Configuring MPLS Traffic Engineering and Enhancements

- Prerequisites for MPLS Traffic Engineering and Enhancements, on page 1
- Restrictions for MPLS Traffic Engineering and Enhancements, on page 1
- Information About MPLS Traffic Engineering and Enhancements, on page 2
- How to Configure MPLS Traffic Engineering and Enhancements, on page 7
- Configuration Examples for MPLS Traffic Engineering and Enhancements, on page 15
- Additional References, on page 18
- Feature History for MPLS Traffic Engineering and Enhancements, on page 19

Prerequisites for MPLS Traffic Engineering and Enhancements

Ensure that your network supports the following Cisco IOS features before you enable MPLS TE:

- Multiprotocol Label Switching
- IP Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF)

Restrictions for MPLS Traffic Engineering and Enhancements

- MPLS TE fast reroute is not supported.
- MPLS TE supports only a single IGP process or instance. Multiple IGP processes or instances are not supported and MPLS TE should not be configured in more than one IGP process or instance.
- The MPLS TE feature does not support routing and signaling of LSPs over unnumbered IP address links. Therefore, do not configure the feature over those links.
- When specifying an explicit path, if you specify the *forward* address (the address of the interface that forwards the traffic to the next router) as the next-hop address, the explicit path might not be used. Using the forward address allows that entry to be treated as a loose hop for path calculation. We recommend that you use the *receive* address (the address of the interface that receives traffic from the sending router) as the next-hop address.

In the following example, switch S3 sends traffic to switch S1. The paths marked a,b and x,y between switches S1 and S2 are parallel paths.

```
S1(a)----(b)S2(c)--(d)S3
  (x)----(y)
```

If you configure an explicit path from S3 to S1 using the *forward* addresses (addresses d and b), the tunnel might reroute traffic over the parallel path (x,y) instead of the explicit path. To ensure that the tunnel uses the explicit path, specify the *receive* addresses as part of the **next-address** command, as shown in the following example:

```
ip explicit-path name path1
  next-address (c)
  next-address (a)
```

Information About MPLS Traffic Engineering and Enhancements

The following sections provide information about MPLS TE and enhancements.

Introduction to MPLS Traffic Engineering and Enhancements

MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what can be achieved only by overlaying a Layer 3 network on a Layer 2 network.

Traffic engineering is essential for service provider and ISP backbones. Such backbones must support a high use of transmission capacity, and the networks must be resilient so that they can withstand link or node failures.

MPLS traffic engineering provides an integrated approach to traffic engineering. With MPLS, traffic engineering capabilities are integrated into Layer 3, which optimizes the routing of IP traffic, given the constraints imposed by backbone capacity and topology.

MPLS traffic engineering supports the following functionalities:

- Enhances standard Interior Gateway Protocols (IGPs), such as IS-IS or OSPF, to map packets to the appropriate traffic flows automatically.
- Transports traffic flows across a network using MPLS forwarding.
- Determines the routes for traffic flows across a network. This is based on the resources the traffic flow requires and the resources available in the network.
- Employs Constraint-based routing, in which the path for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow. In MPLS traffic engineering, the traffic flow has bandwidth requirements, media requirements, a priority that is compared to the priority of other flows, and so forth.
- Recovers from link or node failures by adapting to the new constraints presented by the changed topology.
- Transports packets using MPLS forwarding across a multihop label-switched path (LSP).
- Uses the routing and signaling capability of LSPs across a backbone topology that:
    - Understands the backbone topology and available resources.

- Accounts for link bandwidth and size of traffic flow when determining routes for LSPs across the backbone.
- Has a dynamic adaptation mechanism that enables the backbone to be resilient to failures, even if several primary paths are precalculated offline.
- Includes enhancements to the IGP (IS-IS or OSPF) shortest path first (SPF) calculations to automatically calculate which traffic should be sent over which LSPs.

Benefits of MPLS Traffic Engineering

A WAN connection is an expensive item in an ISP budget. Traffic engineering enables ISPs to route network traffic to offer the best service to their users in terms of throughput and delay. By making the service provider more efficient, traffic engineering reduces the cost of the network.

Currently, some ISPs base their services on an overlay model. In this model, transmission facilities are managed by Layer 2 switching. The routers see only a fully meshed virtual topology, making most destinations appear one hop away. If you use the explicit Layer 2 transit layer, you can precisely control how traffic uses the available bandwidth. However, the overlay model has numerous disadvantages. MPLS traffic engineering achieves the traffic engineering benefits of the overlay model without running a separate network, and without needing a nonscalable, full mesh of router interconnects.

How MPLS Traffic Engineering Works

MPLS TE automatically establishes and maintains LSPs across the backbone by using RSVP. The path that an LSP uses is determined by the LSP resource requirements and network resources, such as bandwidth.

Available resources are flooded by means of extensions to a link state-based IGP.

Traffic engineering tunnels are calculated at the LSP head, based on a fit between the required and available resources (constraint-based routing). The IGP automatically routes the traffic to these LSPs. Typically, a packet traveling across the MPLS TE backbone travels on a single LSP that connects the ingress point to the egress point.

MPLS traffic engineering is built on the following Cisco IOS mechanisms:

- IP tunnel interfaces

  From a Layer 2 standpoint, an MPLS tunnel interface represents the head of an LSP. It is configured with a set of resource requirements, such as bandwidth, media requirements, and priority.

  From a Layer 3 standpoint, an LSP tunnel interface is the headend of a unidirectional virtual link to the tunnel destination.

- MPLS traffic engineering path calculation module

  This calculation module operates at the LSP head. The module determines a path to use for an LSP. The path calculation uses a link-state database containing flooded topology and resource information.

- RSVP with traffic engineering extensions

  RSVP operates at each LSP hop and is used to signal and maintain LSPs based on the calculated path.

- MPLS traffic engineering link management module

This module operates at each LSP hop. It enables link call admission on the RSVP signaling messages, and bookkeeping of topology and resource information to be flooded.

- Link-state IGP (IS-IS or OSPF, each with traffic engineering extensions)

  These IGPs are used to globally flood topology and resource information from the link management module.

- Enhancements to the SPF calculation used by the link-state IGP (IS-IS or OSPF)

  The IGP automatically routes traffic to the appropriate LSP tunnel based on tunnel destination. Static routes can also be used to direct traffic to LSP tunnels.

- Label switching forwarding

  This forwarding mechanism provides routers with a Layer 2-like ability to direct traffic across multiple hops of the LSP established by RSVP signaling.

One approach to engineering a backbone is to define a mesh of tunnels from every ingress device to every egress device. The MPLS TE path calculation and signaling modules determine the path taken by the LSPs for these tunnels, subject to resource availability and the dynamic state of the network. The IGP, operating in an ingress device, determines which traffic should go to which egress device, and steers that traffic into the tunnel from ingress to egress.

A flow from an ingress device to an egress device might be so large that it cannot fit over a single link. And it cannot be carried by a single tunnel. In such a scenario, multiple tunnels between a given ingress and egress can be configured, and the flow load can be shared among them.

Mapping Traffic into Tunnels

This section describes how traffic is mapped to tunnels. It describes how conventional hop-by-hop link-state routing protocols interact with MPLS TE capabilities. This section describes how the shortest path first (SPF) algorithm, sometimes called a Dijkstra algorithm, is enhanced. This enhancement allows a link-state IGP to forward traffic automatically over tunnels that MPLS traffic engineering establishes.

Link-state protocols, such as integrated IS-IS or OSPF, use an SPF algorithm to compute a shortest path tree from the headend node to all the nodes in the network. Routing tables are derived from this shortest path tree. The routing tables contain ordered sets of destination and first-hop information. If a router does normal hop-by-hop routing, the first hop is over a physical interface attached to the router.

New traffic engineering algorithms calculate explicit routes to one or more nodes in the network. The originating router views these explicit routes as logical interfaces. In the context of this document, these explicit routes are represented by LSPs and referred to as traffic engineering tunnels (TE tunnels).

The following sections describe how link-state IGPs can use these shortcuts, and how they can install routes in the routing table that point to these TE tunnels. These tunnels use explicit routes. The path taken by a TE tunnel is controlled by the router that is the headend of the tunnel. In the absence of errors, TE tunnels are guaranteed to not loop, but routers must agree on how to use the TE tunnels. Otherwise, traffic might loop through two or more tunnels.

When specifying an explicit path for an MPLS TE tunnel, you can specify link or node addresses of the next-hop routers in an explicit path. You can also specify a mixture of link and node addresses. There are no restrictions when specifying a mixture of link and node addresses.

Transition of an IS-IS Network to a New Technology

IS-IS, as specified in RFC 1142, includes extensions for MPLS TE. Running MPLS traffic engineering over IS-IS or taking advantage of these other extensions requires transitioning an IS-IS network to this new technology. This section describes these extensions. It discusses two ways in which to migrate an existing IS-IS network from the standard ISO 10589 protocol towards the version of IS-IS specified in RFC 1142. Running MPLS TE over an existing IS-IS network requires a transition to the version of IS-IS specified in RFC 1142. However, running MPLS TE over OSPF does *not* require any similar network transition.

Extensions for the IS-IS Routing Protocol

Extensions for the IS-IS routing protocol serve the following purposes:

- Remove the 6-bit limit on link metrics.
- Allow interarea IP routes.
- Enable IS-IS to carry different kinds of information for traffic engineering. In the future, more extensions might be needed.

To serve these purposes, two new type, length, and value (TLV)objects have been defined:

- TLV 22 describes links (or adjacencies). It serves the same purpose as the IS neighbor option in ISO 10589 (TLV 2).
- TLV 135 describes reachable IP prefixes. It is similar to the IP neighbor options from RFC 1195 (TLVs 128 and 130).

Note For ease of use, these two new TLVs, 22 and 135, are referred to as new-style TLVs and TLVs 2, 128, and 130 are referred to as old-style TLVs.

Both the new TLVs have a fixed length part, followed by optional sub-TLVs. The metric space in these new TLVs has been enhanced from 6 bits to 24 or 32 bits. The sub-TLVs allow you to add new properties to links and prefixes. Traffic engineering is the first technology to use this capability to add new properties to a link.

Solution 1 for Transitioning an IS-IS Network to a New Technology

When you migrate from old-style TLVs towards new-style TLVs, you can advertise the same information twice—once in old-style TLVs and once in new-style TLVs. This ensures that all the devices can understand what is advertised.

There are three disadvantages to using that approach:

- Size of the LSPs: During the transition, the LSPs grow to about twice their original size. This might be a problem in networks where the LSP database (LSPDB) is large. An LSP database might be large because:
  - There are many devices, and therfore, many LSPs.
  - There are many neighbors or IP prefixes per router. A device that advertises lots of information causes the LSPs to be fragmented.

- Unpredictable results: In a large network, this approach can produce unpredictable results. A large network that is in transition pushes the limits with regard to LSP flooding and SPF scaling.
- Ambiguity: If a device encounters different information in the old-style TLVs and the new-style TLVs, it may not be clear what the device should do.
  - You can expect some extra network instability. At this time, you must not test how far you can push an implementation.
  - Traffic engineering extensions might cause LSPs to be reflooded frequently.

Most of these problems can be solved easily by using:

- All the information in the old-style and new-style TLVs in an LSP.
- The adjacency with the lowest link metric if an adjacency is advertised more than once.

The main benefit to advertising the same information twice is that network administrators can use new-style TLVs before all the devices in the network can understand them.

Transition Actions During Solution 1

When transitioning from using IS-IS with old-style TLVs to new-style TLVs, perform the following actions:

- If all the devices run old software, advertise and use only old-style TLVs.
- Upgrade some devices to newer software.
- Configure some devices with new software to advertise both old-style and new-style TLVs. They accept both styles of TLVs. Configure other devices (with old software) to continue advertising and using only old-style TLVs.
- Test traffic engineering in parts of your network.
- If the whole network needs to migrate, upgrade and configure all the remaining devices to advertise and accept both styles of TLVs.
- Configure all the devices to advertise and accept only new-style TLVs.
- Configure metrics larger than 63.

Solution 2 for Transitioning an IS-IS Network to a New Technology

Devices advertise only one style of TLVs at the same time, but can understand both types of TLVs during migration. There are two main benefits to this approach:

- LSPs stay approximately the same size during migration.
- There is no ambiguity when the same information is advertised twice inside one LSP.

This method is useful when you are transitioning the whole network (or a whole area) to use wider metrics (that is, you want a router running IS-IS to generate and accept only new-style TLVs).

The disadvantage is that all devices must understand the new-style TLVs before any router can start advertising new-style TLVs. It does not help the second problem, where network administrators want to use the new-style TLVs for traffic engineering, while some devices are capable of understanding only old-style TLVs.

Transition Actions During Solution 2

If you use the second solution, you can perform the following actions:

- If all the devices run old software, advertise and use only old-style TLVs.
- Upgrade all the devices to newer software.
- Configure all the devices one-by-one to advertise old-style TLVs, but to accept both styles of TLVs.
- Configure all the devices one-by-one to advertise new-style TLVs, but to accept both styles of TLVs.
- Configure all the devices one-by-one to advertise and to accept only new-style TLVs.
- Configure metrics larger than 63.

TLV Configuration Commands

You can use the **metric-style** command to configure the type of TLVs that are accepted by a device. When the device is in IS-IS configuration mode, you can configure the following keywords in the **metric-style** command.

- **metric-style narrow** : Enables the device to generate and accept only old-style TLVs
- **metric-style transition** : Enables the device to generate and accept both old-style and new-style TLVs
- **metric-style wide** : Enables the device to generate and accept only new-style TLVs

You can use either of the following transition schemes when you use the **metric-style** command:

- Narrow to transition to wide.
- Narrow to narrow transition to wide transition to wide.

Implementation in Cisco IOS XE Software

Cisco IOS XE can implement both transition solutions. Network administrators can choose the solution that suits them best. For test networks, solution 1 is best (see Solution 1 for Transitioning an IS-IS Network to a New Technology, on page 5). For a full transition, both solutions can be used. Solution 1 requires fewer steps and less configuration. Solution 2 is for the largest networks, where a risk of doubling the LSP database during transition exists (see Solution 2 for Transitioning an IS-IS Network to a New Technology, on page 6).

How to Configure MPLS Traffic Engineering and Enhancements

The following sections provide information about the steps to configure the MPLS Traffic Engineering and Enhancements feature.

Configuring a Device to Support Tunnels

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	**ip cef** **Example:** `Device(config)# ip cef`	Enables standard Cisco Express Forwarding operation.
Step 4	**mpls traffic-eng tunnels** **Example:** `Device(config)# mpls traffic-eng tunnels`	Enables MPLS traffic engineering tunnels on a device.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.

Configuring an Interface to Support RSVP-Based Tunnel Signaling and IGP Flooding

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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type slot / subslot / port* [*subinterface-number*] **Example:** `Device(config)# interface Port-channel 114`	Configures an interface type and enters interface configuration mode.
Step 4	**mpls traffic-eng tunnels** **Example:** `Device(config-if)# mpls traffic-eng tunnels`	Enables MPLS traffic engineering tunnels on an interface.
Step 5	**ip rsvp bandwidth** *bandwidth* **Example:** `Device(config-if)# ip rsvp bandwidth 1000`	Enables RSVP on an interface and specifies the amount of bandwidth that will be reserved.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IS-IS for MPLS Traffic Engineering

Note MPLS traffic engineering supports only a single IGP process or instance. Multiple IGP processes or instances are not supported. MPLS traffic engineering should not be configured in more than one IGP process or instance.

To configure IS-IS for MPLS traffic engineering, perform the following steps.

Procedure

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** **Example:** `Device(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process. The device enters configuration mode.
Step 4	**mpls traffic-eng level** **Example:** `Device(config-router)# mpls traffic-eng level-1`	Turns on MPLS traffic engineering for IS-IS level 1.
Step 5	**mpls traffic-eng level** **Example:** `Device(config-router)# mpls traffic-eng level-2`	Turns on MPLS traffic engineering for IS-IS level 2.
Step 6	**mpls traffic-eng router-id** *type number* **Example:** `Device(config-router)# mpls traffic-eng router-id loopback 0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface loopback0.
Step 7	**metric-style wide** **Example:** `Device(config-router)# metric-style wide`	Configures a router to generate and accept only new-style type, length, value objects (TLVs).

Configuring OSPF for MPLS Traffic Engineering

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.

	Command or Action	Purpose
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 200`	Configures an OSPF routing process for IP and enters router configuration mode. • The value for the *process-id* argument is an internally used identification parameter for an OSPF routing process. It is locally assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**mpls traffic-eng area** *number* **Example:** `Device(config-router)# mpls traffic-eng area 0`	Turns on MPLS TE for the indicated OSPF area.
Step 5	**mpls traffic-eng router-id loopback0** **Example:** `Device(config-router)# mpls traffic-eng router-id loopback0`	Specifies that the TE router identifier for the node is the IP address associated with interface loopback0.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits to global configuration mode.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel

To configure a preferred explicit path for an MPLS TE tunnel, perform this procedure.

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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface Tunnel0`	Configures an interface type and enters interface configuration mode. • The *number* argument is the number of the tunnel.
Step 4	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered loopback0`	Enables IP processing on an interface without assigning an explicit IP address to the interface. • The *type* and *number* arguments name the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface. • An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Device(config-if)# tunnel destination 192.168.4.4`	Specifies the destination for a tunnel interface. • The *ip-address* argument must be the MPLS traffic engineering router ID of the destination device.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Device(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Device(config-if)# tunnel mpls traffic-eng bandwidth 250`	Configures the bandwidth for the MPLS traffic engineering tunnel. • The *bandwidth* argument is a number in kilobits per second that is set aside for the MPLS traffic engineering tunnel. Range is from 1 through 4294967295. **Note** If automatic bandwidth is configured for the tunnel, use the **tunnel mpls traffic-eng bandwidth** command to configure the initial tunnel bandwidth.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | **identifier** *path-number*}} [**lockdown**]	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database.

	Command or Action	Purpose
	Example: `Device(config-if)# tunnel mpls traffic-eng path-option 10 explicit identifier 321`	• The *number* argument is the preference for this path option. When you configure multiple path options, lower numbered options are preferred. Valid values are from 1 through 1000. • The **dynamic** keyword indicates that the path of the LSP is dynamically calculated. • The **explicit** keyword indicates that the path of the LSP is an IP explicit path. • The **name** *path-name* keyword and argument are the path name of the IP explicit path that the tunnel uses with this option. • The **identifier** *path-number* keyword and argument pair names the path number of the IP explicit path that the tunnel uses with this option. The range is from 1 through 65535. • The **lockdown** keyword specifies that the LSP cannot be reoptimized. **Note** A dynamic path is used if an explicit path is currently unavailable.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring an MPLS Traffic Engineering Tunnel that an IGP Can Use

To configure an MPLS TE tunnel that an IGP can use, perform this procedure.

Procedure

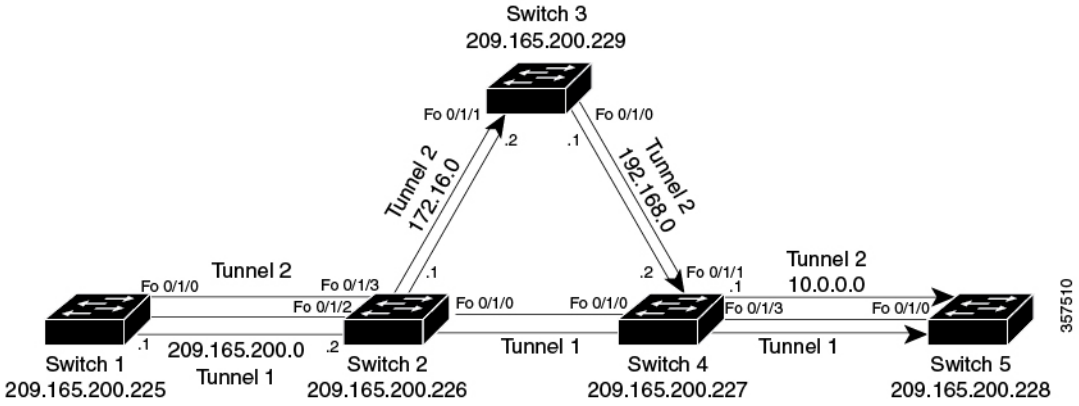
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel10`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Gives the tunnel interface an IP address. An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Device(config-if)# tunnel destination 10.20.1.1`	Specifies the destination for a tunnel. The *ip-address* keyword is the IP address of the host destination expressed in dotted decimal notation.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Device(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Device(config-if)# tunnel mpls traffic-eng bandwidth 1000`	Configures the bandwidth for the MPLS traffic engineering tunnel.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name*} | **identifier** *path-number*} [**lockdown**] **Example:** `Device(config-if)# tunnel mpls traffic-eng path-option 1 explicit identifier 1`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. If an explicit path is currently unavailable a dynamic path is used.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuration Examples for MPLS Traffic Engineering and Enhancements

The following figure illustrates a sample MPLS topology. This example specifies point-to-point outgoing interfaces. The subsequent sections contain sample configuration commands that you should enter to implement MPLS traffic engineering and the basic tunnel configuration shown in Figure 3.

Figure 1: Sample MPLS Traffic Engineering Tunnel Configuration



Example: Configuring MPLS Traffic Engineering Using IS-IS

This example lists the commands you should enter to configure MPLS TE with IS-IS routing enabled (see Figure 1).

Note Enter the following commands in every router in the traffic-engineered portion of your network.

Device 1: MPLS Traffic Engineering Configuration

To configure MPLS traffic engineering, enter the following commands:

```
ip cef
mpls traffic-eng tunnels
interface loopback 0
ip address 10.0.0.0 255.255.255.254
ip router isis
interface Fo 1/0/0
ip address 209.165.200.1 255.255.0.0
ip router isis
mpls traffic-eng tunnels
ip rsvp bandwidth 1000
```

Device 1: IS-IS Configuration

To enable IS-IS routing, enter the following commands:

```
router isis
network 47.0000.0011.0011.00
is-type level-1
metric-style wide
mpls traffic-eng router-id loopback0
mpls traffic-eng level-1
```

Example: Configuring MPLS Traffic Engineering Using OSPF

This example lists the commands that you should enter to configure MPLS traffic engineering with OSPF routing enabled (see Figure 1).

Note Enter the following commands in every router in the traffic-engineered portion of your network.

Device 1: MPLS Traffic Engineering Configuration

To configure MPLS traffic engineering, enter the following commands:

```
ip cef
mpls traffic-eng tunnels
interface loopback 0
ip address 209.165.200.225 255.255.255.255
interface Fo 1/0/0
ip address 209.165.200.1 255.255.0.0
mpls traffic-eng tunnels
  ip rsvp bandwidth 1000
```

Device 1: OSPF Configuration

To enable OSPF, enter the following commands:

```
router ospf 0
network 209.165.200.0.0.0.255.255 area 0
mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
```

Example: Configuring an MPLS Traffic Engineering Tunnel

This example shows you how to configure a dynamic path tunnel and an explicit path in the tunnel. Before you configure MPLS traffic engineering tunnels, enter the appropriate global and interface commands in the specified router (in this case, Router 1).

Device 1: Dynamic Path Tunnel Configuration

To configure a tunnel to use a dynamic path, enter the following commands:

```
interface tunnel1
ip unnumbered loopback 0
tunnel destination 209.165.200.228
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 100
```

```
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng path-option 1 dynamic
```

Device 1: Dynamic Path Tunnel Verification

To verify that the tunnel is up, enter the following commands:

```
show mpls traffic-eng tunnels
show ip interface tunnel1
```

Device 1: Explicit Path Configuration

To configure an explicit path, enter the following commands:

```
ip explicit-path identifier 1
next-address 209.165.200.1
next-address 172.16.0.1
next-address 192.168.0.1
next-address 10.0.0.1
```

Device 1: Explicit Path Tunnel Configuration

To configure a tunnel to use an explicit path, enter the following commands:

```
interface tunnel2
ip unnumbered loopback 0
tunnel destination 209.165.200.228
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng bandwidth 100
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng path-option 1 explicit identifier 1
```

Device 1: Explicit Path Tunnel Verification

To verify that the tunnel is up, enter the following commands:

```
show mpls traffic-eng tunnels
show ip interface tunnel2
```

Example: Configuring Enhanced SPF Routing over a Tunnel

This section includes the commands that cause a tunnel to be considered by the IGP's enhanced SPF calculation, that installs routes over the tunnel for appropriate network prefixes.

Device 1: IGP Enhanced SPF Consideration Configuration

To specify that the IGP should use the tunnel (if the tunnel is up) in its enhanced shortest path first (SPF) calculation, enter the following commands:

```
interface tunnel1
tunnel mpls traffic-eng autoroute announce
```

Device 1: Route and Traffic Verification

To verify that the tunnel is up and that the traffic is routed through the tunnel, enter the following commands:

```
#show mpls traffic-eng tunnels tu12001 brief
Signalling Summary:
LSP Tunnels Process: running
Passive LSP Listener: running
RSVP Process: running
Forwarding: enabled
auto-tunnel:
p2p Disabled (0), id-range:62336-64335

Periodic reoptimization: every 3600 seconds, next in 694 seconds
Periodic FRR Promotion: Not Running
Periodic auto-bw collection: every 300 seconds, next in 94 seconds
SR tunnel max label push: 2 primary path labels (2 repair path labels)
TUNNEL NAME DESTINATION UP IF DOWN IF STATE/PROT
tu12001 2.2.2.2 - Po114 up/up
```

Additional References

The following sections provide references related to the MPLS Traffic Engineering and Enhancements feature.

Related Documents

Related Topic	Document Title
IS-IS commands	*Cisco IOS IP Routing Protocols Command Reference*
OSPF command	*Cisco IOS IP Routing Protocols Command Reference*
MPLS TE commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
RSVP commands	*Cisco IOS Quality of Service Solutions Command Reference*

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
1142	*IS-IS*
1195	*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*
2205	*Resource ReSerVation Protocol (RSVP)*

RFC	Title
2328	*OSPF Version 2*
2370	*The OSPF Opaque LSA Option*

Technical Assistance

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

Feature History for MPLS Traffic Engineering and Enhancements

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Bengaluru 17.6.1	MPLS Traffic Engineering and Enhancements	Multiprotocol Label Switching (MPLS) is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what previously could be achieved only by overlaying a Layer 3 network on a Layer 2 network.
Cisco IOS XE Cupertino 17.7.1	MPLS Traffic Engineering and Enhancements	This feature was implemented on the C9500X-60L4D model.
Cisco IOS XE Dublin 17.10.1b	MPLS Traffic Engineering and Enhancements	This feature was implemented on the C9500X-60L4D model.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/