

MPLS Traffic Engineering AutoTunnel Mesh Groups

The MPLS Traffic Engineering Autotunnel Mesh Groups feature allows a network administrator to configure traffic engineering (TE) label switched paths (LSPs) by using a few command-line interface (CLI) commands.

In a network topology where edge TE label switch routers (LSRs) are connected by core LSRs, the MPLS Traffic Engineering--Autotunnel Mesh Groups feature automatically constructs a mesh of TE LSPs among the provider edge (PE) devices.

- Prerequisites for MPLS Traffic Engineering--AutoTunnel Mesh Groups, on page 1
- Restrictions for MPLS Traffic Engineering--AutoTunnel Mesh Groups, on page 1
- Information About MPLS Traffic Engineering--AutoTunnel Mesh Groups, on page 2
- How to Configure MPLS Traffic Engineering--AutoTunnel Mesh Groups, on page 3
- Configuration Examples for MPLS Traffic Engineering--Autotunnel Mesh Groups, on page 14
- Additional References, on page 15
- Feature Information for MPLS Traffic Engineering--Autotunnel Mesh Groups, on page 16
- Glossary, on page 17

Prerequisites for MPLS Traffic Engineering--AutoTunnel Mesh Groups

- Be knowledgeable about MPLS TE.
- Decide how you will set up autotunnels (that is, identify the tunnel commands that you will include in the template interface).
- Identify a block of addresses that you will reserve for mesh tunnel interfaces.

Restrictions for MPLS Traffic Engineering--AutoTunnel Mesh Groups

- Mesh groups do not support interarea tunnels because the destinations of those tunnels do not exist in the local area TE database.

- You cannot configure a static route to route traffic over autotunnel mesh group TE tunnels. You should use only the autoroute for tunnel selection.
- Intermediate System-to-System (IS-IS) does not support Interior Gateway Protocol (IGP) distribution of mesh group information. For IS-IS, only Access Control Lists (ACLs) can be used.

Information About MPLS Traffic Engineering--AutoTunnel Mesh Groups

AutoTunnel Mesh Groups Description and Benefits

An autotunnel mesh group (referred to as a mesh group) is a set of connections between edge LSRs in a network. There are two types of mesh groups:

- Full--All the edge LSRs are connected. Each PE device has a tunnel to each of the other PE devices.
- Partial--Some of the edge LSRs are not connected to each other by tunnels.

In a network topology where edge TE LSRs are connected by core LSRs, the MPLS Traffic Engineering--Autotunnel Mesh Groups feature automatically constructs a mesh of TE LSPs among the PE devices.

Initially, you must configure each existing TE LSR to be a member of the mesh by using a minimal set of configuration commands. When the network grows (that is, when one or more TE LSRs are added to the network as PE devices), you do not need to reconfigure the existing TE LSR members of that mesh.

Mesh groups have the following benefits:

- Minimize the initial configuration of the network. You configure one template interface per mesh, and it propagates to all mesh tunnel interfaces, as needed.
- Minimize future configurations resulting from network growth. The feature eliminates the need to reconfigure each existing TE LSR to establish a full mesh of TE LSPs whenever a new PE device is added to the network.
- Enable existing devices to configure TE LSPs to new PE devices.
- Enable the construction of a mesh of TE LSPs among the PE devices automatically.

Access Lists for Mesh Tunnel Interfaces

The access list determines the destination addresses for the mesh tunnel interfaces. It is useful if you preallocate a block of related IP addresses. You can use that block of addresses to control the PE devices to which a full or partial mesh of TE tunnel LSPs is established. The access list allows matches for only the addresses that are learned and stored in the TE topology database.

For example, you can create an access list that matches all 10.1.1.1 IP addresses. You configure a template with the access list, then the template creates mesh tunnel interfaces to destinations within the TE topology database that match destinations in that access list.

Whenever the TE topology database is updated (for example, when a new TE LSR is inserted into the Interior Gateway Protocol (IGP), the destination address is stored in the TE topology database of each device in the IGP. At each update, the Mesh Group feature compares the destination address contained in the database to IP addresses in the access list associated with all template interfaces. If there is a match, the Mesh Group feature establishes a mesh tunnel interface to the tunnel destination IP address.

AutoTunnel Template Interfaces

An autotunnel template interface is a logical entity; that is, it is a configuration for a tunnel interface that is not tied to specific tunnel interfaces. It can be applied dynamically, when needed.

Mesh tunnel interfaces are tunnel interfaces that are created, configured dynamically (for example, by the applying [or cloning] of a template interface), used, and then freed when they are no longer needed.

A mesh tunnel interface obtains its configuration information from a template, except for the tunnel's destination address, which it obtains from the TE topology database that matches an access list or from the IGP mesh group advertisement.

The template interface allows you to enter commands once per mesh group. These commands specify how mesh tunnel interfaces are created. Each time a new device is added to the network, a new mesh tunnel interface is created. The configuration of the interface is duplicated from the template. Each mesh tunnel interface has the same path constraints and other parameters configured on the template interface. Only the tunnel destination address is different.

OSPF Flooding of Mesh Group Information

For OSPF to advertise or flood mesh group information, you need to configure a mesh group in OSPF and add that mesh group to an autotemplate interface. When the configuration is complete, OSPF advertises the mesh group IDs to all LSRs. MPLS TE LSPs automatically connect the edge LSRs in each mesh group.

OSPF can advertise mesh group IDs for an OSPF area. OSPF is the only IGP supported in some software releases of the MPLS Traffic Engineering--Autotunnel Mesh Groups feature.

How to Configure MPLS Traffic Engineering--AutoTunnel Mesh Groups

Configuring a Mesh of TE Tunnel LSPs

Perform the following tasks on each PE device in your network to configure a mesh of TE tunnel LSPs:

Note You can perform these tasks in any order.

Enabling Autotunnel Mesh Groups Globally

Perform this task on all PE devices in your network that you want to be part of an autotunnel mesh group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel mesh**
4. **end**

DETAILED STEPS

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 traffic-eng auto-tunnel mesh** **Example:** `Device(config)# mpls traffic-eng auto-tunnel mesh`	Enables autotunnel mesh groups globally.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Creating an Access List Using a Name

The access list determines the destination addresses for the mesh tunnel interfaces. You can use an access list to control the PE devices to which a full or partial mesh of TE tunnel LSPs is established. The access list allows matches for only the addresses that are learned and stored in the TE topology database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} *access-list-name*
4. **permit** *source* [*source-wildcard*]
5. **end**

DETAILED STEPS

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 access-list** {**standard** | **extended**} *access-list-name* **Example:** `Device(config)# ip access-list standard a1`	Defines an IP access list using a name and enters standard named access list configuration mode. • The **standard** keyword specifies a standard IP access list. • The **extended** keyword specifies an extended IP access list. • The *access-list-name* argument is the name of the access list. A name cannot contain a space or quotation mark and must begin with an alphabetic character. This prevents confusion with numbered access lists.

Step 4	**permit** *source* [*source-wildcard*]
	Example:
	`Device(config-std-nacl)# permit 10.0.0.0 0.255.255.255`
		Sets conditions to allow a packet to pass a named IP access list.
		• The *source* argument is the number of the network or host from which the packet is being sent. There are three alternative ways to specify the source:
		  • Use a 32-bit quantity in four-part dotted decimal format.
		  • Use the **any** keyword as an abbreviation for a source and source-wildcard of 0.0.0.0 255.255.255.255.
		  • Use host source as an abbreviation for a source and source-wildcard of source 0.0.0.0.
		• The *source-wildcard* argument is the wildcard bits to be applied to source. There are three alternative ways to specify the source wildcard:
		  • Use a 32-bit quantity in four-part dotted decimal format. Place 1s in the bit positions you want to ignore.

	Command or Action	Purpose
		• Use the **any** keyword as an abbreviation for a source and source-wildcard of 0.0.0.0 255.255.255.255. • Use host source as an abbreviation for a source and source-wildcard of source 0.0.0.0.
Step 5	**end** **Example:** `Device(config-std-nacl)# end`	Exits to privileged EXEC mode.

Creating an Autotunnel Template Interface

Creating an autotunnel template interface helps minimize the initial configuration of the network. You configure one template interface per mesh, which propagates to all mesh tunnel interfaces, as needed.

Note You can use the following commands to create a minimal configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface auto-template** *interface-num*
4. **ip unnumbered** *interface-type interface-number*
5. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **ipip** | **iptalk** | **mpls** | **nos**}
6. **tunnel mpls traffic-eng autoroute announce**
7. **tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*]
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}} [**lockdown**]
9. **tunnel destination access-list** *num*
10. **end**

DETAILED STEPS

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 auto-template** *interface-num* **Example:** `Device(config)# interface auto-template 1`	Creates a template interface and enters interface configuration mode. • The *interface-num* argument is the interface number. Valid values are from 1 to 25.
Step 4	**ip unnumbered** *interface-type interface-number* **Example:** `Device(config-if)# ip unnumbered Loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface. • The *type* and *number* arguments name the type and number of another interface on which the device has an assigned IP address. It cannot be another unnumbered interface.
Step 5	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **ipip** | **iptalk** | **mpls** | **nos**} **Example:** `Device(config-if)# tunnel mode mpls`	Sets the encapsulation mode for the tunnel interface.
Step 6	**tunnel mpls traffic-eng autoroute announce** **Example:** `Device(config-if)# tunnel mpls traffic-eng autoroute announce`	Specifies that the IGP should use the tunnel (if the tunnel is up) in its enhanced shortest path first algorithm (SPF) calculation.
Step 7	**tunnel mpls traffic-eng priority** *setup-priority* [*hold-priority*] **Example:** `Device(config-if)# tunnel mpls traffic-eng priority 1 1`	Configures the setup and reservation priority for an MPLS TE tunnel. • The *setup-priority* argument is the priority used when an LSP is signaled for this tunnel and determines which existing tunnels can be preempted. Valid values are from 0 to 7, where a lower number indicates a higher priority. Therefore, an LSP with a setup priority of 0 can preempt any LSP with a non-0 priority. • The *hold-priority* argument is the priority associated with an LSP for this tunnel and determines if it should be preempted by other LSPs that are being signaled. Valid values are from 0 to 7, where a lower number indicates a higher priority.
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name* | *path-number*}} [**lockdown**]	Configures a path option for an MPLS TE tunnel.

	Command or Action	Purpose
	Example: `Device(config-if)# tunnel mpls traffic-eng path-option 1 dynamic`	• The *number* argument is the number of the path option. When multiple path options are configured, lower numbered options are preferred. • The **dynamic** keyword specifies that the path of the LSP is dynamically calculated. • The **explicit** keyword specifies that the path of the LSP is an IP explicit path. • The **name** *path-name* keyword-argument pair is the path name of the IP explicit path that the tunnel uses with this option. • The *path-number* argument is the path number of the IP explicit path that the tunnel uses with this option. • The **lockdown** keyword specifies that the LSP cannot be reoptimized.
Step 9	**tunnel destination access-list** *num* **Example:** `Device(config-if)# tunnel destination access-list 1`	Specifies the access list that the template interface uses for obtaining the mesh tunnel interface destination address. • The *num* argument is the number of the access list.
Step 10	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Specifying the Range of Mesh Tunnel Interface Numbers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel mesh tunnel-num min** *num* **max** *num*
4. **end**

DETAILED STEPS

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	**mpls traffic-eng auto-tunnel mesh tunnel-num min** *num* **max** *num* **Example:** `Device(config)# mpls traffic-eng auto-tunnel mesh tunnel-num min 1000 max 2000`	Specifies the range of mesh tunnel interface numbers. • The **min** *num* keyword-argument pair specifies the beginning number of the range of mesh tunnel interface numbers. Valid values are from 1 to 65535. • The **max** *num* keyword-argument pair specifies the ending number of the range of mesh tunnel interface numbers. Valid values are from 1 to 65535.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Displaying Configuration Information About Tunnels

SUMMARY STEPS

1. **enable**
2. **show running interface auto-template** *num*
3. **show interface tunnel** *num* **configuration**
4. **exit**

DETAILED STEPS

Procedure

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Device> enable
Device#
```

Step 2 **show running interface auto-template** *num*

Use this command to display interface configuration information for a tunnel interface. For example:

Example:

```
Device# show running interface auto-template 1
interface auto-template1
 ip unnumbered Loopback0
 no ip directed-broadcast
 no keepalive
 tunnel destination access-list 1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
```

This output shows that autotunnel template interface auto-template1 uses an access list (access-list 1) to determine the destination addresses for the mesh tunnel interfaces.

Step 3 **show interface tunnel** *num* **configuration**

Use this command to display the configuration of the mesh tunnel interface. For example:

Example:

```
Device# show interface tunnel 5 configuration
interface tunnel 5
 ip unnumbered Loopback0
 no ip directed-broadcast
 no keepalive
 tunnel destination access-list 1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
```

Step 4 **exit**

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

Example:

```
Device# exit
Device>
```

Monitoring the Autotunnel Mesh Network

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng tunnels property auto-tunnel mesh** [**brief**]
3. **show mpls traffic-eng auto-tunnel mesh**
4. **exit**

DETAILED STEPS

Procedure

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Device> enable
Device#
```

Step 2 **show mpls traffic-eng tunnels property auto-tunnel mesh** [**brief**]

Use this command to monitor mesh tunnel interfaces. This command restricts the output of the **show mpls traffic-eng tunnels** command to display only mesh tunnel interfaces. For example:

Example:

```
Device# show mpls traffic-eng tunnels property auto-tunnel mesh brief
Signalling Summary:
 LSP Tunnels Process:            running
 RSVP Process:                   running
 Forwarding:                     enabled
 Periodic reoptimization:        every 3600 seconds, next in 491 seconds
 Periodic FRR Promotion:         Not Running
 Periodic auto-bw collection:    disabled
TUNNEL NAME                      DESTINATION    UP IF     DOWN IF
STATE/PROT
device_t64336                    10.2.2.2        -        Se2/0
up/up
device_t64337                    10.3.3.3        -        Se2/0
up/up
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Step 3 **show mpls traffic-eng auto-tunnel mesh**

Use this command to display the cloned mesh tunnel interfaces of each autotemplate interface and the current range of mesh tunnel interface numbers. For example:

Example:

```
Device# show mpls traffic-eng auto-tunnel mesh
Auto-Template1:
 Using access-list 1 to clone the following tunnel interfaces:
  Destination  Interface
  -----------  ---------
  10.2.2.2     Tunnel64336
  10.3.3.3     Tunnel64337
Mesh tunnel interface numbers: min 64336 max 65337
```

Step 4 **exit**

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

Example:

```
Device# exit
Device>
```

Troubleshooting Tips

You can configure mesh tunnel interfaces directly. However, you cannot delete them manually, and manual configuration is not permanent. The configuration is overwritten when the template changes or the mesh tunnel interface is deleted and re-created. If you attempt to manually delete a mesh tunnel interface, an error message appears.

You can enter the **show mpls traffic-eng tunnels destination** *address* command to display information about tunnels that are destined for a specified IP address.

Enter the **show mpls traffic-eng tunnels property auto-tunnel mesh** command to display information about mesh tunnel interfaces.

Configuring IGP Flooding for Autotunnel Mesh Groups

Perform the following task to configure IGP flooding for autotunnel mesh groups. Use this task to configure an OSPF-based discovery for identifying mesh group members and advertising the mesh group IDs to all LSRs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng auto-tunnel mesh**
4. **router ospf** *process-id*
5. **mpls traffic-eng mesh-group** *mesh-group-id* *interface-type interface-number* **area** *area-id*
6. **exit**
7. Repeat steps 4 and 5 at other LSRs to advertise the mesh group numbers to which they belong.
8. **interface auto-template** *interface-num*
9. **tunnel destination mesh-group** *mesh-group-id*
10. **end**

DETAILED STEPS

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	**mpls traffic-eng auto-tunnel mesh** **Example:** `Device(config)# mpls traffic-eng auto-tunnel mesh`	Enables autotunnel mesh groups globally.
Step 4	**router ospf** *process-id* **Example:** `Device(config)# router ospf 100`	Enters router configuration mode and configures an OSPF routing process. • The *process-id*argument is an internally used identification parameter for an OSPF routing process. It is locally assigned and can be any positive integer. A unique value is assigned for each OSPF routing process.
Step 5	**mpls traffic-eng mesh-group** *mesh-group-id interface-type interface-number* **area** *area-id* **Example:** `Device(config-router)# mpls traffic-eng mesh-group 10 loopback 0 area 100`	Advertises the autotunnel mesh group number of an LSR. • The *mesh-group-id* is a number that identifies a specific mesh group. • The *interface-type* and *interface-number* arguments specify a type of interface and an interface number. • The **area** *area-id* keyword-argument pair identifies the area.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits to global configuration mode.
Step 7	Repeat steps 4 and 5 at other LSRs to advertise the mesh group numbers to which they belong.	--
Step 8	**interface auto-template** *interface-num* **Example:** `Device(config)# interface auto-template 1`	Creates a template interface and enters interface configuration mode. • The *interface-num* argument identifies the interface number. Valid values are from 1 to 25.
Step 9	**tunnel destination mesh-group** *mesh-group-id* **Example:** `Device(config-if)# tunnel destination mesh-group 10`	Specifies a mesh group that a template interface uses to signal tunnels for all mesh group members. • The *mesh-group-id* is a number that identifies a specific mesh group.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuration Examples for MPLS Traffic Engineering--Autotunnel Mesh Groups

Examples: Configuring a Mesh of TE Tunnel LSPs

This section contains the following configuration examples for configuring a mesh of TE tunnel LSP:

Example: Enabling Autotunnel Mesh Groups Globally

The following example shows how to enable autotunnel mesh groups globally:

```
configure terminal
!
mpls traffic-eng auto-tunnel mesh
end
```

Example: Creating an Access List Using a Name

The following examples shows how to create an access list using a name to determine the destination addresses for the mesh tunnel interfaces:

```
configure terminal
!
ip access-list standard a1
 permit 10.0.0.0 0.255.255.255
end
```

In this example, any IP address in the TE topology database that matches access list a1 causes the creation of a mesh tunnel interface with that destination address.

Example: Creating an AutoTunnel Template Interface

This example shows how to create an AutoTunnel template interface. In the following example, an AutoTunnel template is created and configured with a typical set of TE commands. The mesh group created from the template consists of mesh tunnel interfaces with destination addresses that match access list a1.

Note The following example shows a typical configuration.

```
configure terminal
!
```

```
interface auto-template 1
 ip unnumbered Loopback0
 tunnel mode mpls
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1

 tunnel mpls traffic-eng path-option 1 dynamic
 tunnel destination access-list a1
 end
```

Example: Specifying the Range of Mesh Tunnel Interface Numbers

In the following example, the lowest mesh tunnel interface number can be 1000, and the highest mesh tunnel interface number can be 2000:

```
configure terminal
!
mpls traffic-eng auto-tunnel mesh tunnel-num min 1000 max 2000
end
```

Example: Configuring IGP Flooding for Autotunnel Mesh Groups

In the following example, OSPF is configured to advertise the device membership in mesh group 10:

```
configure terminal
!
mpls traffic-eng auto-tunnel mesh
router ospf 100
 mpls traffic-eng mesh-group 10 loopback 0 area 100
 exit
!
interface auto-template 1
 tunnel destination mesh-group 10
 end
```

Additional References

Related Documents

Related Topic	Document Title
MPLS traffic engineering command descriptions	*Multiprotocol Label Switching Command Reference*

Technical Assistance

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

Feature Information for MPLS Traffic Engineering--Autotunnel Mesh Groups

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

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

Table 1: Feature Information for MPLS Traffic Engineering--Autotunnel Mesh Groups

Feature Name	Releases	Feature Information
MPLS Traffic Engineering--Autotunnel Mesh Groups	12.0(27)S 12.0(29)S 12.2(33)SRA 12.2(33)SXH 12.4(20)T 12.2(33)SRE Cisco IOS XE Release 3.6S	The MPLS Traffic Engineering--AutoTunnel Mesh Groups feature allows a network administrator to configure TE LSPs. In Cisco IOS Release 12.2(27)S, this feature was introduced. In Cisco IOS Release 12.0(29)S, this feature was updated to include Interior Gateway Protocol (IGP) flooding of autotunnel mesh groups. In Cisco IOS Release 12.2(33)SRA, this feature was integrated. In Cisco IOS Release 12.2(33)SXH, support was added. In Cisco IOS Release 12.4(20)T, this feature was integrated. In Cisco IOS Release 12.2(33)SRE, this feature was integrated. A device with autotunnel mesh groups can be configured with stateful switchover (SSO) redundancy. In Cisco IOS XE Release 3.6S, this feature was integrated. These commands were introduced or modified: **mpls traffic-eng auto-tunnel mesh**, **mpls traffic-eng auto-tunnel mesh tunnel-num**, **mpls traffic-eng mesh-group**, **show mpls traffic-eng auto-tunnel mesh**.
MPLS TE--Autotunnel/Auotmesh SSO Coexistence	Cisco IOS XE Release 3.5S 15.0(1)S	In Cisco IOS XE Release 3.5S, this feature was integrated. In Cisco IOS Release 15.0(1)S, this feature was integrated. **Note** Starting with Cisco IOS Release 15.2(2)S and Cisco IOS XE Release 3.6S, the SSO Support for MPLS TE Autotunnel and Automesh feature replaces the MPLS TE - Autotunnel/Automesh SSO Coexistence feature. For more information, see the *MPLS High Availability Configuration Guide* for the new implementation.

Glossary

CE device --customer edge device. A device that is part of a customer's network and interfaces to a provider edge (PE) device.

customer network --A network that is under the control of an end customer. Private addresses can be used in a customer network. Customer networks are logically isolated from each other and from the service provider's network.

edge device --A device at the edge of the network that receives and transmits packets. It can define the boundaries of the Multiprotocol Label Switching (MPLS) network.

headend --The label switch router (LSR) where a tunnel originates. The tunnel's “head” or tunnel interface resides at this LSR as well.

label --A short, fixed-length data construct that tells switching nodes how to forward data (packets).

label switched path (LSP) tunnel --A configured connection between two devices in which label switching is used to carry the packets.

LSP --label switched path. A path that a labeled packet follows over several hops, starting at an ingress LSR and ending at an egress LSR.

LSR --label switch router. A Layer 3 device that forwards a packet based on the value of a label encapsulated in the packet.

mesh group --A set of label switch devices (LSRs) that are members of a full or partial network of traffic engineering (TE) label switched paths (LSPs).

P device --provider core device.

PE device --provider edge device. A device at the edge of the service provider’s network that interfaces to customer edge (CE) devices.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

tailend --The downstream, receive end of a tunnel.

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

tunnel --A secure communication path between two peers, such as two devices. A traffic engineering tunnel is a label switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing.