

MPLS Transport Profile

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

- Reference the Chapter Map here, on page 1
- Restrictions for MPLS Transport Profile, on page 1
- Information About MPLS-TP, on page 2
- How to Configure MPLS Transport Profile, on page 6
- Configuration Examples for MPLS Transport Profile, on page 25
- Additional References for MPLS Transport Profile, on page 26
- Feature Information for MPLS Transport Profile, on page 26

Reference the Chapter Map here

Restrictions for MPLS Transport Profile

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

L2VPN Restrictions

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

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

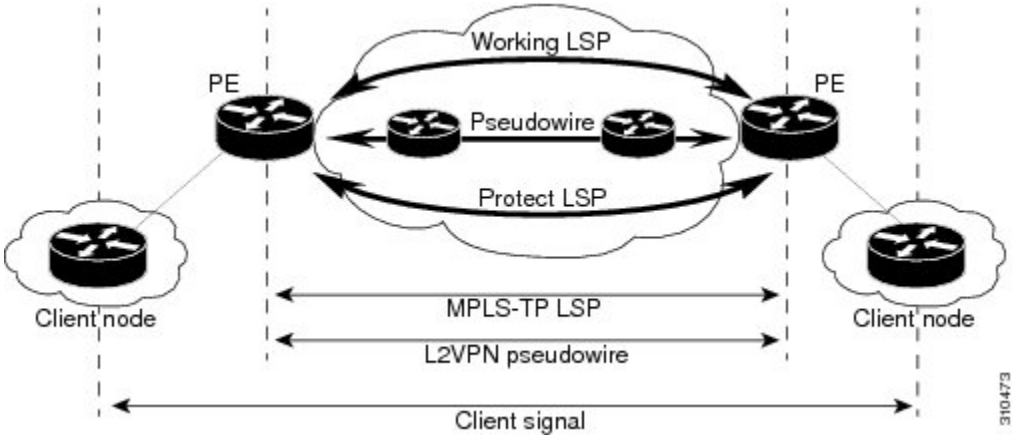
Ping and Trace Restrictions

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

Information About MPLS-TP

How MPLS Transport Profile Works

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



MPLS-TP Path Protection

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

Bidirectional LSPs

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

Support for MPLS Transport Profile OAM

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

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

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

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

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

MPLS Transport Profile Static and Dynamic Multisegment Pseudowires

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

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

MPLS-TP OAM Status for Static and Dynamic Multisegment Pseudowires

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

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

MPLS Transport Profile Links and Physical Interfaces

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

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

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

Link numbers must be unique on the router or node.

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

Tunnel Midpoints

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

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

How to Configure MPLS Transport Profile

Configuring the MPLS Label Range

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

Configuring the Router ID and Global ID

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls tp**
4. **router-id** *node-id*
5. **global-id** *num*
6. **end**

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
		• 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

Procedure

	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.

	Command or Action	Purpose
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

Procedure

	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:**	Specifies the OAM timeout refresh interval.

	Command or Action	Purpose
	`Device(config-st-pw-oam-class)# timeout refresh send 20`	
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

Procedure

	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:**	Creates a pseudowire class and enters pseudowire class configuration mode.

	Command or Action	Purpose
	`Device(config)# pseudowire-class mpls-tp-class1`	
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Device(config-pw-class)# control-word`	Enables the use of the control word.
Step 6	**protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Device(config-pw-class)# protocol none`	Specifies the type of protocol.
Step 7	**preferred-path** {**interface tunnel** *tunnel-number* | **peer** {*ip-address* | *host-name*}} [**disable-fallback**] **Example:** `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

Procedure

	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.
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.

	Command or Action	Purpose
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

Procedure

	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.
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:**	Specifies the BFD template.

	Command or Action	Purpose
	`Device(config-if)# bfd mpls-tp-bfd-2`	
Step 10	**working-lsp** **Example:** `Device(config-if)# working-lsp`	Specifies a working LSP, also known as the primary LSP.
Step 11	**in-label** *num* **Example:** `Device(config-if-working)# in-label 111`	Specifies the in-label number.
Step 12	**out-label** *num* **out-link** *num* **Example:** `Device(config-if-working)# out-label 112 out-link`	Specifies the out-label number and out-link.
Step 13	**exit** **Example:** `Device(config-if-working)# exit`	Exits working LSP interface configuration mode and returns to interface configuration mode.
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

Procedure

	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.

	Command or Action	Purpose
Step 4	**forward-lsp** **Example:** `Device(config-mpls-tp-lsp)# forward-lsp`	Enters MPLS-TP LSP forward LSP configuration mode.
Step 5	**bandwidth** *num* **Example:** `Device(config-mpls-tp-lsp-forw)# bandwidth 100`	Specifies the bandwidth.
Step 6	**in-label** *num* **out-label** *num* **out-link** *num* **Example:** `Device(config-mpls-tp-lsp-forw)# in-label 53 out-label 43 out-link 41`	Specifies the in label, out label, and out link numbers.
Step 7	**exit** **Example:** `Device(config-mpls-tp-lsp-forw)# exit`	Exits MPLS-TP LSP forward LSP configuration mode.
Step 8	**reverse-lsp** **Example:** `Device(config-mpls-tp-lsp)# reverse-lsp`	Enters MPLS-TP LSP reverse LSP configuration mode.
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

Procedure

	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

	Command or Action	Purpose
		the MPLS-TP link from one interface to another without reconfiguring all the MPLS-TP tunnels and LSPs that refer to the link. Link numbers must be unique on the device or node.
Step 6	**ip rsvp bandwidth** [**rdm** [**bc0** *interface-bandwidth*] [[*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]]] [*interface-bandwidth* [*single-flow-bandwidth* [**bc1** *bandwidth* | **sub-pool** *bandwidth*]] | **mam max-reservable-bw** [*interface-bandwidth* [*single-flow-bandwidth*] [**bc0** *interface-bandwidth* [**bc1** *bandwidth*]]] | **percent** *percent-bandwidth* [*single-flow-bandwidth*]] **Example:** `Device(config-if)# ip rsvp bandwidth 1158 100`	Enables Resource Reservation Protocol (RSVP) bandwidth for IP on an interface. For the Cisco 7600 platform, if you configure non-zero bandwidth for the TP tunnel or at a midpoint LSP, make sure that the interface to which the output link is attached has enough bandwidth available. For example, if three tunnel LSPs run over link 1 and each LSP was assigned 1000 with the **tp bandwidth** command, the interface associated with link 1 needs bandwidth of 3000 with the **ip rsvp bandwidth** command.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 8	**show mpls tp link-numbers** **Example:** `Device# show mpls tp link-numbers`	Displays the configured links.

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
	`Device(config-vfi)# mpls label 102 202`	
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

Procedure

	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.

	Command or Action	Purpose
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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *class-name* **Example:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-pw-class)# encapsulation mpls`	Specifies the encapsulation type.
Step 5	**control-word** **Example:** `Device(config-pw-class)# control-word`	Enables the use of the control word.
Step 6	**protocol** {**l2tpv2** | **l2tpv3** | **none**} [*l2tp-class-name*] **Example:** `Device(config-pw-class)# protocol none`	Specifies the type of protocol. Use the **protocol none** command to specify a static pseudowire.
Step 7	**exit** **Example:** `Device(config-pw-class)# exit`	Exits pseudowire class configuration mode and returns to global configuration mode.
Step 8	**pseudowire-class** *class-name* **Example:** `Device(config)# pseudowire-class mpls-tp-class1`	Creates a pseudowire class and enters pseudowire class configuration mode.
Step 9	**encapsulation mpls** **Example:**	Specifies the encapsulation type.

	Command or Action	Purpose
	`Device(config-pw-class)# encapsulation mpls`	
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* **Example:**	Specifies the TLV parameters or invokes a previously configured TLV template.

	Command or Action	Purpose
	`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.