service (CoS) (such as Voice over IP traffic) on an ATM VC with strict constraints (constant bit rate (CBR) or variable bit rate real-time (VBR-rt), for example), while also allowing the network to transport nonreal-time traffic over a more elastic ATM unspecified bit rate (UBR) PVC. UBR is effectively the ATM version of best-effort service. Using a configuration such as this would allow you to make full use of your network capacity.

Benefits

Improved System Performance

This feature is designed to provide a true working solution to class-based services, without the investment of new ATM network infrastructures. Now networks can offer different service classes (sometimes termed

differential service classes) across the entire WAN, not just the routed portion. Mission-critical applications can be given exceptional service during periods of high network usage and congestion. In addition, noncritical traffic can be restricted in its network usage, ensuring greater QoS for more important traffic and user types.

Additional Selection Criteria

This new feature now gives customers the option of using the MPLS EXP level, in addition to IP precedence, as a selection criteria for packet forwarding.

Restrictions

- This feature requires ATM PVC management, as well as Forwarding Information Base (FIB) and Tag Forwarding Information Base (TFIB) switching functionality.
- This feature is not supported on either the ATM interface processor (AIP) or the ATM Lite port adapter (PA-A1).
- The router at the remote end of the network must be using a version of Cisco IOS that supports MPLS and ATM PVC management.

Related Features and Technologies

This feature is similar to the IP to ATM Class of Service feature suite, which is documented in the "Configuring IP to ATM Class of Service" module.

Related Documents

- Cisco IOS Quality of Service Solutions Command Reference
- Cisco IOS Switching Services Command Reference
- "Frame Relay PVC Bundles with QoS Support for IP and MPLS " module
- Cisco IOS Wide-Area Networking Command Reference
- "IP to ATM SVC Bundles for Class of Service (CoS) Mapping " module
- "MPLS Label Distribution Protocol (LDP) Overview " module

Supported Platforms

- Cisco 3600 series

The ATM Adapter PA-A3 is not supported on either the Cisco 3620 router or the Cisco 3640 router. Because certain QoS features (for example, WRED) require the ATM Adapter PA-A3, specific limitations may apply. For more information about platform and feature support, refer to Cisco Feature Navigator (described below).

- Cisco 3725
- Cisco 3745

- Cisco 7200 series
- Cisco 7500 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register .

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards MIBs and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS 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

No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the ATM PVC Bundle Enhancement MPLS EXP-Based PVC Selection feature. Each task in the list is identified as either required or optional.

Enabling MPLS

SUMMARY STEPS

1. Router(config)# **ip cef**
2. Router(config)# **mpls label protocol ldp**
3. Router(config)# **interface** *type number* [*name-tag*]
4. Router(config-if)# **mpls ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **ip cef**	Enables Cisco Express Forwarding (CEF) on the Route Processor (RP) card. An optional keyword **distributed** can be used with this command to enable distributed CEF (dCEF) for the Versatile Interface Processor (VIP)-based platforms.
Step 2	Router(config)# **mpls label protocol ldp**	Specifies the default label distribution protocol for a platform.
Step 3	Router(config)# **interface** *type number* [*name-tag*]	Configures an interface type and enters interface configuration mode.
Step 4	Router(config-if)# **mpls ip**	Enables MPLS forwarding of IPv4 packets along normally routed paths for the platform.

Creating a VC Bundle

Command	Purpose
`Device(config-if)#` **bundle** *`bundle-name`*	Creates the specified bundle and enters bundle configuration mode.

Applying Parameters to Bundles

Configuring Bundle-Level Parameters

Command	Purpose
`Router(config-if-atm-bundle)#` **protocol** *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**]	Configures a static map or enables Inverse Address Resolution Protocol (Inverse ARP) or Inverse ARP broadcasts for the bundle. **Note** Note that some parameters applied through a VC class or directly to the bundle can be superseded by commands that you directly apply to individual VCs in bundle-vc configuration mode.
`Router(config-if-atm-bundle)#` **encapsulation** *aal-encap*	Configures the ATM adaptation layer (AAL) and encapsulation type for the bundle.

Configuring a VC Bundle Member Directly

Command	Purpose
`Router(config-if-atm-member)#` **ubr** *output-pcr* [*input-pcr*]	Configures the VC for UBR QoS and specifies the output peak cell rate (PCR) for it.
`Router(config-if-atm-member)#` **vbr-nrt** *output-pcr output-scr output-mbs* [*input-pcr*] [*input-scr*] [*input-mbs*]	Configures the VC for variable bit rate nonreal-time (VBR-nrt) QoS and specifies the output PCR, output sustainable cell rate, and output maximum burst cell size for it.
`Router(config-if-atm-member)#` **mpls experimental** [**other** | *range*]	Configures the MPLS EXP levels for the VC.
`Router(config-if-atm-member)#` **bump** {**implicit** | **explicit** *precedence-level* | **traffic**}	Configures the bumping rules for the VC.
`Router(config-if-atm-member)#` **protect** {**group** | **vc**}	Configures the VC to belong to the protected group of the bundle or to be an individually protected VC bundle member.

Configuring VC Class Parameters to Apply to a Bundle

Command	Purpose
`Router(config-vc-class)#` **oam-bundle** [**manage**] [*frequency*]	Enables end-to-end F5 Operation, Administration, and Maintenance (OAM) loopback cell generation and OAM management for all VCs in the bundle. **Note** If you are using a VC class to configure the bundle, you must attach the VC class to the bundle. To do this, complete the procedure in the section "Attaching a Class to a Bundle, on page 70."

Attaching a Class to a Bundle

Command	Purpose
`Router(config-if-atm-bundle)#` **class-bundle** *vc-class-name*	Configures a bundle with the bundle-level commands contained in the specified VC class.

Verifying the Configuration

Command	Purpose
`Router#` **debug atm bundle error**	Displays debug messages for PVC bundle errors.
`Router#` **debug atm bundle events**	Displays PVC bundle events.
`Router#` **show atm map**	Displays the list of all configured ATM static maps to remote hosts on an ATM network.
`Router#` **show atm bundle** *bundle-name*	Displays the bundle attributes assigned to each VC member and the current working status of the VC members.
`Router#` **show mpls forwarding-table**	Displays the contents of the MPLS FIB.

Configuration Examples

Example VC Bundle Configuration Using a VC Class

This example configures VC bundle management on a router that uses Intermediate System-to-Intermediate System (IS-IS) as its IP routing protocol.

Bundle-Class Class

At the outset, this configuration defines a VC class called "bundle-class," which includes commands that set VC parameters. When the class bundle-class is applied at the bundle level, these parameters are applied to all VCs that belong to the bundle. Note that any commands applied directly to an individual VC of a bundle in bundle-vc mode take precedence over commands applied globally at the bundle level. Taking into account hierarchy precedence rules, VCs belonging to any bundle to which the class bundle-class is applied will be characterized by the following parameters: aal5snap encapsulation, broadcast on, use of Inverse ARP to resolve IP addresses, and OAM enabled.

```
router isis
 net 49.0000.0000.0000.1111.00
vc-class atm bundle-class
 encapsulation aal5snap
 broadcast
 protocol ip inarp
 oam-bundle manage 3
 oam 4 3 10
```

The following four sections of the configuration define specific VC classes. Each of these classes contains commands used to specify parameters that can then be applied to individual VCs in a bundle by assigning the class to that VC.

Control-Class Class

When the class called "control-class" is applied to a VC, the VC carries traffic whose MPLS EXP level is 7. When the VC to which this class is assigned goes down, it takes the bundle down with it because this class makes the VC a protected one. The QoS type of a VC using this class is vbr-nrt.

```
vc-class atm control-class
 mpls experimental 7
 protect vc
 vbr-nrt 1000 5000 32
```

Premium-Class Class

When the class called "premium-class" is applied to a VC, the VC carries traffic whose MPLS EXP levels are 6 and 5. The VC does not allow other traffic to be bumped onto it. When the VC to which this class is applied goes down, its bumped traffic will be redirected to a VC whose MPLS EXP level is 7. This class makes a VC a member of the protected group of the bundle. When all members of a protected group go down, the bundle goes down. The QoS type of a VC using this class is vbr-nrt.

```
vc-class atm premium-class
 mpls experimental 6-5
```

```
 no bump traffic
 protect group
 bump explicitly 7
 vbr-nrt 20000 10000 32
```

Priority-Class Class

When the class called "priority-class" is applied to a VC, the VC is configured to carry traffic with an MPLS EXP level in the 4 - 2 range. The VC uses the implicit bumping rule, it allows traffic to be bumped, and it belongs to the protected group of the bundle. The QoS type of a VC using this class is ubr+.

```
vc-class atm priority-class
 mpls experimental 4-2
 protect group
 ubr+ 10000 3000
```

Basic-Class Class

When the class called "basic-class" is applied to a VC, the VC is configured through the **mpls experimental other** command to carry traffic with MPLS EXP levels not specified in the profile. The VC using this class belongs to the protected group of the bundle. The QoS type of a VC using this class is ubr.

```
vc-class atm basic-class
 mpls experimental other
 protect group
 ubr 10000
```

The following sets of commands configure three bundles that the router subinterface uses to connect to three of its neighbors. These bundles are called "new-york," "san-francisco," and "los-angeles." Bundle new-york has four VC members, bundle san-francisco has four VC members, and bundle los-angeles has three VC members.

new-york Bundle

The first part of this example specifies the IP address of the subinterface, the router protocol--the router uses IS-IS as an IP routing protocol--and it creates the first bundle called "new-york" and enters bundle configuration mode:

```
interface a1/0.1 multipoint
 ip address 10.0.0.1 255.255.255.0
 ip router isis
 bundle new-york
```

From within bundle configuration mode, the next portion of the configuration uses two protocol commands to enable IP and Open Systems Interconnect (OSI) traffic flows in the bundle. The OSI routing packets will use the highest MPLS EXP VC in the bundle. The OSI data packets, if any, will use the lowest MPLS EXP VC in the bundle. If configured, other protocols, such as Internet Packet Exchange (IPX) or AppleTalk, will always use the lowest MPLS EXP VC in the bundle.

As the indentation levels of the preceding and following commands suggest, subordinate to bundle new-york is a command that configures its protocol and a command that applies the class called "bundle-class" to it.

```
  protocol ip 1.1.1.2 broadcast
  protocol clns 49.0000.0000.2222.00 broadcast
  class-bundle bundle-class
```

The class called "bundle-class," which is applied to the bundle new-york, includes a **protocol ip inarp** command. According to inheritance rules, **protocol ip**, configured at the bundle level, takes precedence over **protocol ip inarp** specified in the class bundle-class.

The next set of commands beginning with **pvc-bundle ny-control 207**, which are further subordinate, add four VCs (called "ny-control," "ny-premium," "ny-priority," and "ny-basic") to the bundle new-york. A particular class--that is, one of the classes predefined in this configuration example--is applied to each VC to configure it with parameters specified by commands included in the class.

As is the case for this configuration, to configure individual VCs belonging to a bundle, the router must be in bundle mode for the mother bundle. For each VC belonging to the bundle, the subordinate mode is pvc-mode for the specific VC.

The following commands configure the individual VCs for the bundle new-york:

```
pvc-bundle ny-control 207
 class-vc control-class
pvc-bundle ny-premium 206
 class-vc premium-class
pvc-bundle ny-priority 204
 class-vc priority-class
pvc-bundle ny-basic 201
 class-vc basic-class
```

san-francisco Bundle

The following set of commands create and configure a bundle called "san-francisco." At the bundle configuration level, the configuration commands included in the class bundle-class are ascribed to the bundle san-francisco and to the individual VCs that belong to the bundle. Then, the **pvc-bundle** command is executed for each individual VC to add it to the bundle. After a VC is added and bundle-vc configuration mode is entered, a particular, preconfigured class is assigned to the VC. The configuration commands comprising that class are used to configure the VC. Rules of hierarchy apply at this point. Command parameters contained in the applied class are superseded by the same parameters applied at the bundle configuration level, which are superseded by the same parameters applied directly to a VC.

```
bundle san-francisco
 protocol clns 49.0000.0000.0000.333.00 broadcast
 inarp 1
 class-bundle bundle-class
 pvc-bundle sf-control 307
  class-vc control-class
 pvc-bundle sf-premium 306
  class-vc premium-class
 pvc-bundle sf-priority 304
  class-vc priority-class
 pvc-bundle sf-basic 301
  class-vc basic-class
```

los-angeles Bundle

The following set of commands create and configure a bundle called "los-angeles." At the bundle configuration level, the configuration commands included in the class bundle-class are ascribed to the bundle los-angeles and to the individual VCs that belong to the bundle. Then, the **pvc-bundle** command is executed for each individual VC to add it to the bundle. After a VC is added and bundle-vc configuration mode is entered, the MPLS EXP level is set for the VC, and the VC is either configured as a member of a protected group (protect group) or as an individually protected VC. A particular class is then assigned to each VC to further characterize it. Rules of hierarchy apply. Parameters of commands applied directly and discretely to a VC take precedence

over the same parameters applied within a class to the VC at the bundle-vc configuration level, which take precedence over the same parameters applied to the entire bundle at the bundle configuration level.

```
bundle los-angeles
 protocol ip 1.1.1.4 broadcast
 protocol clns 49.0000.0000.4444.00 broadcast
 inarp 1
 class-bundle bundle-class
 pvc-bundle la-high 407
  mpls experimental 7-5
  protect vc
  class-vc premium-class
 pvc-bundle la-mid 404
  mpls experimental 4-2
  protect group
  class-vc priority-class
 pvc-bundle la-low 401
  mpls experimental other
  protect group
  class-vc basic-class
```

Command Reference

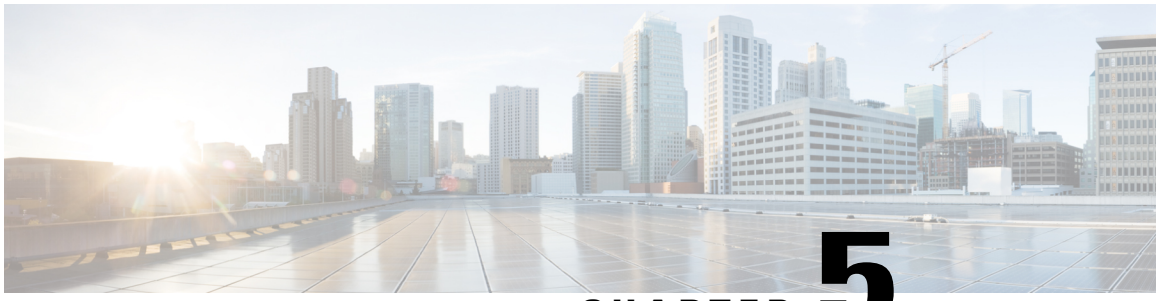
The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* at http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html . For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

New Commands

- **mpls experimental**

Modified Commands

- **show mpls forwarding-table**



CHAPTER 5

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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.

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. **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	**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

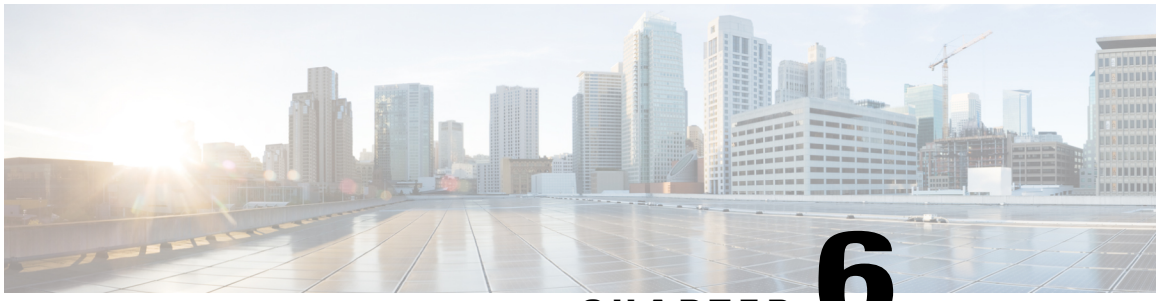
Feature Information for 6PE Multipath

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 4: Feature Information for 6PE Multipath

Feature Name	Releases	Feature Information
6PE Multipath	12.2(25)S 12.2(28)SB 12.2(33)SRA 12.4(6)T	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 6

IPv6 Switching: Provider Edge Device 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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.

Prerequisites for IPv6 Switching: Provider Edge Device over MPLS

Before the IPv6 Provider Edge Device 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 Device 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.

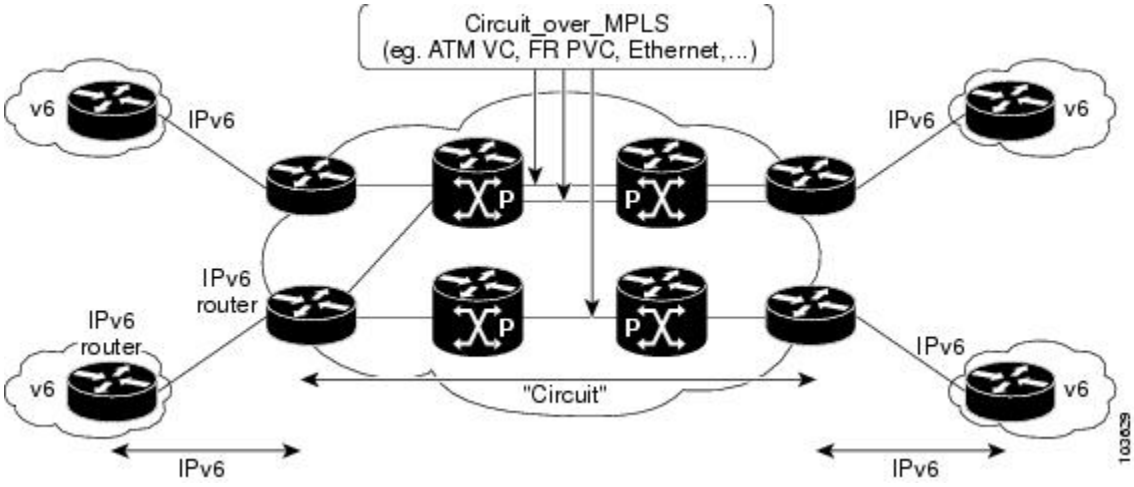
Additionally, the inherent Virtual Private Network (VPN) and MPLS traffic engineering (MPLS-TE) services available within an MPLS environment allow IPv6 networks to be combined into IPv4 VPNs or extranets over an infrastructure supporting IPv4 VPNs and MPLS-TE.

IPv6 over a Circuit Transport over MPLS

Using any circuit transport for deploying IPv6 over MPLS networks has no impact on the operation or infrastructure of MPLS, and requires no configuration changes to the core or provider edge devices. Communication between the remote IPv6 domains runs native IPv6 protocols over a dedicated link, where the underlying mechanisms are fully transparent to IPv6. The IPv6 traffic is tunneled using the Any Transport over MPLS (MPLS/AToM) or Ethernet over MPLS (EoMPLS) feature with the devices connected through an ATM OC-3 or Ethernet interface, respectively.

The figure below shows the configuration for IPv6 over any circuit transport over MPLS.

Figure 4: IPv6 over a Circuit Transport over MPLS

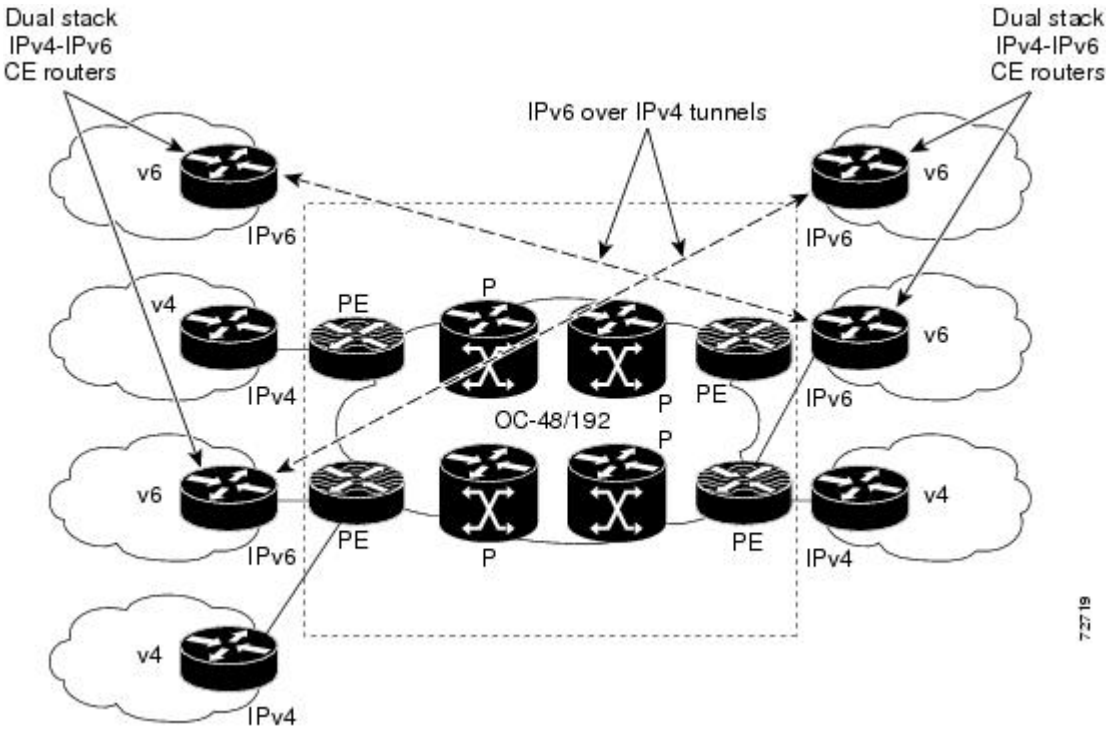


IPv6 Using Tunnels on the Customer Edge Devices

Using tunnels on the customer edge (CE) devices is the simplest way of deploying IPv6 over MPLS networks with no impact on the operation or infrastructure of MPLS, and no configuration changes to the core or provider edge devices. Communication between the remote IPv6 domains uses standard tunneling mechanisms and

requires the CE devices to be configured to run dual IPv4 and IPv6 protocol stacks. The figure below shows the configuration using tunnels on the CE devices.

Figure 5: IPv6 Using Tunnels on the CE Devices



Refer to Implementing Tunneling for IPv6 for configuration information on manually configured tunnels, automatic tunnels, and 6to4 tunnels.

Limitations on using tunnels involve the manual configuring of a mesh of tunnels on the CE devices, creating scaling issues for large networks.

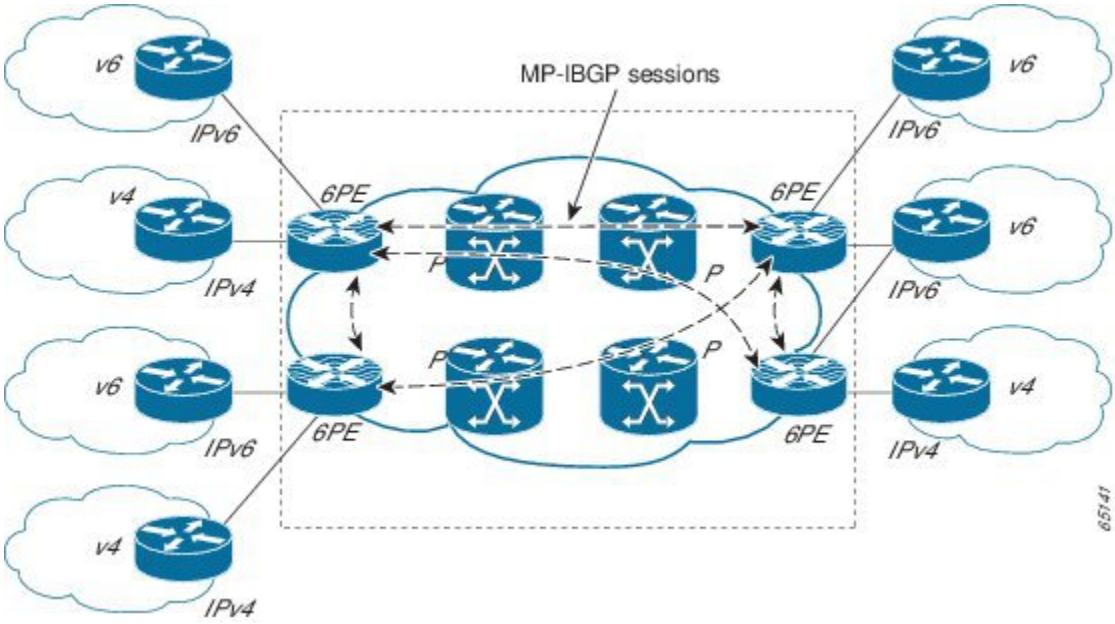
IPv6 on the Provider Edge Devices

The Cisco implementation of IPv6 provider edge device 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 Device over MPLS

Deploying IPv6 over a Circuit Transport over MPLS

To deploy IPv6 over a circuit transport over MPLS, the IPv6 devices must be configured for IPv6 connectivity. The MPLS device configuration requires AToM configuration or EoMPLS configuration.

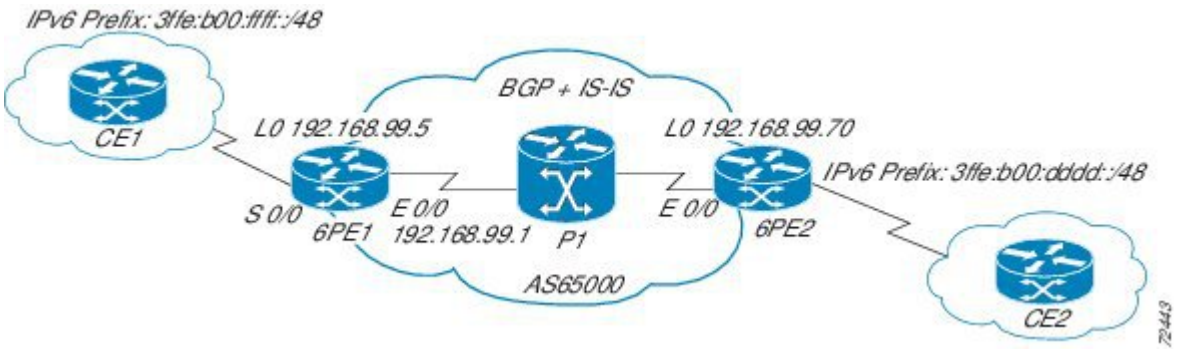
Deploying IPv6 on the Provider Edge Devices (6PE)

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 Serial 0/0`	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.

	Command or Action	Purpose
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:** `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 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. **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	**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 Device over MPLS

Example: Customer Edge Device

This example shows that the serial interface 0/0 of the customer edge device--CE1 in the figure above--is connected to the service provider and is assigned an IPv6 address. IPv6 is enabled and a default static route is installed using the IPv6 address of serial interface 0/0 of the 6PE1 device.

```
ip cef
!
ipv6 unicast-routing
!
interface Serial 0/0
 description to_6PE1_router
 no ip address
 ipv6 address 2001:DB8:FFFF::2/64
!
ipv6 route ::/0 Serial 0/0 FE80::210:XXXX:FEE1:1001
```

Example: Provider Edge Device

The 6PE device--Device 6PE1 in the figure above--is configured for both IPv4 and IPv6 traffic. Ethernet interface 0/0 is configured with an IPv4 address and is connected to a device in the core of the network--device P1 in the figure above. 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--Device 6PE2 in the figure above--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. If IPv6 packets are generated in the local device, the IPv6 address for MPLS processing will be the address of loopback interface 0.

This example shows that the serial interface 0/0 connects to the customer and the IPv6 prefix delegated to the customer is 2001:DB8:ffff::/48, which is determined from the service provider IPv6 prefix. A static route is configured to route IPv6 packets between the 6PE route and the CE device.

```
ip cef
ipv6 cef
ipv6 unicast-routing
!
mpls ipv6 source-interface Loopback0
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 192.168.99.5 255.255.255.255
 ipv6 address 2001:DB8:1000:1::1/64
!
interface Ethernet0/0
 description to_P_router
 ip address 192.168.99.1 255.255.255.252
 ip router isis
 tag-switching ip
!
interface Serial0/0
 description to_CE_router
 no ip address
 ipv6 address 2001:DB8:FFFF::1/64
!
router isis
 passive-interface Loopback0
 net 49.0001.1921.6809.9005.00
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.99.70 remote-as 65000
 neighbor 192.168.99.70 description to_6PE2
 neighbor 192.168.99.70 update-source Loopback0
 !
 address-family ipv6
 neighbor 192.168.99.70 activate
 neighbor 192.168.99.70 send-label
 network 2001:DB8:FFFF::/48
 exit-address-family
!
ipv6 route 2001:DB8:FFFF::/48 Ethernet0/0 2001:DB8:FFFF::2
```

Example: Core Device

This example shows that the device in the core of the network--Device P in the figure above--is running MPLS, IS-IS, and IPv4 only. The 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. TDP and tag switching are enabled on both the 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 Ethernet0/0
 description to_6PE1
 ip address 192.168.99.2 255.255.255.252
 ip router isis
 tag-switching ip
!
interface Ethernet0/1
 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

This example shows output information about an IPv6 route 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
```

This example shows output information about a BGP peer, including the IPv6 label capability, 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
```

This example shows output information linking the MPLS label with prefixes 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  1.1.1.1/32       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
```

This example shows output information about the top of the stack label with label switching information 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
```

This example shows output information about labels from the Cisco Express Forwarding table 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}
```

This example shows output information from the IPv6 routing table 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.

The 6PE2 device has advertised the IPv6 prefix of 2001:DB8:dddd::/48 configured for the CE2 device 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 device.

```
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 ::, Ethernet0/0
C 2001:DB8:FFFF::/64 [0/0]
  via ::, Ethernet0/0
S 2001:DB8:FFFF::/48 [1/0]
  via 2001:DB8:B00:FFFF::2, Ethernet0/0
```

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

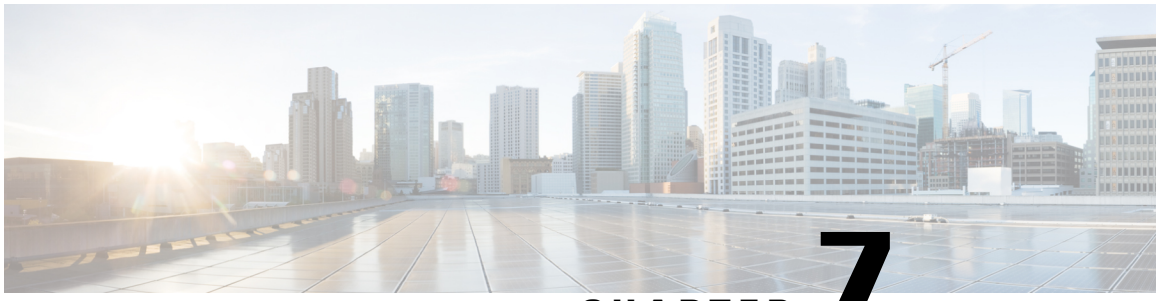
Feature Information for IPv6 Switching: Provider Edge Device over MPLS

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 5: Feature Information for IPv6 Switching: Provider Edge Device over MPLS

Feature Name	Releases	Feature Information
IPv6 Switching: Provider Edge Router over MPLS	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(15)T	The Cisco implementation of IPv6 provider edge device over MPLS enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS LSPs. The following commands were introduced or modified: **address-family ipv6**, **ipv6 address**, **ipv6 cef**, **ipv6 unicast-routing**, **maximum-paths ibgp**, **neighbor activate**, **neighbor remote-as**, **neighbor send-label**, **neighbor update-source**, **no bgp default ipv4-unicast**, **router bgp**, **show bgp ipv6**, **show bgp ipv6 labels**, **show bgp ipv6 neighbors**, **show ipv6 cef**, **show ipv6 route**, **show mpls forwarding-table**.



CHAPTER 7

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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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.

Prerequisites for MPLS Multilink PPP Support

- Cisco Express Forwarding or distributed 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

Restrictions for MPLS Multilink PPP Support

The MPLS Multilink PPP Support feature is limited by platform-specific restrictions that apply to the use of Multilink PPP (MLP) and distributed MLP (dMLP).

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 6: 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

MPLS L3 VPN Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
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 7: 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
Set MPLS EXP bits using the modular QoS Command-Line Interface (MQC)	Supported	Supported	Supported
Matching on MPLS EXP using MQC	Supported	Supported	Supported

MPLS QoS Feature	CE-to-PE Links	PE-to-P Links	CSC CE-to-PE Links
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.

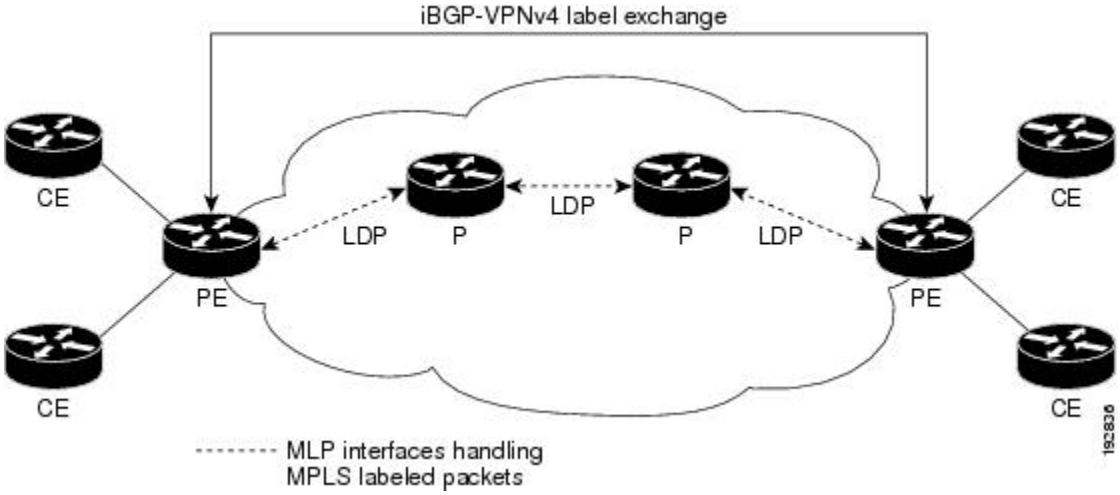
The PE-to-CE routing protocols that are supported for the MPLS Multilink PPP Support feature are external Border Gateway Protocol (eBGP), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP). Static routes are also supported between the CE and PE devices.

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), compressed Real-Time Transport Protocol (cRTP), 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 8: MLP on PE-to-P and P-to-P Links



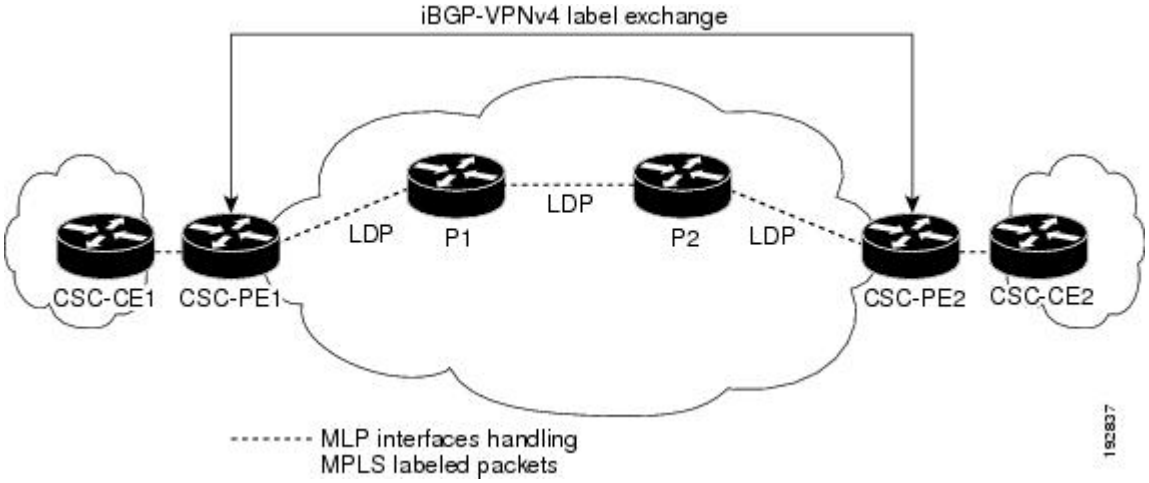
You employ MLP in the PE-to-P or P-to-P links primarily so that you can reduce the number of Interior Gateway Protocol (IGP) adjacencies and facilitate the load sharing of traffic.

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

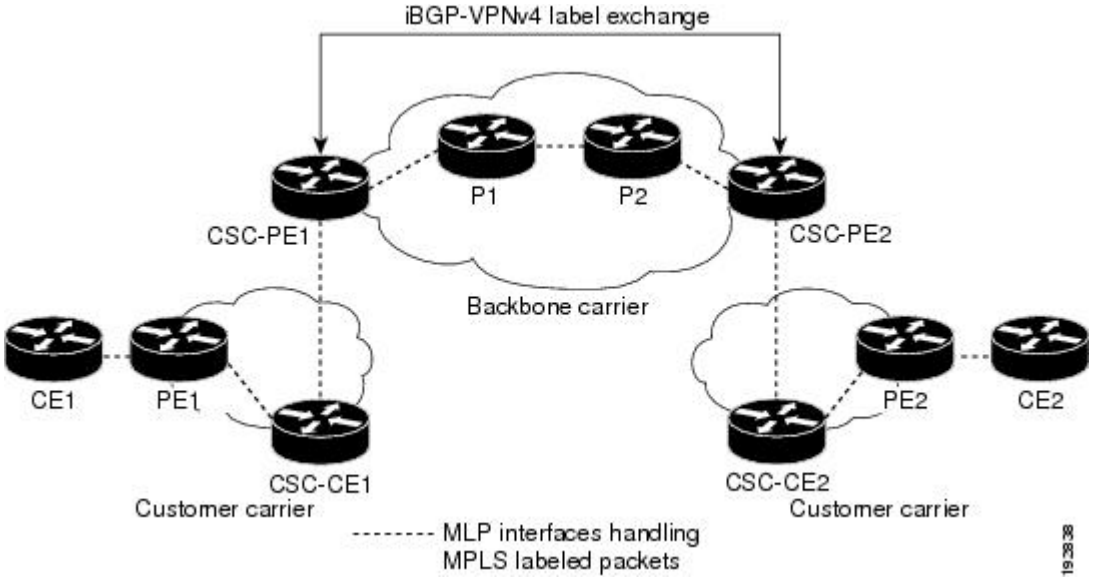
The figure below shows a typical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) network where Multilink PPP (MLP) is configured on the CSC customer edge (CE)-to-provider edge (PE) links.

Figure 9: 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.

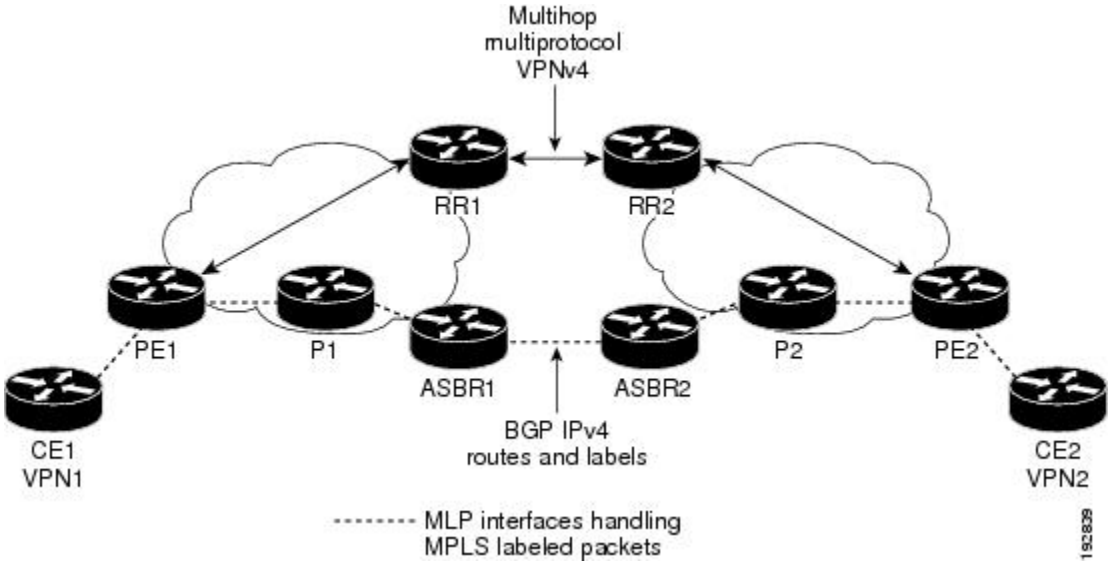
Figure 10: MLP Supported Links with MPLS VPN Carrier Supporting Carrier



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 11: 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 or Distributed Cisco Express Forwarding

Perform the following task to enable Cisco Express Forwarding or distributed Cisco Express Forwarding.

Before You Begin

Multilink PPP (MLP) requires the configuration of Cisco Express Forwarding. Distributed MLP (dMLP) requires the configuration of distributed Cisco Express Forwarding.

Cisco Express Forwarding is enabled by default on most Cisco platforms running Cisco software. To find out if Cisco Express Forwarding is enabled on your platform, enter the **show ip cef** command. If Cisco Express Forwarding is enabled, you receive output that looks like this:

```
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
```

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

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

Distributed Cisco Express Forwarding is enabled by default on devices such as the Catalyst 6500 series switch, the Cisco 7500 series router, and the Cisco 12000 series Internet router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
   - **ip cef**
   - **ip cef distributed**
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	Enter one of the following commands: • **ip cef** • **ip cef distributed** **Example:** `Device(config)# ip cef`	Enables Cisco Express Forwarding switching. or Enables distributed Cisco Express Forwarding switching.

	Command or Action	Purpose
	Example: `Device(config)# ip cef distributed`	
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.

	Command or Action	Purpose
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* / *port* **:** *channel-group*
7. **ip route-cache** [**cef** | **distributed**]
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*	Defines the time slots that belong to each T1 or E1 circuit.

	Command or Action	Purpose
	Example: `Device(config-controller)# channel-group 1 timeslots 1`	• 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. 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* / *port* **:** *channel-group* **Example:** `Device(config)# interface serial 1/0:1`	Configures a serial interface for a Cisco 7500 series router with channelized T1 or E1 and enters interface configuration mode. • The *slot* argument indicates the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument indicates the port number. Refer to the appropriate hardware manual for slot and port information. • The *:channel-group* argument indicates the channel group number. Cisco 7500 series routers specify the channel group number in the range of 0 to 4 defined with the **channel-group** controller configuration command.
Step 7	**ip route-cache** [**cef** | **distributed**] **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. • The **distributed** keyword enables distributed switching on the interface.
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.

	Command or Action	Purpose
		• 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 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.

	Command or Action	Purpose
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:

```
Device# show ip interface brief

Locolrface                  IP-Address      OK? Method Status                Prot
FastEthernet1/0/0         10.3.62.106      YES NVRAM  up                    up
FastEthernet0/0/1          unassigned      YES NVRAM  administratively down down
FastEthernet0/0/0          unassigned      YES NVRAM  administratively down down
FastEthernet0/0/1          unassigned      YES NVRAM  administratively down down
FastEthernet0/0/2          unassigned      YES NVRAM  administratively down down
FastEthernet0/1/0          unassigned      YES NVRAM  administratively down down
FastEthernet0/1/1          unassigned      YES NVRAM  administratively down down
FastEthernet0/1/2          unassigned      YES NVRAM  administratively down down
FastEthernet1/2/0          unassigned      YES NVRAM  administratively down down
```

```
FastEthernet1/0/1          unassigned      YES NVRAM  administratively down down
FastEthernet1/1/0          unassigned      YES NVRAM  administratively down down
FastEthernet1/1/1          unassigned      YES NVRAM  administratively down down
FastEthernet1/1/2          unassigned      YES NVRAM  administratively down down
Serial1/1/0:1              unassigned      YES NVRAM  administratively down down
Serial1/1/0:2              unassigned      YES NVRAM  administratively down down
Serial1/1/1:1              unassigned      YES NVRAM  up                    up
Serial1/1/1:2              unassigned      YES NVRAM  up                    down
Serial1/1/3:1              unassigned      YES NVRAM  up                    up
Serial1/1/3:2              unassigned      YES NVRAM  up                    up
Multilink6                 10.30.0.2       YES NVRAM  up                    up
Multilink8                 unassigned      YES NVRAM  administratively down down
Multilink10                10.34.0.2       YES NVRAM  up                    up
Loopback0                  10.0.0.1        YES NVRAM  up                    up
```

Step 3 **show ppp multilink**

Verifies that you have created a multilink bundle.

Example:

```
Device# show ppp multilink

Multilink1, bundle name is group 1
 Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links: 4 active, 0 inactive (max no set, min not set)
 Serial1/0/0/:1
 Serial1/0/0/:2
 Serial1/0/0/:3
 Serial1/0/0/:4
```

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

Displays information about a specific MLP interface.

Example:

```
Device# show ppp multilink interface multilink6

Multilink6, bundle name is router
  Bundle up for 00:42:46, 1/255 load
  Receive buffer limit 24384 bytes, frag timeout 1524 ms
  Bundle is Distributed
    0/0 fragments/bytes in reassembly list
    1 lost fragments, 48 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x4D7 received sequence, 0x0 sent sequence
  Member links: 2 active, 0 inactive (max not set, min not set)
    Se1/1/3:1, since 00:42:46, 240 weight, 232 frag size
    Se1/1/3:2, since 00:42:46, 240 weight, 232 frag size
```

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

Displays information about serial interfaces in your configuration.

Example:

```
Device# show interface serial 1/1/3:1

Serial1/1/3:1 is up, line protocol is up
  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# show interface serial 1/1/3:2

Serial1/1/3:2 is up, line protocol is up
  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

Sample MPLS Multilink PPP Support Configurations

The following examples show sample configurations for Multilink PPP (MLP) on a Cisco 7200 router, on a Cisco 7500 router, and on a Carrier Supporting Carrier (CSC) network. The configuration of MLP on an interface is the same for provider edge-to-customer edge (PE-to-CE) links, PE-to-provider (P) links, and P-to-P links.

Example: Sample Multilink PPP Configuration on Cisco 7200 Series Router

The following sample configuration is for a Cisco 7200 router, which is connected with a T1 line card and configured with an MPLS Multilink PPP interface:

```
controller T1 1/3
 framing esf
 clock source internal
 linecode b8zs
 channel-group 1 timeslots 1
 channel-group 2 timeslots 2
 no yellow generation
 no yellow detection
!
interface Multilink6
 ip address 10.37.0.1 255.0.0.0
 ppp multilink interleave
 tag-switching ip
 load-interval 30
 multilink-group 6
!
interface Serial1/3:1
 encapsulation ppp
 no ip address
 ppp multilink
 tx-queue-limit 26
 multilink-group 6
 peer neighbor-route
!
interface Serial1/3:2
 encapsulation ppp
 no ip address
 ppp multilink
 tx-queue-limit 26
 multilink-group 6
 peer neighbor-route
```

Example: Sample Multilink PPP Configuration for Cisco 7500 Series Router

The following sample configuration is for a Cisco 7500 router, which is connected with a T1 line card and configured with an MPLS Multilink PPP interface:

```
controller T1 1/1/3
 framing esf
 clock source internal
 linecode b8zs
 channel-group 1 timeslots 1
 channel-group 2 timeslots 2
 no yellow generation
```

```
 no yellow detection
!
interface Multilink6
 ip address 10.37.0.2 255.0.0.0
 ppp multilink interleave
 tag-switching ip
 load-interval 30
 multilink-group 6
!
interface Serial1/1/3:1
 encapsulation ppp
 no ip address
 ppp multilink
 tx-queue-limit 26
 multilink-group 6
 peer neighbor-route
!
interface Serial1/1/3:2
 encapsulation ppp
 no ip address
 ppp multilink
 tx-queue-limit 26
 multilink-group 6
 peer neighbor-route
```

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
!
controller T1 1/0
 framing esf
 clock source internal
 linecode b8zs
 channel-group 1 timeslots 1
 channel-group 2 timeslots 2
 no yellow generation
 no yellow detection
!
interface Serial1/0:1
 no ip address
 encapsulation ppp
 tx-ring-limit 26
 ppp multilink
 ppp multilink group 1
!
interface Serial1/0:2
 no ip address
 encapsulation ppp
 tx-ring-limit 26
 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 or Distributed Cisco Express Forwarding

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

```
enable
configure terminal
ip cef
```

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

```
enable
configure terminal
ip cef distribute
```

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

interface serial 1/0/0/:1
 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp multilink
 ppp multilink group 1
interface serial 1/0/0/:2
 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
interface serial 1/0/0/:3
 no ip address
 encapsulation ppp
 ip route-cache cef
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
interface serial 1/0/0/:4
 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 8: Feature Information for MPLS Multilink PPP Support

Feature Name	Releases	Feature Information
MPLS Multilink PPP Support	12.2(8)T 12.2(15)T10 12.2(28)SB 12.3(5a) 12.3(7)T 12.4(20)T 15.4(1)S	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). In 12.2(8)T, MLP support on CE-to-PE links was introduced. In 12.2(15)T10 and 12.3(5a), MLP support for MPLS networks was extended to PE-to-P links, PE-to-PE links, Carrier Supporting Carrier (CSC) CE-to-PE links, and interautonomous system (Inter-AS) PE-to-PE links. In 12.3(7)T, 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 Release 15.4(1)S, support was added for the Cisco ASR 901S Router.

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.