Implementing MPLS Traffic Engineering

Multiprotocol Label Switching (MPLS) is a standards-based solution driven by the Internet Engineering Task Force (IETF) that was devised to convert the Internet and IP backbones from best-effort networks into business-class transport mediums.

MPLS, with its label switching capabilities, eliminates the need for an IP route look-up and creates a virtual circuit (VC) switching function, allowing enterprises the same performance on their IP-based network services as with those delivered over traditional networks such as Frame Relay or Asynchronous Transfer Mode (ATM).

MPLS traffic engineering (MPLS-TE) software enables an MPLS backbone to replicate and expand upon the TE capabilities of Layer 2 ATM and Frame Relay networks. MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what now can be achieved only by overlaying a Layer 3 network on a Layer 2 network.

The LMP and GMPLS-NNI features are not supported on PRP hardware.

### Feature History for Implementing MPLS-TE

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 5.0.0</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
| Release 5.2.1 | Support was added for these features: 
  • Point-to-Multipoint Traffic-Engineering
  • Policy-Based Tunnel Selection |
| Release 5.2.5 | Interarea P2MP Path Expansion within a Domain feature was added. |
| Release 6.1.2 | Named Tunnel feature was added.                        |
| Release 6.4.1 | Enabling Forward Class Zero in PBTS feature was added. |
Prerequisites for Implementing Cisco MPLS Traffic Engineering

These prerequisites are required to implement MPLS TE:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Router that runs Cisco IOS XR software.
- Installed composite mini-image and the MPLS package, or a full composite image.
- IGP activated.
- To configure Point-to-Multipoint (P2MP)-TE, a base set of RSVP and TE configuration parameters on ingress, midpoint, and egress nodes in the MPLS network is required. In addition, Point-to-Point (P2P) parameters are required.

Information About Implementing MPLS Traffic Engineering

To implement MPLS-TE, you should understand these concepts:

Overview of MPLS Traffic Engineering

MPLS-TE software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what now can be achieved only by overlaying a Layer 3 network on a Layer 2 network.

MPLS-TE is essential for service provider and Internet service provider (ISP) backbones. Such backbones must support a high use of transmission capacity, and the networks must be very resilient so that they can withstand link or node failures. MPLS-TE 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.

Related Topics
- Configuring Forwarding over the MPLS-TE Tunnel, on page 26

Benefits of MPLS Traffic Engineering

MPLS-TE 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 the overlay model, transmission facilities are managed by Layer 2 switching. The routers see only a fully meshed virtual topology, making most destinations appear one hop away. If you use the explicit Layer 2 transit layer, you can precisely control how traffic uses available bandwidth. However, the overlay model has numerous disadvantages. MPLS-TE achieves the TE benefits of the overlay model without running a separate network and without a non-scalable, full mesh of router interconnects.

**How MPLS-TE Works**

MPLS-TE automatically establishes and maintains label switched paths (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 Interior Gateway Protocol (IGP).

MPLS-TE tunnels are calculated at the LSP headend router, 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 crossing the MPLS-TE backbone travels on a single LSP that connects the ingress point to the egress point. MPLS-TE is built on these mechanisms:

**Tunnel interfaces**

From a Layer 2 standpoint, an MPLS tunnel interface represents the headend of an LSP. It is configured with a set of resource requirements, such as bandwidth and media requirements, and priority. From a Layer 3 standpoint, an LSP tunnel interface is the headend of a unidirectional virtual link to the tunnel destination.

**MPLS-TE path calculation module**

This calculation module operates at the LSP headend. 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 TE extensions**

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

**MPLS-TE link management module**

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

**Link-state IGP (Intermediate System-to-Intermediate System [IS-IS] or Open Shortest Path First [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 shortest path first (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 at an ingress device) determines which traffic should go to which egress device, and steers that traffic into the tunnel from ingress to egress. A flow from an ingress device to an egress device might be so large that it cannot fit over a single link, so it cannot be carried by a single tunnel. In this case, multiple tunnels between a given ingress and egress can be configured, and the flow is distributed using load sharing among the tunnels.

Note
GRE over MPLS-TE tunnel is not supported. Hence, you cannot carry GRE traffic over an LSP established for MPLS-TE tunnel using RSVP-TE. This restriction also applies to SR-TE tunnels.

Related Topics
- Building MPLS-TE Topology, on page 21
- Creating an MPLS-TE Tunnel, on page 24
- Build MPLS-TE Topology and Tunnels: Example, on page 65

Protocol-Based CLI
Cisco IOS XR software provides a protocol-based command line interface. The CLI provides commands that can be used with the multiple IGP protocols supported by MPLS-TE.

Differentiated Services Traffic Engineering
MPLS Differentiated Services (Diff-Serv) Aware Traffic Engineering (DS-TE) is an extension of the regular MPLS-TE feature. Regular traffic engineering does not provide bandwidth guarantees to different traffic classes. A single bandwidth constraint is used in regular TE that is shared by all traffic. To support various classes of service (CoS), users can configure multiple bandwidth constraints. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

MPLS DS-TE provides the ability to configure multiple bandwidth constraints on an MPLS-enabled interface. Available bandwidths from all configured bandwidth constraints are advertised using IGP. TE tunnel is configured with bandwidth value and class-type requirements. Path calculation and admission control take the bandwidth and class-type into consideration. RSVP is used to signal the TE tunnel with bandwidth and class-type requirements.

MPLS DS-TE is deployed with either Russian Doll Model (RDM) or Maximum Allocation Model (MAM) for bandwidth calculations.

Cisco IOS XR software supports two DS-TE modes: Prestandard and IETF.

Related Topics
- Confirming DiffServ-TE Bandwidth
- Bandwidth Configuration (MAM): Example
- Bandwidth Configuration (RDM): Example

Prestandard DS-TE Mode
Prestandard DS-TE uses the Cisco proprietary mechanisms for RSVP signaling and IGP advertisements. This DS-TE mode does not interoperate with third-party vendor equipment. Note that prestandard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces.
Prestandard Diff-Serve TE mode supports a single bandwidth constraint model a Russian Doll Model (RDM) with two bandwidth pools: global-pool and sub-pool.

TE class map is not used with Prestandard DS-TE mode.

**Related Topics**
- Configuring a Prestandard DS-TE Tunnel, on page 31
- Configure IETF DS-TE Tunnels: Example, on page 66

**IETF DS-TE Mode**

IETF DS-TE mode uses IETF-defined extensions for RSVP and IGP. This mode interoperates with third-party vendor equipment.

IETF mode supports multiple bandwidth constraint models, including RDM and MAM, both with two bandwidth pools. In an IETF DS-TE network, identical bandwidth constraint models must be configured on all nodes.

TE class map is used with IETF DS-TE mode and must be configured the same way on all nodes in the network.

**Bandwidth Constraint Models**

IETF DS-TE mode provides support for the RDM and MAM bandwidth constraints models. Both models support up to two bandwidth pools.

Cisco IOS XR software provides global configuration for the switching between bandwidth constraint models. Both models can be configured on a single interface to preconfigure the bandwidth constraints before swapping to an alternate bandwidth constraint model.

---

**Note**

NSF is not guaranteed when you change the bandwidth constraint model or configuration information.

By default, RDM is the default bandwidth constraint model used in both pre-standard and IETF mode.

**Maximum Allocation Bandwidth Constraint Model**

The MAM constraint model has the following characteristics:

- Easy to use and intuitive.
- Isolation across class types.
- Simultaneously achieves isolation, bandwidth efficiency, and protection against QoS degradation.

**Related Topics**
- Configuring an IETF DS-TE Tunnel Using MAM, on page 35

**Russian Doll Bandwidth Constraint Model**

The RDM constraint model has these characteristics:

- Allows greater sharing of bandwidth among different class types.
- Ensures bandwidth efficiency simultaneously and protection against QoS degradation of all class types.
• Specifies that it is used in conjunction with preemption to simultaneously achieve isolation across class-types such that each class-type is guaranteed its share of bandwidth, bandwidth efficiency, and protection against QoS degradation of all class types.

**Note**

We recommend that RDM not be used in DS-TE environments in which the use of preemption is precluded. Although RDM ensures bandwidth efficiency and protection against QoS degradation of class types, it does not guarantee isolation across class types.

**Related Topics**

- Configuring an IETF DS-TE Tunnel Using RDM, on page 33

## TE Class Mapping

Each of the eight available bandwidth values advertised in the IGP corresponds to a TE class. Because the IGP advertises only eight bandwidth values, there can be a maximum of only eight TE classes supported in an IETF DS-TE network.

TE class mapping must be exactly the same on all routers in a DS-TE domain. It is the responsibility of the operator to configure these settings properly as there is no way to automatically check or enforce consistency.

The operator must configure TE tunnel class types and priority levels to form a valid TE class. When the TE class map configuration is changed, tunnels already up are brought down. Tunnels in the down state, can be set up if a valid TE class map is found.

The default TE class and attributes are listed. The default mapping includes four class types.

**Table 1: TE Classes and Priority**

<table>
<thead>
<tr>
<th>TE Class</th>
<th>Class Type</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Unused</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>Unused</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Unused</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>Unused</td>
<td>—</td>
</tr>
</tbody>
</table>
Flooding

Available bandwidth in all configured bandwidth pools is flooded on the network to calculate accurate constraint paths when a new TE tunnel is configured. Flooding uses IGP protocol extensions and mechanisms to determine when to flood the network with bandwidth.

Flooding Triggers

TE Link Management (TE-Link) notifies IGP for both global pool and sub-pool available bandwidth and maximum bandwidth to flood the network in these events:

- Periodic timer expires (this does not depend on bandwidth pool type).
- Tunnel origination node has out-of-date information for either available global pool or sub-pool bandwidth, causing tunnel admission failure at the midpoint.
- Consumed bandwidth crosses user-configured thresholds. The same threshold is used for both global pool and sub-pool. If one bandwidth crosses the threshold, both bandwidths are flooded.

Flooding Thresholds

Flooding frequently can burden a network because all routers must send out and process these updates. Infrequent flooding causes tunnel heads (tunnel-originating nodes) to have out-of-date information, causing tunnel admission to fail at the midpoints.

You can control the frequency of flooding by configuring a set of thresholds. When locked bandwidth (at one or more priority levels) crosses one of these thresholds, flooding is triggered.

Thresholds apply to a percentage of the maximum available bandwidth (the global pool), which is locked, and the percentage of maximum available guaranteed bandwidth (the sub-pool), which is locked. If, for one or more priority levels, either of these percentages crosses a threshold, flooding is triggered.

Note

Setting up a global pool TE tunnel can cause the locked bandwidth allocated to sub-pool tunnels to be reduced (and hence to cross a threshold). A sub-pool TE tunnel setup can similarly cause the locked bandwidth for global pool TE tunnels to cross a threshold. Thus, sub-pool TE and global pool TE tunnels can affect each other when flooding is triggered by thresholds.

Fast Reroute

Fast Reroute (FRR) provides link protection to LSPs enabling the traffic carried by LSPs that encounter a failed link to be rerouted around the failure. The reroute decision is controlled locally by the router connected to the failed link. The headend router on the tunnel is notified of the link failure through IGP or through RSVP. When it is notified of a link failure, the headend router attempts to establish a new LSP that bypasses the failure. This provides a path to reestablish links that fail, providing protection to data transfer.

FRR (link or node) is supported over sub-pool tunnels the same way as for regular TE tunnels. In particular, when link protection is activated for a given link, TE tunnels eligible for FRR are redirected into the protection LSP, regardless of whether they are sub-pool or global pool tunnels.
The ability to configure FRR on a per-LSP basis makes it possible to provide different levels of fast restoration to tunnels from different bandwidth pools.

You should be aware of these requirements for the backup tunnel path:

- Backup tunnel must not pass through the element it protects.
- Primary tunnel and a backup tunnel should intersect at least at two points (nodes) on the path: point of local repair (PLR) and merge point (MP). PLR is the headend of the backup tunnel, and MP is the tailend of the backup tunnel.

When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.

MPLS-TE and Fast Reroute over Link Bundles

These link bundle types are supported for MPLS-TE/FRR:

- Over Ethernet link bundles.
- Over VLANs over Ethernet link bundles.
- Number of links are limited to 100 for MPLS-TE and FRR.
- VLANs go over any Ethernet interface (for example, ).

FRR is supported over bundle interfaces in the following ways:

- Uses minimum links as a threshold to trigger FRR over a bundle interface.
- Uses the minimum total available bandwidth as a threshold to trigger FRR.

Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE

The Ignore Intermediate System-to-Intermediate System (IS-IS) Overload Bit Setting in MPLS-TE feature ensures that the RSVP-TE LSPs are not broken because of routers that enabled the IS-IS overload bit.

The current implementation does not allow nodes that have indicated an overload situation through the IS-IS overload bit.
Therefore, an overloaded node cannot be used. The IS-IS overload bit limitation is an indication of an overload situation in the IP topology. The feature provides a method to prevent an IS-IS overload condition from affecting MPLS-TE.

**Enhancement Options of IS-IS OLA**

**Related Topics**
- Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE, on page 38
- Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 67

**Flexible Name-based Tunnel Constraints**

MPLS-TE Flexible Name-based Tunnel Constraints provides a simplified and more flexible means of configuring link attributes and path affinities to compute paths for MPLS-TE tunnels.

In the traditional TE scheme, links are configured with attribute-flags that are flooded with TE link-state parameters using Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF).

MPLS-TE Flexible Name-based Tunnel Constraints lets you assign, or map, up to 32 color names for affinity and attribute-flag attributes instead of 32-bit hexadecimal numbers. After mappings are defined, the attributes can be referred to by the corresponding color name in the command-line interface (CLI). Furthermore, you can define constraints using `include`, `include-strict`, `exclude`, and `exclude-all` arguments, where each statement can contain up to 10 colors, and define include constraints in both loose and strict sense.

**Note**
You can configure affinity constraints using attribute flags or the Flexible Name Based Tunnel Constraints scheme; however, when configurations for both schemes exist, only the configuration pertaining to the new scheme is applied.

**Related Topics**
- Assigning Color Names to Numeric Values, on page 39
- Associating Affinity-Names with TE Links, on page 40
- Associating Affinity Constraints for TE Tunnels, on page 41
- Configure Flexible Name-based Tunnel Constraints: Example, on page 67

**MPLS Traffic Engineering Interarea Tunneling**

These topics describe the following new extensions of MPLS-TE:

- Interarea Support, on page 10
- Multiarea Support, on page 10
- Loose Hop Expansion, on page 11
- Loose Hop Reoptimization, on page 11
- Fast Reroute Node Protection, on page 12
Interarea Support

The MPLS-TE interarea tunneling feature allows you to establish P2P tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thereby eliminating the requirement that headend and tailend routers reside in a single area.

Interarea support allows the configuration of a TE LSP that spans multiple areas, where its headend and tailend label switched routers (LSRs) reside in different IGP areas.

Multiarea and Interarea TE are required by the customers running multiple IGP area backbones (primarily for scalability reasons). This lets you limit the amount of flooded information, reduces the SPF duration, and lessens the impact of a link or node failure within an area, particularly with large WAN backbones split in multiple areas.

*Figure 1: Interarea (OSPF) TE Network Diagram*

This figure shows a typical interarea TE network.

Multiarea Support

Multiarea support allows an area border router (ABR) LSR to support MPLS-TE in more than one IGP area. A TE LSP is still confined to a single area.

Multiarea and Interarea TE are required when you run multiple IGP area backbones. The Multiarea and Interarea TE allows you to:

- Limit the volume of flooded information.
- Reduce the SPF duration.
- Decrease the impact of a link or node failure within an area.

*Figure 2: Interlevel (IS-IS) TE Network*
As shown in the figure, R2, R3, R7, and R4 maintain two databases for routing and TE information. For example, R3 has TE topology information related to R2, flooded through Level-1 IS-IS LSPs plus the TE topology information related to R4, R9, and R7, flooded as Level 2 IS-IS Link State PDUs (LSPs) (plus, its own IS-IS LSP).

**Note**
You can configure multiple areas within an IS-IS Level 1. This is transparent to TE. TE has topology information about the IS-IS level, but not the area ID.

**Loose Hop Expansion**

Loose hop optimization allows the reoptimization of tunnels spanning multiple areas and solves the problem which occurs when an MPLS-TE LSP traverses hops that are not in the LSP's headend's OSPF area and IS-IS level.

Interarea MPLS-TE allows you to configure an interarea traffic engineering (TE) label switched path (LSP) by specifying a loose source route of ABRs along the path. It is the then the responsibility of the ABR (having a complete view of both areas) to find a path obeying the TE LSP constraints within the next area to reach the next hop ABR (as specified on the headend). The same operation is performed by the last ABR connected to the tailend area to reach the tailend LSR.

You must be aware of these considerations when using loose hop optimization:

- You must specify the router ID of the ABR node (as opposed to a link address on the ABR).
- When multiarea is deployed in a network that contains subareas, you must enable MPLS-TE in the subarea for TE to find a path when loose hop is specified.
- You must specify the reachable explicit path for the interarea tunnel.

**Loose Hop Reoptimization**

Loose hop reoptimization allows the reoptimization of the tunnels spanning multiple areas and solves the problem which occurs when an MPLS-TE headend does not have visibility into other IGP areas.

Whenever the headend attempts to reoptimize a tunnel, it tries to find a better path to the ABR in the headend area. If a better path is found then the headend initiates the setup of a new LSP. In case a suitable path is not found in the headend area, the headend initiates a querying message. The purpose of this message is to query the ABRs in the areas other than the headend area to check if there exist any better paths in those areas. The purpose of this message is to query the ABRs in the areas other than the headend area, to check if a better path exists. If a better path does not exist, ABR forwards the query to the next router downstream. Alternatively, if better path is found, ABR responds with a special Path Error to the headend to indicate the existence of a better path outside the headend area. Upon receiving the Path Error that indicates the existence of a better path, the headend router initiates the reoptimization.

**ABR Node Protection**

Because one IGP area does not have visibility into another IGP area, it is not possible to assign backup to protect ABR node. To overcome this problem, node ID sub-object is added into the record route object of the primary tunnel so that at a PLR node, backup destination address can be checked against primary tunnel record-route object and assign a backup tunnel.
Fast Reroute Node Protection

If a link failure occurs within an area, the upstream router directly connected to the failed link generates an RSVP path error message to the headend. As a response to the message, the headend sends an RSVP path tear message and the corresponding path option is marked as invalid for a specified period and the next path-option (if any) is evaluated.

To retry the ABR immediately, a second path option (identical to the first one) should be configured. Alternatively, the retry period (path-option hold-down, 2 minutes by default) can be tuned to achieve a faster retry.

Related Topics
- Protecting MPLS Tunnels with Fast Reroute, on page 28

MPLS-TE Forwarding Adjacency

The MPLS-TE Forwarding Adjacency feature allows a network administrator to handle a traffic engineering, label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network based on the Shortest Path First (SPF) algorithm. A forwarding adjacency can be created between routers regardless of their location in the network.

MPLS-TE Forwarding Adjacency Benefits

TE tunnel interfaces are advertised in the IGP network just like any other links. Routers can then use these advertisements in their IGPs to compute the SPF even if they are not the head end of any TE tunnels.

Related Topics
- Configuring MPLS-TE Forwarding Adjacency, on page 45
- Configure Forwarding Adjacency: Example, on page 69

MPLS-TE Forwarding Adjacency Restrictions

The MPLS-TE Forwarding Adjacency feature has these restrictions:

- Using the MPLS-TE Forwarding Adjacency increases the size of the IGP database by advertising a TE tunnel as a link.
- The MPLS-TE Forwarding Adjacency is supported by Intermediate System-to-Intermediate System (IS-IS).
- When the MPLS-TE Forwarding Adjacency is enabled on a TE tunnel, the link is advertised in the IGP network as a Type-Length-Value (TLV) 22 without any TE sub-TLV.
- MPLS-TE forwarding adjacency tunnels must be configured bidirectionally.
- Multicast intact is not supported with MPLS-TE Forwarding Adjacency.

MPLS-TE Forwarding Adjacency Prerequisites

Your network must support the following features before enabling the MPLS-TE Forwarding Adjacency feature:

- MPLS
- IP Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS)

**Path Computation Element**

Path Computation Element (PCE) solves the specific issue of inter-domain path computation for MPLS-TE label switched path (LSPs), when the head-end router does not possess full network topology information (for example, when the head-end and tail-end routers of an LSP reside in different IGP areas).

PCE uses area border routers (ABRs) to compute a TE LSP spanning multiple IGP areas as well as computation of Inter-AS TE LSP.

PCE is usually used to define an overall architecture, which is made of several components, as follows:

**Path Computation Element (PCE)**

Represents a software module (which can be a component or application) that enables the router to compute paths applying a set of constraints between any pair of nodes within the router’s TE topology database. PCEs are discovered through IGP.

**Path Computation Client (PCC)**

Represents a software module running on a router that is capable of sending and receiving path computation requests and responses to and from PCEs. The PCC is typically an LSR (Label Switching Router).

**PCC-PCE communication protocol (PCEP)**

Specifies that PCEP is a TCP-based protocol defined by the IETF PCE WG, and defines a set of messages and objects used to manage PCEP sessions and to request and send paths for multi-domain TE LSPs. PCEP is used for communication between PCC and PCE (as well as between two PCEs) and employs IGP extensions to dynamically discover PCE.

*Figure 3: Path Computation Element Network Diagram*

This figure shows a typical PCE implementation.

Path computation elements provide support for the following message types and objects:

- **Message types**: Open, PCReq, PCRep, PCErr, Close
Policy-Based Tunnel Selection

These topics provide information about policy-based tunnel selection (PBTS):

Policy-Based Tunnel Selection

Policy-Based Tunnel Selection (PBTS) provides a mechanism that lets you direct traffic into specific TE tunnels based on different criteria. PBTS will benefit Internet service providers (ISPs) who carry voice and data traffic through their MPLS and MPLS/VPN networks, who want to route this traffic to provide optimized voice service.

PBTS works by selecting tunnels based on the classification criteria of the incoming packets, which are based on the IP precedence, experimental (EXP), or type of service (ToS) field in the packet.

Figure 4: Policy-Based Tunnel Selection Implementation

This figure illustrates a PBTS implementation.

PBTS is supported on the ingress interface and any of the L3 interfaces (physical, sub-interface, and bundle interface).

PBTS supports modification of the class-map and forward-group to TE association.

Related Topics

Configuring Policy-based Tunnel Selection, on page 50

Policy-Based Tunnel Selection Functions

The following PBTS functions are supported:

- IPv4 traffic arrives unlabeled on the VRF interface and the non-VRF interface.
- MPLS traffic is supported on the VRF interface and the non-VRF interface.
• Load balancing across multiple TE tunnels with the same traffic class attribute is supported.
• Selected TE tunnels are used to service the lowest tunnel class as default tunnels.
• LDP over TE tunnel and single-hop TE tunnel are supported.
• Both Interior Gateway Protocol (IGP) and Label Distribution Protocol (LDP) paths are used as the default path for all traffic that belongs to a class that is not configured on the TE tunnels.
• According to the quality-of-service (QoS) policy, tunnel selection is based on the outgoing experimental (EXP) value and the remarked EXP value.

Related Topics
Configuring Policy-based Tunnel Selection, on page 50

PBTS Restrictions
When implementing PBTS, the following restrictions are listed:
• When QoS EXP remarking on an interface is enabled, the EXP value is used to determine the egress tunnel interface, not the incoming EXP value.
• Egress-side remarking does not affect PBTS tunnel selection.
• When no default tunnel is available for forwarding, traffic is dropped.

MPLS-TE Automatic Bandwidth
The MPLS-TE automatic bandwidth feature measures the traffic in a tunnel and periodically adjusts the signaled bandwidth for the tunnel.

These topics provide information about MPLS-TE automatic bandwidth:

MPLS-TE Automatic Bandwidth Overview
MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every head-end. MPLS-TE monitors the traffic rate on a tunnel interface. Periodically, MPLS-TE resizes the bandwidth on the tunnel interface to align it closely with the traffic in the tunnel. MPLS-TE automatic bandwidth can perform these functions:
• Monitors periodic polling of the tunnel output rate
• Resizes the tunnel bandwidth by adjusting the highest rate observed during a given period

For every traffic-engineered tunnel that is configured for an automatic bandwidth, the average output rate is sampled, based on various configurable parameters. Then, the tunnel bandwidth is readjusted automatically based upon either the largest average output rate that was noticed during a certain interval, or a configured maximum bandwidth value.
This table lists the automatic bandwidth functions.

Table 2: Automatic Bandwidth Variables

<table>
<thead>
<tr>
<th>Function</th>
<th>Command</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application frequency</td>
<td>application</td>
<td>Configures how often the tunnel bandwidths changed for each tunnel. The</td>
<td>24 hours</td>
</tr>
<tr>
<td></td>
<td>command</td>
<td>application period is the period of A minutes between the bandwidth</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>applications during which the output rate collection is done.</td>
<td></td>
</tr>
<tr>
<td>Requested bandwidth</td>
<td>bw-limit</td>
<td>Limits the range of bandwidth within the automatic-bandwidth feature that</td>
<td>0 Kbps</td>
</tr>
<tr>
<td></td>
<td>command</td>
<td>can request a bandwidth.</td>
<td></td>
</tr>
<tr>
<td>Collection frequency</td>
<td>auto-bw collect</td>
<td>Configures how often the tunnel output rate is polled globally for all</td>
<td>5 min</td>
</tr>
<tr>
<td></td>
<td>command</td>
<td>tunnels.</td>
<td></td>
</tr>
<tr>
<td>Highest collected</td>
<td>—</td>
<td>You cannot configure this value.</td>
<td>—</td>
</tr>
<tr>
<td>bandwidth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delta</td>
<td>—</td>
<td>You cannot configure this value.</td>
<td>—</td>
</tr>
</tbody>
</table>

The output rate on a tunnel is collected at regular intervals that are configured by using the `application` command in MPLS-TE auto bandwidth interface configuration mode. When the application period timer expires, and when the difference between the measured and the current bandwidth exceeds the adjustment threshold, the tunnel is reoptimized. Then, the bandwidth samples are cleared to record the new largest output rate at the next interval.

When reoptimizing the LSP with the new bandwidth, a new path request is generated. If the new bandwidth is not available, the last good LSP continues to be used. This way, the network experiences no traffic interruptions.

If minimum or maximum bandwidth values are configured for a tunnel, the bandwidth, which the automatic bandwidth signals, stays within these values.

---

**Note**

When more than 100 tunnels are `auto-bw` enabled, the algorithm will jitter the first application of every tunnel by a maximum of 20% (max 1 hour). The algorithm does this to avoid too many tunnels running auto bandwidth applications at the same time.

If a tunnel is shut down, and is later brought again, the adjusted bandwidth is lost and the tunnel is brought back with the initial configured bandwidth. In addition, the application period is reset when the tunnel is brought back.

**Related Topics**

- Configuring the Collection Frequency, on page 52
- Configuring the Automatic Bandwidth Functions, on page 53
- Configure Automatic Bandwidth: Example, on page 71
Adjustment Threshold

Adjustment Threshold is defined as a percentage of the current tunnel bandwidth and an absolute (minimum) bandwidth. Both thresholds must be fulfilled for the automatic bandwidth to resignal the tunnel. The tunnel bandwidth is resized only if the difference between the largest sample output rate and the current tunnel bandwidth is larger than the adjustment thresholds.

For example, assume that the automatic bandwidth is enabled on a tunnel in which the highest observed bandwidth $B$ is 30 Mbps. Also, assume that the tunnel was initially configured for 45 Mbps. Therefore, the difference is 15 Mbit/s. Now, assuming the default adjustment thresholds of 10% and 10kbps, the tunnel is signalled with 30 Mbps when the application timer expires. This is because 10% of 45 Mbit/s is 4.5 Mbit/s, which is smaller than 15 Mbit/s. The absolute threshold, which by default is 10kbps, is also crossed.

Overflow Detection

Overflow detection is used if a bandwidth must be resized as soon as an overflow condition is detected, without having to wait for the expiry of an automatic bandwidth application frequency interval.

For overflow detection one configures a limit $N$, a percentage threshold $Y\%$ and optionally, a minimum bandwidth threshold $Z$. The percentage threshold is defined as the percentage of the actual signalled tunnel bandwidth. When the difference between the measured bandwidth and the actual bandwidth are both larger than $Y\%$ and $Z$ threshold, for $N$ consecutive times, then the system triggers an overflow detection.

The bandwidth adjustment by the overflow detection is triggered only by an increase of traffic volume through the tunnel, and not by a decrease in the traffic volume. When you trigger an overflow detection, the automatic bandwidth application interval is reset.

By default, the overflow detection is disabled and needs to be manually configured.

Underflow Detection

Underflow detection is used when the bandwidth on a tunnel drops significantly, which is similar to overflow but in reverse.

Underflow detection applies the highest bandwidth value from the samples which triggered the underflow. For example, if you have an underflow limit of three, and the following samples trigger the underflow for 10 kbps, 20 kbps, and 15 kbps, then, 20 kbps is applied.

Unlike overflow, the underflow count is not reset across an application period. For example, with an underflow limit of three, you can have the first two samples taken at the end of an application period and then the underflow gets triggered by the first sample of the next application period.

Restrictions for MPLS-TE Automatic Bandwidth

When the automatic bandwidth cannot update the tunnel bandwidth, the following restrictions are listed:

- Tunnel is in a fast reroute (FRR) backup, active, or path protect active state. This occurs because of the assumption that protection is a temporary state, and there is no need to reserve the bandwidth on a backup tunnel. You should prevent taking away the bandwidth from other primary or backup tunnels.

- Reoptimization fails to occur during a lockdown. In this case, the automatic bandwidth does not update the bandwidth unless the bandwidth application is manually triggered by using the `mpls traffic-eng auto-bw apply` command in EXEC mode.
Point-to-Multipoint Traffic-Engineering

Point-to-Multipoint Traffic-Engineering Overview

The Point-to-Multipoint (P2MP) Resource Reservation Protocol-Traffic Engineering (RSVP-TE) solution allows service providers to implement IP multicast applications, such as IPTV and real-time video, broadcast over the MPLS label switch network. The RSVP-TE protocol is extended to signal point-to-point (P2P) and P2MP label switched paths (LSPs) across the MPLS networks.

By using RSVP-TE extensions as defined in RFC 4875, multiple subLSPs are signaled for a given TE source. The P2MP tunnel is considered as a set of Source-to-Leaf (S2L) subLSPs that connect the TE source to multiple leaf Provider Edge (PE) nodes.

At the TE source, the ingress point of the P2MP-TE tunnel, IP multicast traffic is encapsulated with a unique MPLS label, which is associated with the P2MP-TE tunnel. The traffic continues to be label-switched in the P2MP tree. If needed, the labeled packet is replicated at branch nodes along the P2MP tree. When the labeled packet reaches the egress leaf (PE) node, the MPLS label is removed and forwarded onto the IP multicast tree across the PE-CE link.

To enable end-to-end IP multicast connectivity, RSVP is used in the MPLS-core for P2MP-TE signaling and PIM is used for PE-CE link signaling.

- All edge routers are running PIM-SSM or Source-Specific Multicast (SSM) to exchange multicast routing information with the directly-connected Customer Edge (CE) routers.
- In the MPLS network, RSVP P2MP-TE replaces PIM as the tree building mechanism, RSVP-TE grafts or prunes a given P2MP tree when the end-points are added or removed in the TE source configuration (explicit user operation).

These are the definitions for Point-to-Multipoint (P2MP) tunnels:

**Source**

Configures the node in which Label Switched Path (LSP) signaling is initiated.

**Mid-point**

Specifies the transit node in which LSP signaling is processed (for example, not a source or receiver).

**Receiver, Leaf, and Destination**

Specifies the node in which LSP signaling ends.

**Branch Point**

Specifies the node in which packet replication is performed.

**Source-to-Leaf (S2L) SubLSP**

Specifies the P2MP-TE LSP segment that runs from the source to one leaf.

---

**Note**

Cisco NCS 6000 Series Routers supports only P2MP TE mid-point functionality. The MPLS and the multicast packages are required the mid point router for the P2MP TE feature to work.
Point-to-Multipoint Traffic-Engineering Features

- P2MP RSVP-TE (RFC 4875) is supported. RFC 4875 is based on nonaggregate signaling; for example, per S2L signaling. Only P2MP LSP is supported.
- `interface tunnel-mte` command identifies the P2MP interface type on the Head-end.
- P2MP tunnel setup is supported with label replication.
- Fast-Reroute (FRR) protection is supported with sub-50 msec for traffic loss.
- Explicit routing is supported by using under utilized links.
- Reoptimization is supported by calculating a better set of paths to the destination with no traffic loss.

Note  
Per-S2L reoptimization is not supported.

- IPv4 and IPv6 payloads are supported.
- IPv4 and IPv6 multicast forwarding are supported on a P2MP tunnel interface through a static IGMP and MLD group configuration on the Head-end.
- Both IP multicast and P2MP Label Switch Multicast (LSM) coexist in the same network; therefore, both use the same forwarding plane (LFIB or MPLS Forwarding Infrastructure [MFI]).
- P2MP label replication supports only Source-Specific Multicast (SSM) traffic. SSM configuration supports the default value, none.
- Static mapping for multicast groups to the P2MP-TE tunnel is required on the Head-end.

Point-to-Multipoint Traffic-Engineering Benefits

- Single point of traffic control ensures that signaling and path engineering parameters (for example, protection and diversity) are configured only at the TE source node.
- Ability to configure explicit paths to enable optimized traffic distribution and prevention of single point of failures in the network.
- Link protection of MPLS-labeled traffic traversing branch paths of the P2MP-TE tree.
- Ability to do bandwidth Admission Control (AC) during set up and signaling of P2MP-TE paths in the MPLS network.

Related Topics

Point-to-Multipoint RSVP-TE, on page 19

Point-to-Multipoint RSVP-TE

RSVP-TE signals a P2MP tunnel base that is based on a manual configuration. If all Source-to-Leaf (S2L)s use an explicit path, the P2MP tunnel creates a static tree that follows a predefined path based on a constraint such as a deterministic Label Switched Path (LSP). If the S2L uses a dynamic path, RSVP-TE creates a P2MP tunnel base on the best path in the RSVP-TE topology. RSVP-TE supports bandwidth reservation for constraint-based routing.
When an explicit path option is used, specify both the local and peer IP addresses in the explicit path option, provided the link is a GigabitEthernet or a TenGigE based interface. For point-to-point links like POS or bundle POS, it is sufficient to mention the remote or peer IP address in the explicit path option.

RSVP-TE distributes stream information in which the topology tree does not change often (where the source and receivers are). For example, large scale video distribution between major sites is suitable for a subset of multicast applications. Because multicast traffic is already in the tunnel, the RSVP-TE tree is protected as long as you build a backup path.

Fast-Reroute (FRR) capability is supported for P2MP RSVP-TE by using the unicast link protection. You can choose the type of traffic to go to the backup link.

The P2MP tunnel is applicable for all TE Tunnel destination (IntraArea and InterArea). Inter-AS is not supported.

The P2MP tunnel is signaled by the dynamic and explicit path option in the IGP intra area. Only interArea and interAS, which are used for the P2MP tunnels, are signaled by the verbatim path option.

Related Topics
- Point-to-Multipoint Fast Reroute, on page 20

**Point-to-Multipoint Fast Reroute**

MPLS-TE Fast Reroute (FRR) is a mechanism to minimize interruption in traffic delivery to a TE Label Switched Path (LSP) destination as a result of link failures. FRR enables temporarily fast switching of LSP traffic along an alternative backup path around a network failure, until the TE tunnel source signals a new end-to-end LSP.

Both Point-to-Point (P2P) and P2MP-TE support only the Facility FRR method from RFC 4090.

P2P LSPs are used to backup P2MP S2L (source 2 Leaf). Only link and bandwidth protection for P2MP S2Ls are supported. Node protection is not supported.

MPLS-TE link protection relies on the fact that labels for all primary LSPs and subLSPs are using the MPLS global label allocation. For example, one single (global) label space is used for all MPLS-TE enabled physical interfaces on a given MPLS LSP.

Related Topics
- Point-to-Multipoint Traffic-Engineering Overview, on page 18
- Point-to-Multipoint RSVP-TE, on page 19

**Point-to-Multipoint Label Switch Path**

The Point-to-Multipoint Label Switch Path (P2MP LSP) has only a single root, which is the Ingress Label Switch Router (LSR). The P2MP LSP is created based on a receiver that is connected to the Egress LSR. The Egress LSR initiates the creation of the tree (for example, tunnel grafting or pruning is done by performing an individual sub-LSP operation) by creating the Forwarding Equivalency Class (FEC) and Opaque Value.

Grafting and pruning operate on a per destination basis.

The Opaque Value contains the stream information that uniquely identifies the tree to the root. To receive label switched multicast packets, the Egress Provider Edge (PE) indicates to the upstream router (the next hop closest to the root) which label it uses for the multicast source by applying the label mapping message.
The upstream router does not need to have any knowledge of the source; it needs only the received FEC to identify the correct P2MP LSP. If the upstream router does not have any FEC state, it creates it and installs the assigned downstream outgoing label into the label forwarding table. If the upstream router is not the root of the tree, it must forward the label mapping message to the next hop upstream. This process is repeated hop-by-hop until the root is reached.

By using downstream allocation, the router that wants to receive the multicast traffic assigns the label for it. The label request, which is sent to the upstream router, is similar to an unsolicited label mapping (that is, the upstream does not request it). The upstream router that receives that label mapping uses the specific label to send multicast packets downstream to the receiver. The advantage is that the router, which allocates the labels, does not get into a situation where it has the same label for two different multicast sources. This is because it manages its own label space allocation locally.

**Interarea P2MP Path Expansion within a Domain**

Interarea P2MP (Point-to-Multipoint) path expansion within a domain feature matches the domain of the subsequent auto-discovered ABR (Area Border Router) with the domain of the incoming interface where the Path message is received. This feature restricts the ERO (Explicit Route Object) expansion using the same domain as associated with the incoming interface where the Path message is received. This restriction applies to both loose-hop ABR and dynamically discovered ABR.

Configure this feature using the `path-selection loose-expansion domain-match` command in MPLS-TE configuration.

Interarea P2MP path expansion within a domain configuration applies to:

- All interarea TE (Traffic Engineering) path expansions on the ABR node
- Both P2P (Point-to-Point) and P2MP interarea TE LSPs
- Midpoint nodes

**Limitation**

The ERO expansion domain-match is not supported for multiple incoming IGPs.

**How to Implement Traffic Engineering**

Traffic engineering requires coordination among several global neighbor routers, creating traffic engineering tunnels, setting up forwarding across traffic engineering tunnels, setting up FRR, and creating differential service.

These procedures are used to implement MPLS-TE:

**Building MPLS-TE Topology**

Perform this task to configure MPLS-TE topology (required for traffic engineering tunnel operations).

**Before you begin**

Before you start to build the MPLS-TE topology, you must have enabled:
- IGP such as OSPF or IS-IS for MPLS-TE.
- MPLS Label Distribution Protocol (LDP).
- RSVP on the port interface.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.
- If you are going to use nondefault holdtime or intervals, you must decide the values to which they are set.

**SUMMARY STEPS**

1. **configure**
2. **mpls traffic-eng**
3. **interface type interface-path-id**
4. **exit**
5. **exit**
6. **router ospf process-name**
7. **area area-id**
8. **exit**
9. **mpls traffic-eng router-id ip-address**
10. **commit**
11. (Optional) **show mpls traffic-eng topology**
12. (Optional) **show mpls traffic-eng link-management advertisements**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
<td>Enables traffic engineering on a particular interface on the originating node and enters MPLS-TE interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>mpls traffic-eng</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>interface type interface-path-id</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#interface POS0/6/0/0</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if)#</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exits the current configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 5**

**Example:**

```
RP/0/RP0/CPU0:router(config-mpls-te-if)# exit
RP/0/RP0/CPU0:router(config-mpls-te)#
```

**Step 6**

**Purpose:**

Enters a name for the OSPF process.

```
router ospf process-name
```

**Example:**

```
RP/0/RP0/CPU0:router(config)# router ospf 1
```

**Step 7**

**Purpose:**

Configures an area for the OSPF process.

- Backbone areas have an area ID of 0.
- Non-backbone areas have a non-zero area ID.

```
area area-id
```

**Example:**

```
RP/0/RP0/CPU0:router(config-router)# area 0
```

**Step 8**

**Purpose:**

Exits the current configuration mode.

```
exit
```

**Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar)# exit
RP/0/RP0/CPU0:router(config-ospf)#
```

**Step 9**

**Purpose:**

Sets the MPLS-TE loopback interface.

```
mpls traffic-eng router-id ip-address
```

**Example:**

```
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1
```

**Step 10**

**Purpose:**

```
commit
```

**Step 11**

**Purpose:**

Verifies the traffic engineering topology.

```
(Optional) show mpls traffic-eng topology
```

**Example:**

```
RP/0/RP0/CPU0:router# show mpls traffic-eng topology
```

**Step 12**

**Purpose:**

Displays all the link-management advertisements for the links on this node.

```
(Optional) show mpls traffic-eng link-management advertisements
```

**Example:**
Creating an MPLS-TE Tunnel

Creating an MPLS-TE tunnel is a process of customizing the traffic engineering to fit your network topology. Perform this task to create an MPLS-TE tunnel after you have built the traffic engineering topology.

Before you begin

The following prerequisites are required to create an MPLS-TE tunnel:

• You must have a router ID for the neighboring router.

• Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

• If you are going to use nondefault holdtime or intervals, you must decide the values to which they are set.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. destination ip-address
4. ipv4 unnumbered type interface-path-id
5. path-option preference - priority dynamic
6. signalled- bandwidth {bandwidth [class-type ct ] | sub-pool bandwidth}
7. commit
8. (Optional) show mpls traffic-eng tunnels
9. (Optional) show ipv4 interface brief
10. (Optional) show mpls traffic-eng link-management admission-control

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router# show mpls traffic-eng link-management advertisements
```

Related Topics

- How MPLS-TE Works, on page 3
- Build MPLS-TE Topology and Tunnels: Example, on page 65
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router# interface tunnel-te 1</td>
<td>Assigns a destination address on the new tunnel. The destination address is the remote node’s MPLS-TE router ID.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Sets the path option to dynamic and assigns the path ID.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Sets the CT0 bandwidth required on this interface. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7).</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Verifies that the tunnel is connected (in the UP state) and displays all configured TE tunnels.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Displays all TE tunnel interfaces.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Displays all the tunnels on this node.</td>
</tr>
</tbody>
</table>
### Configuring Forwarding over the MPLS-TE Tunnel

Perform this task to configure forwarding over the MPLS-TE tunnel created in the previous task. This task allows MPLS packets to be forwarded on the link between network neighbors.

#### Before you begin

The following prerequisites are required to configure forwarding over the MPLS-TE tunnel:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

#### SUMMARY STEPS

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `ipv4 unnumbered type interface-path-id`
4. `autoroute announce`
5. `exit`
6. `router static address-family ipv4 unicast prefix mask ip-address interface type`
7. `commit`
8. (Optional) `ping {ip-address | hostname}`
9. (Optional) `show mpls traffic-eng autoroute`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>interface tunnel-te tunnel-id</code></td>
<td>Enters MPLS-TE interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Related Topics
- **Overview of MPLS Traffic Engineering**, on page 2
- **Creating an MPLS-TE Tunnel**, on page 24
Protecting MPLS Tunnels with Fast Reroute

Perform this task to protect MPLS-TE tunnels, as created in the previous task.

Although this task is similar to the previous task, its importance makes it necessary to present as part of the tasks required for traffic engineering on Cisco IOS XR software.

Before you begin

The following prerequisites are required to protect MPLS-TE tunnels:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.
- You must first configure a primary tunnel.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. fast-reroute
4. exit
5. mpls traffic-eng
6. interface type interface-path-id
7. backup-path tunnel-te tunnel-number
8. exit
9. exit
10. interface tunnel-te tunnel-id
11. backup-bw {backup bandwidth | sub-pool {bandwidth | unlimited} | global-pool {bandwidth | unlimited}}
12. ipv4 unnumbered type interface-path-id
13. path-option preference-priority {explicit name explicit-path-name}
14. destination ip-address
15. commit
16. (Optional) show mpls traffic-eng tunnels backup
17. (Optional) show mpls traffic-eng tunnels protection frr
18. (Optional) show mpls traffic-eng fast-reroute database

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
</tbody>
</table>
| Step 2 | interface tunnel-te tunnel-id  
Example:  
RP/0/RP0/CPU0:router# interface tunnel-te 1 | Configures an MPLS-TE tunnel interface. |
|-------|---------------------------------------------------|---------------------------------|
| Step 3 | fast-reroute  
Example:  
RP/0/RP0/CPU0:router(config-if)# fast-reroute | Enables fast reroute. |
| Step 4 | exit  
Example:  
RP/0/RP0/CPU0:router(config-if)# exit | Exits the current configuration mode. |
| Step 5 | mpls traffic-eng  
Example:  
RP/0/RP0/CPU0:router(config)# mpls traffic-eng  
RP/0/RP0/CPU0:router(config-mpls-te)# | Enters MPLS-TE configuration mode. |
| Step 6 | interface type interface-path-id  
Example:  
RP/0/RP0/CPU0:router(config-mpls-te)# interface pos0/6/0/0  
RP/0/RP0/CPU0:router(config-mpls-te-if)# | Enables traffic engineering on a particular interface on the originating node. |
| Step 7 | backup-path tunnel-te tunnel-number  
Example:  
RP/0/RP0/CPU0:router(config-mpls-te-if)# backup-path tunnel-te 2 | Sets the backup path to the backup tunnel. |
| Step 8 | exit  
Example:  
RP/0/RP0/CPU0:router(config-mpls-te-if)# exit  
RP/0/RP0/CPU0:router(config-mpls-te)# | Exits the current configuration mode. |
| Step 9 | exit  
Example:  
RP/0/RP0/CPU0:router(config-mpls-te)# exit  
RP/0/RP0/CPU0:router(config-mpls-te)# | Exits the current configuration mode. |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 10**

  `interface tunnel-te tunnel-id`  
  **Example:**
  ```
  RP/0/RP0/CPU0:router(config)# interface tunnel-te 2
  ```

  Configures an MPLS-TE tunnel interface.

| **Step 11**

  `backup-bw {backup bandwidth | sub-pool {bandwidth | unlimited} | global-pool {bandwidth | unlimited} }`  
  **Example:**
  ```
  RP/0/RP0/CPU0:router(config)# backup-bw global-pool 5000
  ```

  Sets the CT0 bandwidth required on this interface.  
  **Note** Because the default tunnel priority is 7, tunnels use the default TE class map.

| **Step 12**

  `ipv4 unnumbered type interface-path-id`  
  **Example:**
  ```
  RP/0/RP0/CPU0:router(config)# ipv4 unnumbered Loopback0
  ```

  Assigns a source address to set up forwarding on the new tunnel.

| **Step 13**

  `path-option preference-priority {explicit name explicit-path-name}  
  **Example:**
  ```
  RP/0/RP0/CPU0:router(config)# path-option l explicit name backup-path
  ```

  Sets the path option to explicit with a given name (previously configured) and assigns the path ID.

| **Step 14**

  `destination ip-address`  
  **Example:**
  ```
  RP/0/RP0/CPU0:router(config)# destination 192.168.92.125
  ```

  Assigns a destination address on the new tunnel.  
  - Destination address is the remote node’s MPLS-TE router ID.  
  - Destination address is the merge point between backup and protected tunnels.  
  **Note** When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.

| **Step 15**

  `commit`  

| **Step 16**

  (Optional) `show mpls traffic-eng tunnels backup`  
  **Example:**

  Displays the backup tunnel information.
### Configuring a Prestandard DS-TE Tunnel

Perform this task to configure a Prestandard DS-TE tunnel.

#### Before you begin

The following prerequisites are required to configure a Prestandard DS-TE tunnel:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

#### SUMMARY STEPS

1. **configure**
2. **rsvp interface type interface-path-id**
3. **bandwidth [total reservable bandwidth] [bc0 bandwidth] [global-pool bandwidth] [sub-pool reservable-bw]**
4. **exit**
5. **exit**
6. **interface tunnel-te tunnel-id**
7. **signalled-bandwidth {bandwidth [class-type ct] | sub-pool bandwidth}**
## 8. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters RSVP configuration mode and selects an RSVP interface.</td>
</tr>
</tbody>
</table>
| **Step 2** rsvp interface *type interface-path-id*  
  **Example:**  
  RP/0/RP0/CPU0:router(config)# rsvp interface pos0/6/0/0 | Enters RSVP configuration mode and selects an RSVP interface. |
| **Step 3** bandwidth [*total reservable bandwidth*] [bc0 *bandwidth*]  
  [global-pool *bandwidth*] [sub-pool reservable-bw]  
  **Example:**  
  RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 100 150 sub-pool 50 | Sets the reserved RSVP bandwidth available on this interface by using the prestandard DS-TE mode. The range for the *total reserve bandwidth* argument is 0 to 4294967295. Physical interface bandwidth is not used by MPLS-TE. |
| **Step 4** exit  
  **Example:**  
  RP/0/RP0/CPU0:router(config-rsvp-if)# exit  
  RP/0/RP0/CPU0:router(config-rsvp)# | Exits the current configuration mode. |
| **Step 5** exit  
  **Example:**  
  RP/0/RP0/CPU0:router(config-rsvp)# exit  
  RP/0/RP0/CPU0:router(config)# | Exits the current configuration mode. |
| **Step 6** interface tunnel-te *tunnel-id*  
  **Example:**  
  RP/0/RP0/CPU0:router(config)# interface tunnel-te 2 | Configures an MPLS-TE tunnel interface. |
| **Step 7** signalled-bandwidth {*bandwidth* [class-type *ct*] [sub-pool *bandwidth*]}  
  **Example:**  
  RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth sub-pool 10 | Sets the bandwidth required on this interface. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7). |
| **Step 8** commit | |
Related Topics

- Configuring Traffic Engineering Tunnel Bandwidth
- Prestandard DS-TE Mode, on page 4
- Configure IETF DS-TE Tunnels: Example, on page 66

## Configuring an IETF DS-TE Tunnel Using RDM

Perform this task to create an IETF mode DS-TE tunnel using RDM.

### Before you begin

The following prerequisites are required to create an IETF mode DS-TE tunnel using RDM:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

### SUMMARY STEPS

1. configure
2. rsvp interface type interface-path-id
3. bandwidth rdm \{total-reservable-bw | bc0 | global-pool\} \{sub-pool | bc1 reservable-bw\}
4. exit
5. exit
6. mpls traffic-eng
7. ds-te mode ietf
8. exit
9. interface tunnel-te tunnel-id
10. signalled-bandwidth \{bandwidth [class-type ct] | sub-pool bandwidth\}
11. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
<td>Enters RSVP configuration mode and selects an RSVP interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>rsvp interface type interface-path-id</td>
<td>Sets the reserved RSVP bandwidth available on this interface by using the Russian Doll Model (RDM) bandwidth constraints model. The range for the total reserve bandwidth argument is 0 to 4294967295.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router(config)# rsvp interface pos0/6/0/0
```

Example:

```
RP/0/RP0/CPU0:router(config)# bandwidth rdm total-reservable-bw
```

Example:

```
RP/0/RP0/CPU0:router(config)# bandwidth rdm bc0
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth rdm 100 150</td>
<td></td>
<td>Physical interface bandwidth is not used by MPLS-TE.</td>
</tr>
</tbody>
</table>

Step 4  
exit  
Example:  
```bash  
RP/0/RP0/CPU0:router(config-rsvp-if)# exit  
RP/0/RP0/CPU0:router(config-rsvp)  
```

Exits the current configuration mode.

Step 5  
exit  
Example:  
```bash  
RP/0/RP0/CPU0:router(config-rsvp) exit  
RP/0/RP0/CPU0:router(config)  
```

Exits the current configuration mode.

Step 6  
mpls traffic-eng  
Example:  
```bash  
RP/0/RP0/CPU0:router(config)# mpls traffic-eng  
RP/0/RP0/CPU0:router(config-mpls-te)#  
```

Enters MPLS-TE configuration mode.

Step 7  
ds-te mode ietf  
Example:  
```bash  
RP/0/RP0/CPU0:router(config-mpls-te)# ds-te mode ietf  
```

Enables IETF DS-TE mode and default TE class map. IETF DS-TE mode is configured on all network nodes.

Step 8  
exit  
Example:  
```bash  
RP/0/RP0/CPU0:router(config-mpls-te)# exit  
```

Exits the current configuration mode.

Step 9  
interface tunnel-te tunnel-id  
Example:  
```bash  
RP/0/RP0/CPU0:router(config)# interface tunnel-te 4  
RP/0/RP0/CPU0:router(config-if)#  
```

Configures an MPLS-TE tunnel interface.

Step 10  
signalled-bandwidth {bandwidth [class-type ct] | sub-pool bandwidth}  
Example:  
```bash  
```

Configures the bandwidth required for an MPLS TE tunnel. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7).
### Configuring an IETF DS-TE Tunnel Using MAM

Perform this task to configure an IETF mode differentiated services traffic engineering tunnel using the Maximum Allocation Model (MAM) bandwidth constraint model.

**Before you begin**

The following prerequisites are required to configure an IETF mode differentiated services traffic engineering tunnel using the MAM bandwidth constraint model:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

### SUMMARY STEPS

1. configure
2. rsvp interface type interface-path-id
3. bandwidth mam {total reservable bandwidth | max-reservable-bw maximum-reservable-bw} [bc0 reservable bandwidth] [bc1 reservable bandwidth]
4. exit
5. exit
6. mpls traffic-eng
7. ds-te mode ietf
8. ds-te bc-model mam
9. exit
10. interface tunnel-te tunnel-id
11. signalled-bandwidth {bandwidth [class-type ct] | sub-pool bandwidth}
12. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters RSVP configuration mode and selects the RSVP interface.</td>
<td></td>
</tr>
<tr>
<td>rsvp interface type interface-path-id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# rsvp interface pos0/6/0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Sets the reserved RSVP bandwidth available on this interface.</td>
<td></td>
</tr>
<tr>
<td>bandwidth mam {total reservable bandwidth</td>
<td>max-reservable-bw maximum-reservable-bw} [bc0 reservations bandwidth] [bc1 reservations bandwidth]</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth mam max-reservable-bw 400 bc0 300 bc1 200</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Exits the current configuration mode.</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-rsvp-if)# exit RP/0/RP0/CPU0:router(config-rsvp)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits the current configuration mode.</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-rsvp)# exit RP/0/RP0/CPU0:router(config)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Enters MPLS-TE configuration mode.</td>
<td></td>
</tr>
<tr>
<td>mpls traffic-eng</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Enables IETF DS-TE mode and default TE class map. Configure IETF DS-TE mode on all nodes in the network.</td>
<td></td>
</tr>
<tr>
<td>ds-te mode ietf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# ds-te mode ietf</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Enables the MAM bandwidth constraint model globally.</td>
<td></td>
</tr>
<tr>
<td>ds-te bc-model mam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# ds-te bc-model mam</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action** | **Purpose**
--- | ---
**Exitsthecurrentconfigurationmode.** | **exit**

**Example:**

```
RP/0/RP0/CPU0:router(config-mpls-te)# exit
```

**ConfiguresanMPLS-TEtunnelinterface.**

**interface tunnel-te tunnel-id**

**Example:**

```
RP/0/RP0/CPU0:router(config)# interface tunnel-te 4
RP/0/RP0/CPU0:router(config-if)#
```

**Configures the bandwidth required for an MPLS TE tunnel.**

Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7).

**signalled-bandwidth ** `{bandwidth [class-type ct] | sub-pool bandwidth}`

**Example:**

```
RP/0/RP0/CPU0:router(config-rsvp-if)#
```

**Related Topics**

- Configuring Traffic Engineering Tunnel Bandwidth
- Maximum Allocation Bandwidth Constraint Model, on page 5

## Configuring MPLS -TE and Fast-Reroute on OSPF

Perform this task to configure MPLS-TE and Fast Reroute (FRR) on OSPF.

### Before you begin

**Note**

Only point-to-point (P2P) interfaces are supported for OSPF multiple adjacencies. These may be either native P2P interfaces or broadcast interfaces on which the OSPF P2P configuration command is applied to force them to behave as P2P interfaces as far as OSPF is concerned. This restriction does not apply to IS-IS.

The tunnel-te interface is not supported under IS-IS.

### SUMMARY STEPS

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `path-option [protecting ] preference-priority {dynamic [pce [address ipv4 address] | explicit [name pathname] [identifier path-number] ] [isis instance name [level level] ] [ospf instance name [area area ID] ] ] [verbatim] [lockdown]`
4. Repeat Step 3 as many times as needed.
5. commit
6. `show mpls traffic-eng tunnels [tunnel-number]`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>interface tunnel-te tunnel-id</code>&lt;br&gt;<code>Example:</code>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config)# interface tunnel-te 1&lt;br&gt;RP/0/RP0/CPU0:router(config-if)#</code></td>
<td>Configures an MPLS-TE tunnel interface. The range for the tunnel ID number is 0 to 65535.</td>
</tr>
<tr>
<td>3</td>
<td>`path-option [protecting] preference-priority {dynamic [pce [address ipv4 address]</td>
<td>explicit {name pathname</td>
</tr>
<tr>
<td>4</td>
<td>Repeat Step 3 as many times as needed.&lt;br&gt;<code>Example:</code>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config-if)# path-option 2 explicit name 234 ospf 3 area 7 verbatim</code></td>
<td>Configures another explicit path option.</td>
</tr>
<tr>
<td>5</td>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>show mpls traffic-eng tunnels [tunnel-number]</code>&lt;br&gt;<code>Example:</code>&lt;br&gt;<code>RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 1</code></td>
<td>Displays information about MPLS-TE tunnels.</td>
</tr>
</tbody>
</table>

### Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE

Perform this task to configure an overload node avoidance in MPLS-TE. When the overload bit is enabled, tunnels are brought down when the overload node is found in the tunnel path.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. path-selection ignore overload
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 2** mpls traffic-eng
Example:
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# | Enters MPLS-TE configuration mode. |
| **Step 3** path-selection ignore overload
Example:
RP/0/RP0/CPU0:router(config-mpls-te)# path-selection ignore overload | Ignores the Intermediate System-to-Intermediate System (IS-IS) overload bit setting for MPLS-TE. |
| **Step 4** commit | |

Related Topics

- Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE, on page 8
- Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 67

Configuring Flexible Name-based Tunnel Constraints

To fully configure MPLS-TE flexible name-based tunnel constraints, you must complete these high-level tasks in order:

1. Assigning Color Names to Numeric Values, on page 39
2. Associating Affinity-Names with TE Links, on page 40
3. Associating Affinity Constraints for TE Tunnels, on page 41

Assigning Color Names to Numeric Values

The first task in enabling the new coloring scheme is to assign a numerical value (in hexadecimal) to each value (color).
An affinity color name cannot exceed 64 characters. An affinity value cannot exceed a single digit. For example, magenta1.

### SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng`
3. `affinity-map affinity name {affinity value | bit-position value}`
4. `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
</tr>
<tr>
<td>Step 3 affinity-map affinity name {affinity value</td>
<td>bit-position value}</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# affinity-map</td>
</tr>
<tr>
<td></td>
<td>red 1</td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- Flexible Name-based Tunnel Constraints, on page 9
- Configure Flexible Name-based Tunnel Constraints: Example, on page 67

### Associating Affinity-Names with TE Links

The next step in the configuration of MPLS-TE Flexible Name-based Tunnel Constraints is to assign affinity names and values to TE links. You can assign up to a maximum of 32 colors. Before you assign a color to a link, you must define the name-to-value mapping for each color.

### SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng`
3. `interface type interface-path-id`
4. `attribute-names attribute name`
5. **commit**

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
</tbody>
</table>
| Step 2 | mpls traffic-eng  
**Example:**  
RP/0/RP0/CPU0:router(config)# mpls traffic-eng  
RP/0/RP0/CPU0:router(config-mpls-te)# | Enters MPLS-TE configuration mode. |
| Step 3 | interface type interface-path-id  
**Example:**  
RP/0/RP0/CPU0:router(config-mpls-te)# interface tunnel-te 2  
RP/0/RP0/CPU0:router(config-mpls-te-if)# | Enables MPLS-TE on an interface and enters MPLS-TE interface configuration mode. |
| Step 4 | attribute-names attribute name  
**Example:**  
RP/0/RP0/CPU0:router(config-mpls-te-if)# attribute-names red | Assigns colors to TE links over the selected interface. |
| Step 5 | commit | |

---

**Related Topics**

- Flexible Name-based Tunnel Constraints, on page 9  
- Configure Flexible Name-based Tunnel Constraints: Example, on page 67  
- Assigning Color Names to Numeric Values, on page 39

### Associating Affinity Constraints for TE Tunnels

The final step in the configuration of MPLS-TE Flexible Name-based Tunnel Constraints requires that you associate a tunnel with affinity constraints.

Using this model, there are no masks. Instead, there is support for four types of affinity constraints:

- include  
- include-strict  
- exclude  
- exclude-all
For the affinity constraints above, all but the exclude-all constraint may be associated with up to 10 colors.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. affinity {affinity-value  mask  mask-value  |  exclude  name  |  exclude -all  |  include  name  |  include-strict name}
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel-te tunnel-id</td>
<td>Configures link attributes for links comprising a tunnel. You can have up to ten colors.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 1</td>
<td>Multiple include statements can be specified under tunnel configuration. With this configuration, a link is eligible for CSPF if it has at least a red color or has at least a green color. Thus, a link with red and any other colors as well as a link with green and any additional colors meet the above constraint.</td>
</tr>
<tr>
<td><strong>Step 3</strong> affinity {affinity-value  mask  mask-value</td>
<td>exclude  name</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# affinity include red</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Flexible Name-based Tunnel Constraints, on page 9
Configure Flexible Name-based Tunnel Constraints: Example, on page 67

Configuring IS-IS to Flood MPLS-TE Link Information

Perform this task to configure a router running the Intermediate System-to-Intermediate System (IS-IS) protocol to flood MPLS-TE link information into multiple IS-IS levels.

This procedure shows how to enable MPLS-TE in both IS-IS Level 1 and Level 2.

SUMMARY STEPS

1. configure
2. `router isis instance-id`
3. `net network-entity-title`
4. `address-family {ipv4 | ipv6} {unicast}`
5. `metric-style wide`
6. `mpls traffic-eng level`
7. `commit`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters an IS-IS instance.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>router isis instance-id</code> <strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config)# <code>router isis 1</code></td>
<td>Enters an IS-IS instance.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>net network-entity-title</code> <strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-isis)# <code>net 47.0001.0000.0000.0002.00</code></td>
<td>Enters an IS-IS network entity title (NET) for the routing process.</td>
</tr>
<tr>
<td>Step 4</td>
<td>`address-family {ipv4</td>
<td>ipv6} {unicast}<code>**Example:**&lt;br&gt;RP/0/RP0/CPU0:router(config-isis)#</code>address-family ipv4 unicast`</td>
</tr>
<tr>
<td>Step 5</td>
<td><code>metric-style wide</code> <strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-isis-af)# <code>metric-style wide</code></td>
<td>Enters the new-style type, length, and value (TLV) objects.</td>
</tr>
<tr>
<td>Step 6</td>
<td><code>mpls traffic-eng level</code> <strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-isis-af)# <code>mpls traffic-eng level-1-2</code></td>
<td>Enters the required MPLS-TE level or levels.</td>
</tr>
<tr>
<td>Step 7</td>
<td><code>commit</code></td>
<td></td>
</tr>
</tbody>
</table>
Configuring an OSPF Area of MPLS-TE

Perform this task to configure an OSPF area for MPLS-TE in both the OSPF backbone area 0 and area 1.

SUMMARY STEPS

1. configure
2. router ospf process-name
3. mpls traffic-eng router-id ip-address
4. area area-id
5. interface type interface-path-id
6. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters a name that uniquely identifies an OSPF routing process.</td>
</tr>
<tr>
<td>Step 2 router ospf process-name</td>
<td>Any alphanumeric string no longer than 40 characters without spaces.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/Rf0/CPU0:router(config)# router ospf 100</td>
<td></td>
</tr>
<tr>
<td>Step 3 mpls traffic-eng router-id ip-address</td>
<td>Enters the MPLS interface type. For more information, use the question mark (?) online help function.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/Rf0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1</td>
<td></td>
</tr>
<tr>
<td>Step 4 area area-id</td>
<td>Enters an OSPF area identifier.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/Rf0/CPU0:router(config-ospf)# area 0</td>
<td>Either a decimal value or an IP address.</td>
</tr>
<tr>
<td>Step 5 interface type interface-path-id</td>
<td>Identifies an interface ID. For more information, use the question mark (?) online help function.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/Rf0/CPU0:router(config-ospf-ar)# interface POS 0/2/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 6 commit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Explicit Paths with ABRs Configured as Loose Addresses

Perform this task to specify an IPv4 explicit path with ABRs configured as loose addresses.
SUMMARY STEPS

1. configure
2. explicit-path name name
3. index index-id next-address [loose] ipv4 unicast ip-address
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 explicit-path name name</td>
<td>Enters a name for the explicit path.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# explicit-path name interarea1</td>
<td></td>
</tr>
<tr>
<td>Step 3 index index-id next-address [loose] ipv4 unicast ip-address</td>
<td>Includes an address in an IP explicit path of a tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-expl-path)# index 1 next-address loose ipv4 unicast 10.10.10.10</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring MPLS-TE Forwarding Adjacency

Perform this task to configure forwarding adjacency on a specific tunnel-te interface.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. forwarding-adjacency holdtime value
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface tunnel-te tunnel-id</td>
<td>Enters MPLS-TE interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te</td>
<td></td>
</tr>
</tbody>
</table>
Configuring a Path Computation Client and Element

Perform these tasks to configure Path Computation Client (PCC) and Path Computation Element (PCE):

- Configuring a Path Computation Client, on page 46
- Configuring a Path Computation Element Address, on page 47
- Configuring PCE Parameters, on page 48

Configuring a Path Computation Client

Perform this task to configure a TE tunnel as a PCC.

Note

Only one TE-enabled IGP instance can be used at a time.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. path-option preference-priority dynamic pce
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring a Path Computation Element Address

**Purpose**
Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.

**Command or Action**
`interface tunnel-te tunnel-id`

**Example:**
RP/0/RP0/CPU0:router(config)# interface tunnel-te 6

**Step 2**
Configures a TE tunnel as a PCC.

**Command or Action**
`path-option preference-priority dynamic pce`

**Example:**
RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic pce

**Step 3**
Commits the configuration.

**Command or Action**
`commit`

**Step 4**

**Related Topics**
- [Path Computation Element](#), on page 13
- [Configure PCE: Example](#), on page 69

---

### Configuring a Path Computation Element Address

**Summary Steeps**
1. `configure`
2. `mpls traffic-eng`
3. `pce address ipv4 address`
4. `commit`

**Detailed Steps**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><code>mpls traffic-eng</code></td>
<td>Enters the MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><code>pce address ipv4 address</code></td>
<td>Configures a PCE IPv4 address.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>

---

**Note**
Only one TE-enabled IGP instance can be used at a time.
Configuring PCE Parameters

Perform this task to configure PCE parameters, including a static PCE peer, periodic reoptimization timer values, and request timeout values.

**SUMMARY STEPS**

1. **configure**
2. **mpls traffic-eng**
3. **pce address ipv4** `address`
4. **pce peer ipv4** `address`
5. **pce keepalive interval**
6. **pce deadtimer value**
7. **pce reoptimize value**
8. **pce request-timeout value**
9. **pce tolerance keepalive value**
10. **commit**
11. **show mpls traffic-eng pce peer [address | all]**
12. **show mpls traffic-eng pce tunnels**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>mpls traffic-eng</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Enters MPLS-TE configuration mode.</strong></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>pce address ipv4</strong> <code>address</code></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Configures a PCE IPv4 address.</strong></td>
</tr>
</tbody>
</table>

**Related Topics**

- Path Computation Element, on page 13
- Configure PCE: Example, on page 69
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>pce peer ipv4 <em>address</em></td>
<td>Configures a static PCE peer address. PCE peers are also discovered dynamically through OSPF or ISIS.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce peer address ipv4 10.1.1.1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>pce keepalive <em>interval</em></td>
<td>Configures a PCEP keepalive interval. The range is from 0 to 255 seconds. When the keepalive interval is 0, the LSR does not send keepalive messages.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce keepalive 10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>pce deadtimer <em>value</em></td>
<td>Configures a PCE deadtimer value. The range is from 0 to 255 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce deadtimer 50</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>pce reoptimize <em>value</em></td>
<td>Configures a periodic reoptimization timer value. The range is from 60 to 604800 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce reoptimize 200</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>pce request-timeout <em>value</em></td>
<td>Configures a PCE request-timeout. Range is from 5 to 100 seconds. PCC or PCE keeps a pending path request only for the request-timeout period.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce request-timeout 10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>pce tolerance keepalive <em>value</em></td>
<td>Configures a PCE tolerance keepalive value (which is the minimum acceptable peer proposed keepalive).</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce tolerance keepalive 10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>show mpls traffic-eng pce peer [address</td>
<td>all]</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng pce peer</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Policy-based Tunnel Selection

Perform this task to configure policy-based tunnel selection (PBTS).

**SUMMARY STEPS**

1. configure
2. interface tunnel-te tunnel-id
3. ipv4 unnumbered type interface-path-id
4. signalled-bandwidth {bandwidth [class-type ct] | sub-pool bandwidth}
5. autoroute announce
6. destination ip-address
7. policy-class {1 - 7} | {default}
8. path-option preference-priority {explicit name explicit-path-name}
9. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface tunnel-te tunnel-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te 6</td>
</tr>
<tr>
<td></td>
<td>Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>ipv4 unnumbered type interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
</tr>
<tr>
<td></td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel.</td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

| Step 4 | `signalled-bandwidth {bandwidth [class-type ct] | sub-pool bandwidth}` | Configures the bandwidth required for an MPLS TE tunnel. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7). |
|--------|-------------------------------------------------|-----------------------------------------------------------------|
| Example: | | **Example:** |
| | | `RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10 class-type 1` |
| Step 5 | `autoroute announce` | Enables messages that notify the neighbor nodes about the routes that are forwarding. |
| Example: | | **Example:** |
| | | `RP/0/RP0/CPU0:router(config-if)# autoroute announce` |
| Step 6 | `destination ip-address` | Assigns a destination address on the new tunnel. |
| Example: | | **Example:** |
| | | `RP/0/RP0/CPU0:router(config-if)# destination 10.1.1.1` |
| Step 7 | `policy-class {1 - 7} | {default}` | Configures PBTS to direct traffic into specific TE tunnels or default class. |
| Example: | | **Example:** |
| | | `RP/0/RP0/CPU0:router(config-if)# policy-class 1` |
| Step 8 | `path-option preference-priority {explicit name explicit-path-name}` | Sets the path option to explicit with a given name (previously configured) and assigns the path ID. |
| Example: | | **Example:** |
| | | `RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-path` |
| Step 9 | `commit` | |

**Related Topics**

- Policy-Based Tunnel Selection Functions, on page 14
- Policy-Based Tunnel Selection, on page 14

### Configuring the Automatic Bandwidth

Perform these tasks to configure the automatic bandwidth:
Configuring the Collection Frequency

Perform this task to configure the collection frequency. You can configure only one global collection frequency.

SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. auto-bw collect frequency minutes
4. commit
5. show mpls traffic-eng tunnels [auto-bw]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 auto-bw collect frequency minutes</td>
<td>Configures the automatic bandwidth collection frequency, and controls the manner in which the bandwidth for a tunnel collects output rate information; but does not adjust the tunnel bandwidth.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# auto-bw collect frequency 1</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
<tr>
<td>Step 5 show mpls traffic-eng tunnels [auto-bw]</td>
<td>Displays information about MPLS-TE tunnels for the automatic bandwidth. The globally configured collection frequency is displayed.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls traffic tunnels auto-bw</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
- **MPLS-TE Automatic Bandwidth Overview**, on page 15
- **Configure Automatic Bandwidth: Example**, on page 71

Forcing the Current Application Period to Expire Immediately

Perform this task to force the current application period to expire immediately on the specified tunnel. The highest bandwidth is applied on the tunnel before waiting for the application period to end on its own.
### SUMMARY STEPS

1. `mpls traffic-eng auto-bw apply {all | tunnel-te tunnel-number}
2. `commit`
3. `show mpls traffic-eng tunnels [auto-bw]`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>`mpls traffic-eng auto-bw apply {all</td>
<td>tunnel-te tunnel-number}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# mpls traffic-eng auto-bw apply tunnel-te 1</code></td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** |
| `commit` | |

| **Step 3** |
| `show mpls traffic-eng tunnels [auto-bw]` | Displays information about MPLS-TE tunnels for the automatic bandwidth. |
| **Example:** |
| `RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-bw` | |

### Configuring the Automatic Bandwidth Functions

Perform this task to configure the following automatic bandwidth functions:

**Application frequency**

Configures the application frequency in which a tunnel bandwidth is updated by the automatic bandwidth.

**Bandwidth collection**

Configures only the bandwidth collection.

**Bandwidth parameters**

Configures the minimum and maximum automatic bandwidth to set on a tunnel.

**Adjustment threshold**

Configures the adjustment threshold for each tunnel.

**Overflow detection**

Configures the overflow detection for each tunnel.

### SUMMARY STEPS

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `auto-bw`
4. `application minutes`
5. `bw-limit {min bandwidth} {max bandwidth}`
6. `adjustment-threshold percentage [min minimum-bandwidth]`
7. `overflow threshold percentage [min bandwidth] limit limit`
8. `commit`
9. `show mpls traffic-eng tunnels [auto-bw]`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td><code>interface tunnel-te tunnel-id</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config)# interface tunnel-te 6</code></td>
<td><code>RP/0/RP0/CPU0:router(config-if)#</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures automatic bandwidth on a tunnel interface and enters MPLS-TE automatic bandwidth interface configuration mode.</td>
</tr>
<tr>
<td><code>auto-bw</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if)# auto-bw</code></td>
<td><code>RP/0/RP0/CPU0:router(config-if-tunte-autobw)#</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures the application frequency in minutes for the applicable tunnel.</td>
</tr>
<tr>
<td><code>application minutes</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if-tunte-autobw)# application 1000</code></td>
<td><strong>minutes</strong></td>
</tr>
<tr>
<td><code>frequency in minutes for the automatic bandwidth application. Range is from 5 to 10080 (7 days). The default value is 1440 (24 hours).</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Configures the minimum and maximum automatic bandwidth set on a tunnel.</td>
</tr>
<tr>
<td><code>bw-limit {min bandwidth} {max bandwidth}</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-if-tunte-autobw)# bw-limit min 30 max 80</code></td>
<td><strong>min</strong></td>
</tr>
<tr>
<td><code>Applies the minimum automatic bandwidth in kbps on a tunnel. Range is from 0 to 4294967295.</code></td>
<td></td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>Applies the maximum automatic bandwidth in kbps on a tunnel. Range is from 0 to 4294967295.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Specifies the tunnel bandwidth change threshold to trigger an adjustment.</td>
</tr>
</tbody>
</table>
| adjustment-threshold *percentage* [min minimum-bandwidth] | **percentage**
Bandwidth change percent threshold to trigger an adjustment if the largest sample percentage is higher or lower than the current tunnel bandwidth. Range is from 1 to 100 percent. The default value is 5 percent. |
| **min**
Configures the bandwidth change value to trigger an adjustment. The tunnel bandwidth is changed only if the largest sample is higher or lower than the current tunnel bandwidth. Range is from 10 to 4294967295 kilobits per second (kbps). The default value is 10 kbps. |
| Example: RP/0/RP0/CPU0:router(config-if-tunte-autobw)# adjustment-threshold 50 min 800 | |
| **Step 7** | Specifies the tunnel overflow detection. |
| overflow threshold *percentage* [min bandwidth] limit limit | **percentage**
Bandwidth change percent to trigger an overflow. Range is from 1 to 100 percent. |
| **limit**
Configures the number of consecutive collection intervals that exceed the threshold. The bandwidth overflow triggers an early tunnel bandwidth update. Range is from 1 to 10 collection periods. The default value is none. |
| **min**
Configures the bandwidth change value in kbps to trigger an overflow. Range is from 10 to 4294967295. The default value is 10. |
| Example: RP/0/RP0/CPU0:router(config-if-tunte-autobw)# overflow threshold 100 limit 1 | |
| **Step 8** | Specifies the number of consecutive collection intervals that exceed the threshold. |
| commit | Displays the MPLS-TE tunnel information only for tunnels in which the automatic bandwidth is enabled. |
| **Step 9** | Displays the MPLS-TE tunnel information only for tunnels in which the automatic bandwidth is enabled. |
| show mpls traffic-eng tunnels [auto-bw] | **Related Topics**
MPLS-TE Automatic Bandwidth Overview, on page 15
Configure Automatic Bandwidth: Example, on page 71 |
| Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-bw |
Configuring the Shared Risk Link Groups

To activate the MPLS traffic engineering SRLG feature, you must configure the SRLG value of each link that has a shared risk with another link.

Implementing Associated Bidirectional Label Switched Paths

This section describes how to configure MPLS Traffic Engineering Associated Bidirectional Label Switched Paths (MPLS-TE LSPs).

Associated Bidirectional Label Switched Paths are LSP instances where the forward and the reverse direction paths are setup, monitored and protected independently and associated together during signaling. You use a RSVP Association object to bind the two forward and reverse LSPs together to form either a co-routed or non co-routed associated bidirectional TE tunnel.

Signaling Methods and Object Association for Bidirectional LSPs, on page 56, Associated Bidirectional Non Co-routed and Co-routed LSPs, on page 57 provides details.

You can associate a protecting MPLS-TE tunnel with either a working MPLS-TE LSP, protecting MPLS-TE LSP, or both. The working LSP is the primary LSP backed up by the protecting LSP. When a working LSP goes down, the protecting LSP is automatically activated. You can configure a MPLS-TE tunnel to operate without protection as well.

Path Protection, on page 61 provides details.

Signaling Methods and Object Association for Bidirectional LSPs

This section provides an overview of the association signaling methods for the bidirectional LSPs. Two unidirectional LSPs can be bound to form an associated bidirectional LSP in the following scenarios:

- No unidirectional LSP exists, and both must be established.
- Both unidirectional LSPs exist, but the association must be established.
- One unidirectional LSP exists, but the reverse associated LSP must be established.

Configuration information regarding the LSPs can be provided at one or both endpoints of the associated bidirectional LSP. Depending on the method chosen, there are two models of creating an associated bidirectional LSP; single-sided provisioning, and double-sided provisioning.

- Single-sided Provisioning: For the single-sided provisioning, the TE tunnel is configured only on one side. An LSP for this tunnel is initiated by the initiating endpoint with the Association Object inserted in the Path message. The other endpoint then creates the corresponding reverse TE tunnel and signals the reverse LSP in response to this. Currently, there is no support available for configuring single-sided provisioning.

- Double-sided Provisioning: For the double-sided provisioning, two unidirectional TE tunnels are configured independently on both sides. The LSPs for the tunnels are signaled with Association Objects inserted in the Path message by both sides to indicate that the two LSPs are to be associated to form a bidirectional LSP.

Consider this topology (an example of associated bidirectional LSP):
Here, LSP1 from A to B, takes the path A,D,B and LSP2 from B to A takes the path B,D,C,A. These two LSPs, once established and associated, form an associated bidirectional LSP between node A and node B. For the double sided provisioning model, both LSP1 and LSP2 are signaled independently with (Extended) Association Object inserted in the Path message, in which the Association Type indicating double-sided provisioning. In this case, the two unidirectional LSPs are bound together to form an associated bidirectional LSP based on identical Association Objects in the two LSPs' Path messages.

**Association Object:** An Association Object is used to bind unidirectional LSPs originating from both endpoints. The Association Object takes the following values:

- **Association Type:** In order to bind two reverse unidirectional LSPs to be an associated bidirectional LSP, the Association Type must be set to indicate either single sided or double sided LSPs.
- **Association ID:** For both single sided and double sided provisioning, Association ID must be set to a value assigned by the node that originates the association for the bidirectional LSP. This is set to the Tunnel ID of the bound LSP or the Tunnel ID of the binding LSP.
- **Association Source:** For double sided provisioning, Association Source must be set to an address selected by the node that originates the association for the bidirectional LSP. For single sided provisioning, Association Source must be set to an address assigned to the node that originates the LSP.
- **Global ID:** This is the global ID for the association global source. This must be set to the global ID of the node that originates the association for the bidirectional LSP.

**Note**
You must provide identical values for the content of the Association Object on either end of the participating LSPs to ensure successful binding of the LSPs.

Configure Associated Bidirectional Co-routed LSPs, on page 59 describes the procedure to create associated bidirectional co-routed LSPs.

### Associated Bidirectional Non Co-routed and Co-routed LSPs

This section provides an overview of associated bidirectional non co-routed and co-routed LSPs. Establishment of MPLS TE-LSP involves computation of a path between a head-end node to a tail-end node, signaling along the path, and modification of intermediate nodes along the path. The signaling process ensures bandwidth reservation (if signaled bandwidth is lesser than 0 and programming off forwarding entries.

Path computation is performed by the head-end nodes of both the participating LSPs using Constrained Shortest Path First (CSPF). CSPF is the 'shortest path (measured in terms of cost) that satisfies all relevant LSP TE constraints or attributes, such as required bandwidth, priority and so on.

**Associated Bidirectional Non Co-routed LSPs:** A non co-routed bidirectional TE LSP follows two different paths, that is, the forward direction LSP path is different than the reverse direction LSP path. Here is an illustration.
In the above topology:

- The outer paths (in green) are working LSP pairs.
- The inner paths (in red) are protecting LSP pairs.
- Router 1 sets up working LSP to Router 3 and protecting LSP to Router 3 independently.
- Router 3 sets up working LSP to Router 1 and protecting LSP to Router 1 independently.

Non co-routed bidirectional TE LSP is available by default, and no configuration is required.

---

**Note**

In case of non co-routed LSPs, the head-end nodes relax the constraint on having identical forward and reverse paths. Hence, depending on network state you can have identical forward and reverse paths, though the bidirectional LSP is co-routed.

**Associated Bidirectional Co-routed LSPs:** A co-routed bidirectional TE LSP denotes a bidirectional tunnel where the forward direction LSP and reverse direction LSP must follow the same path, for example, the same nodes and paths. Here is an illustration.

In the above topology:

- Paths at the top of the figure (in green) indicate working co-routed LSP pairs.
- Paths at the bottom of the figure (in red) indicate protecting co-routed LSP pairs.
- Router 1 sets up working LSP to Router 3 (in red) after performing bidirectional CSPF and sends reverse explicit route object (ERO) to Router 3. Node Router 3 uses the received reverse ERO to set up reverse red working LSP to Router 1.
- Router 3 sets up protecting LSP to Router 1 (in green) after performing bidirectional CSPF and sends reverse ERO to Router 1. Node Router 1 uses the received reverse ERO to set up reverse green protecting LSP to Router 3.
Configure Associated Bidirectional Co-routed LSPs

A co-routed bidirectional packet LSP is a combination of two LSPs (one in the forward direction and the other in reverse direction) sharing the same path between a pair of ingress and egress nodes. It is established using the extensions to RSVP-TE. This type of LSP can be used to carry any of the standard types of MPLS-based traffic, including Layer 2 VPNs, Layer 2 circuits, and Layer 3 VPNs. You can configure a single BFD session for the bidirectional LSP (that is, you do not need to configure a BFD session for each LSP in each direction). You can also configure a single standby bidirectional LSP to provide a backup for the primary bidirectional LSP.

Before you begin

- You must have symmetric source and destination TE router IDs in order for bidirectional LSPs to be associated.
- Tunnels attributes must be configured identically on both sides of co-routed bidirectional LSP.

**SUMMARY STEPS**

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `bidirectional`
4. `association { id <0-65535> | source-address <IP address> } [global-id <0-4294967295>]`
5. `association type co-routed`
6. `commit`
7. `show mpls traffic-eng tunnels bidirectional-associated co-routed`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>interface tunnel-te tunnel-id</code></td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router# interface tunnel-te 1</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>bidirectional</code></td>
<td>Configure the ingress router for the LSP and include the bidirectional statement to specify that the LSP be established as a bidirectional LSP.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/0/CPU0:router(config-if)# bidirectional</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>`association { id &lt;0-65535&gt;</td>
<td>source-address &lt;IP address&gt; } [global-id &lt;0-4294967295&gt;]`</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/0/CPU0:router(config-if-bidir)# association id 1 source-address 11.0.0.1</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 5** | association type co-routed  
**Example:**  
RP/0/0/CPU0:router(config-if-bidir)#association type co-routed  
| Note  
Association ID, association source and global ID must be configured identically on both the endpoints.  
Specify that the LSP be established as a associated co-routed bidirectional LSP. |
| **Step 6** | commit  
| **Step 7** | show mpls traffic-eng tunnels bidirectional-associated co-routed  
**Example:**  
RP/0/0/CPU0:router#show mpls traffic-eng tunnels bidirectional-associated co-routed  
| Shows details of an associated co-routed bidirectional LSP. |

**Show output for an associated co-routed bidirectional LSP configuration**

This is a sample of the output for the `show mpls traffic-eng tunnels role head` command.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels role head  
Name: tunnel-te1 Destination: 49.49.49.2  
Signalled-Name: IMC0_t1  
Status:  
  Admin: up Oper: up Path: valid Signalling: connected  
  path option 1, type dynamic (Basis for Setup, path weight 20 (reverse 20))  
  path option 1, type dynamic (Basis for Standby, path weight 20 (reverse 20))  
  G-PID: 0x0800 (derived from egress interface properties)  
  Bandwidth Requested: 0 kbps CT0  
  Creation Time: Sun May 4 12:09:56 2014 (03:24:11 ago)  
Config Parameters:  
  Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff  
  Metric Type: TE (default)  
  Hop-limit: disabled  
  Cost-limit: disabled  
  AutoRoute: disabled LockDown: disabled Policy class: not set  
  Forward class: 0 (default)  
  Forwarding-Adjacency: disabled  
  Loadshare: 0 equal loadshares  
  Auto-bw: disabled  
  Fast Reroute: Disabled, Protection Desired: None  
  Path Protection: Enabled  
  Association Type: Single Sided Bidirectional LSPs, Co-routed: YES  
  Association ID: 100, Source: 49.49.49.2  
  Reverse Bandwidth: 0 kbps (CT0), Standby: 0 kbps (CT0)  
  BFD Fast Detection: Enabled  
  BFD Parameters: Min-interval 100 ms (default), Multiplier 3 (default)  
  BFD Bringup Timeout: Interval 60 seconds (default)  
  BFD Initial Dampening: 16000 ms (default)  
  BFD Maximum Dampening: 600000 ms (default)  
  BFD Secondary Dampening: 20000 ms (default)  
  Periodic LSP Ping: Interval 120 seconds (default)  
  Session Down Action: ACTION_REOPTIMIZE, Reopt Timeout: 300  
```
Path Protection

Path protection provides an end-to-end failure recovery mechanism (that is, full path protection) for associated bidirectional MPLS-TE LSPs. Associated bidirectional MPLS-TE LSPs support 1:1 path protection. You can configure the working and protecting LSPs as part of configuring the MPLS-TE tunnel. The working LSP is the primary LSP used to route traffic, while the protecting LSP is a backup for a working LSP. If the working LSP fails, traffic is switched to the protecting LSP until the working LSP is restored, at which time traffic forwarding reverts back to the working LSP.

When FRR is not enabled on a tunnel, and when GAL-BFD and/or Fault OAM is enabled on an associated bidirectional co-routed LSP, path-protection is activated by the FIB running on the line card that hosts the working LSP. The failure on the working LSP can be detected using BFD or Fault OAM.

Configure Path Protection for Associated Bidirectional LSPs, on page 61 provides procedural details.

You can use the `show mpls traffic-eng fast-reroute log` command to confirm whether protection switching has been activated by FIB.

Configure Path Protection for Associated Bidirectional LSPs

SUMMARY STEPS

1. `configure`
2. `interface tunnel-te tunnel-id`
3. `ipv4 unnumbered type interface-path-id`
4. `bfd {fast-detect | encap-mode}
5. `destination ip-address`
6. `bidirectional`
7. `bidirectional association {id <0-65535> | source-address <IP address>} [global-id <0-4294967295>]
8. `association type co-routed`
9. `path-protection`
10. `path-option preference - priority {dynamic | explicit}
11. `commit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>interface tunnel-te tunnel-id</code> Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router# interface tunnel-te 1</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>ipv4 unnumbered type interface-path-id</code> Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered</code></td>
</tr>
</tbody>
</table>

BFD Encap Mode: GAL
Reoptimization after affinity failure: Enabled
Soft Preemption: Disabled
### Configure Path Protection for Associated Bidirectional LSPs

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Loopback0</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Step 4**  
**bfd** `{fast-detect | encap-mode}`  
**Example:**  
RP/0/RSP0/CPU0:IMC0(config-if)# bfd  
RP/0/RSP0/CPU0:IMC0(config-if-tunte-bfd)# fast-detect  
RP/0/RSP0/CPU0:IMC0(config-if-tunte-bfd)# encap-mode gal  

Specify if you want BFD enabled for the LSP over a Generic Associated Channel (G-ACh) or over a IP channel. IP channel is the default.

**Step 5**  
**destination ip-address**  
**Example:**  
RP/0/RP0/CPU0:router(config-if)# destination 49.49.49.2  

Assigns a destination address on the new tunnel. The destination address is the remote node’s MPLS-TE router ID.

**Step 6**  
**bidirectional**  
**Example:**  
Router(config-if)# bidirectional  

Configure the ingress router for the LSP and include the bidirectional statement to specify that the LSP be established as a bidirectional LSP.

**Step 7**  
**bidirectional association** `{id <0-65535> | source-address <IP address>} [global-id <0-4294967295>}`  
**Example:**  
Router(config-if-bidir)# association id 1 source-address 11.0.0.1  

Set the association ID that uniquely identifies the association of LSPs, which is the tunnel ID of the bound LSP (master/slave mode) or the tunnel ID of the binding LSP (peer mode). Also, set the source address to the tunnel sender address of the bound LSP (master/slave mode) or the tunnel sender address of the binding LSP (peer mode). Also, set the ID for associating the global source.  

**Note**  
Association ID, association source and optional global-id must be configured identically on both the endpoints.

**Step 8**  
**association type co-routed**  
**Example:**  
Router(config-if-bidir)# association type co-routed  

Specify that the LSP be configured as a co-routed bidirectional LSP.

**Step 9**  
**path-protection**  
**Example:**  
RP/0/RSP0/CPU0:IMC0(config-if-bidir-co-routed)# path-protection  

Enable path protection.

**Step 10**  
**path-option preference - priority** `{dynamic | explicit}`  
**Example:**  
RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic  

Sets the path option and assigns the path-option ID. Both sides of the co-routed bidirectional LSPs must use dynamic or matching co-routed strict-hop explicit path-option.

**Step 11**  
**commit**

---

**Implementing MPLS Traffic Engineering**

---
Example

Here is a sample configuration with path protection defined for the Associated Bidirectional LSP.

```
RP/0/RSP0/CP00:IMC0#config
RP/0/RSP0/CP00:IMC0(config)#interface tunnel-te 1
RP/0/RSP0/CP00:IMC0(config-if)#ipv4 unnumbered loopback0
RP/0/RSP0/CP00:IMC0(config-if)#destination 49.49.49.2
RP/0/RSP0/CP00:IMC0(config-if)#bidirectional
RP/0/RSP0/CP00:IMC0(config-if-bidir)#association id 100 source-address 49.49.49.2
RP/0/RSP0/CP00:IMC0(config-if-bidir)#association type co-routed
RP/0/RSP0/CP00:IMC0(config-if-bidir-co-routed)#path-protection
RP/0/RSP0/CP00:IMC0(config-if)#path-option 1 dynamic
RP/0/RSP0/CP00:IMC0(config-if)#commit
```

OAM Support for Associated Bidirectional LSPs

You can opt to configure operations, administration and management (OAM) support for Associated Bidirectional LSPs in the following areas:

- **Continuity check**: You can configure bidirectional forwarding detection (BFD) over a Generic Associated Channel (G-ACh) with hardware assist. This allows for BFD Hello packets to be generated and processed in hardware making smaller Hello intervals such as 3.3 ms feasible. For more information on BFD and BFD hardware offload see [Implementing BFD](#) module in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

- **Fault notification**: You can run Fault OAM over associated bidirectional co-routed LSPs to convey fault notification from mid-point to end-point of the LSP. The following fault OAM messages are supported:

  - **Link Down Indication (LDI)**: generated when an interface goes down (for example, to fiber-cut) at mid-point.
  - **Lock Report (LKR)**: generated when an interface is shutdown at mid-point.

  You can configure fault OAM to generate OAM message at mid-point or enable protection switching due to fault OAM at end-point. [Generate Fault OAM Messages at Mid-point, on page 63](#) and [Generate Fault OAM Messages at End-point, on page 64](#) provides procedural details.

- **Fault diagnostics**: You can use the ping and traceroute features as a means to check connectivity and isolate failure points for both co-routed and non-co-routed bidirectional TE tunnels. [MPLS Network Management with MPLS LSP Ping and MPLS SP Traceroute](#) provides details.

Generate Fault OAM Messages at Mid-point

To program all bi-directional LSPs to generate fault OAM message at mid-point use the following steps:

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. fault-oam
4. commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enable fault OAM for an associated bidirectional LSP.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:IMO(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> fault-oam</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:IMC0(config-mpls-te)#fault-oam</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

#### Generate Fault OAM Messages at End-point

In order to enable protection switching due to fault OAM at end-point use the following steps:

#### SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. bidirectional association type co-routed fault-oam
4. commit

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel-te tunnel-id</td>
<td>Enable fault OAM for an associated co-routed bidirectional LSP.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:IMC0(config)#interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bidirectional association type co-routed fault-oam</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:IMC0(config-if)#bidirectional association type co-routed fault-oam</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

#### Pseudowire Call Admission Control

You can use the Pseudowire Call Admission Control (PW CAC) process to check for bandwidth constraints and ensure that once the path is signaled, the links (pseudowires) participating in the bidirectional LSP association have the required bandwidth. Only pseudowires with sufficient bandwidth are admitted in the bidirectional LSP association process. *Configure Pseudowire Bandwidth* in the *Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide* provides procedural details.
Configuration Examples for Cisco MPLS-TE

These configuration examples are used for MPLS-TE:

Build MPLS-TE Topology and Tunnels: Example

The following examples show how to build an OSPF and IS-IS topology:

(OSPF)
...
configure
  mpls traffic-eng
  interface pos 0/6/0/0
  router id loopback 0
  router ospf 1
  router-id 192.168.25.66
  area 0
  interface pos 0/6/0/0
  interface loopback 0
  mpls traffic-eng router-id 192.168.70.1
  mpls traffic-eng area 0
  rsvp
  interface pos 0/6/0/0
  bandwidth 100
  commit
  show mpls traffic-eng topology
  show mpls traffic-eng link-management advertisement
  !
(IX-IX)
...
configure
  mpls traffic-eng
  interface pos 0/6/0/0
  router id loopback 0
  router isis lab
  address-family ipv4 unicast
  mpls traffic-eng level 2
  mpls traffic-eng router-id 192.168.70.2
  !
  interface POS0/0/0/0
  address-family ipv4 unicast
  !

The following example shows how to configure tunnel interfaces:

interface tunnel-te1
  destination 192.168.92.125
  ipv4 unnumbered loopback 0
  path-option 1 dynamic
  bandwidth 100
  commit
  show mpls traffic-eng tunnels
  show ipv4 interface brief
  show mpls traffic-eng link-management admission-control
  !
  interface tunnel-te1
  autoroute announce
Configure IETF DS-TE Tunnels: Example

The following example shows how to configure DS-TE:

```
rsvp
  interface pos 0/6/0/0
  bandwidth rdm 100 150 bc1 50
  mpls traffic-eng
ds-te mode ietf
  interface tunnel-te 1
  bandwidth 10 class-type 1
  commit

configure
  rsvp interface 0/6/0/0
  bandwidth mam max-reservable-bw 400 bc0 300 bc1 200
  mpls traffic-eng
ds-te mode ietf
ds-te mode mam
  interface tunnel-te 1 bandwidth 10 class-type 1
  commit
```

Related Topics
- Configuring a Prestandard DS-TE Tunnel, on page 31
- Prestandard DS-TE Mode, on page 4
Configure MPLS-TE and Fast-Reroute on OSPF: Example

CSPF areas are configured on a per-path-option basis. The following example shows how to use the traffic-engineering tunnels (tunnel-te) interface and the active path for the MPLS-TE tunnel:

```configure
interface tunnel-te 0
  path-option 1 explicit id 6 ospf 126 area 0
  path-option 2 explicit name 234 ospf 3 area 7 verbatim
  path-option 3 dynamic isis mtbf level 1 lockdown
commit
```

Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example

This example shows how to configure the IS-IS overload bit setting in MPLS-TE:

```configure
mpls traffic-eng
  path-selection ignore overload
commit
```

Related Topics

- Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE, on page 38
- Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE, on page 8

Configure Flexible Name-based Tunnel Constraints: Example

The following configuration shows the three-step process used to configure flexible name-based tunnel constraints.

```R2
line console
  exec-timeout 0 0
  width 250
!
logging console debugging
explicit-path name mypath
  index 1 next-address loose ipv4 unicast 3.3.3.3 !
explicit-path name ex_path1
  index 10 next-address loose ipv4 unicast 2.2.2.2 index 20 next-address loose ipv4 unicast 3.3.3.3 !
interface Loopback0
  ipv4 address 22.22.22.22 255.255.255.255 !
interface tunnel-te1
  ipv4 unnumbered Loopback0
  signalled-bandwidth 1000000
  destination 3.3.3.3
  affinity include green
  affinity include yellow
  affinity exclude white
  affinity exclude orange
  path-option 1 dynamic
!```
router isis 1
  is-type level-1
  net 47.0001.0000.0000.0001.00
  nsf cisco
  address-family ipv4 unicast
    metric-style wide
  mpls traffic-eng level-1
  mpls traffic-eng router-id 192.168.70.1
!
interface Loopback0
  passive
  address-family ipv4 unicast
!
  !
interface GigabitEthernet0/1/0/0
  address-family ipv4 unicast
!
  !
interface GigabitEthernet0/1/0/1
  address-family ipv4 unicast
!
  !
interface GigabitEthernet0/1/0/2
  address-family ipv4 unicast
!
  !
interface GigabitEthernet0/1/0/3
  address-family ipv4 unicast
!

rsvp
interface GigabitEthernet0/1/0/0
  bandwidth 1000000 1000000
!
interface GigabitEthernet0/1/0/1
  bandwidth 1000000 1000000
!
interface GigabitEthernet0/1/0/2
  bandwidth 1000000 1000000
!
interface GigabitEthernet0/1/0/3
  bandwidth 1000000 1000000
!

mpls traffic-eng
interface GigabitEthernet0/1/0/0
  attribute-names red purple
!
interface GigabitEthernet0/1/0/1
  attribute-names red orange
!
interface GigabitEthernet0/1/0/2
  attribute-names green purple
!
interface GigabitEthernet0/1/0/3
  attribute-names green orange
!
affinity-map red 1
affinity-map blue 2
affinity-map black 80
affinity-map green 4
affinity-map white 40
affinity-map orange 20
Configure an Interarea Tunnel: Example

The following configuration example shows how to configure a traffic engineering interarea tunnel.

```
configure
interface Tunnel-te1
    ipv4 unnumbered Loopback0
    destination 192.168.20.20
    signalled-bandwidth 300
    path-option 1 explicit name path-tunnel1

explicit-path name path-tunnel1
index 10 next-address loose ipv4 unicast 192.168.40.40
index 20 next-address loose ipv4 unicast 192.168.60.60
index 30 next-address loose ipv4 unicast 192.168.20.20
```

Configure Forwarding Adjacency: Example

The following configuration example shows how to configure an MPLS-TE forwarding adjacency on tunnel-te 68 with a holdtime value of 60:

```
configure
interface tunnel-te 68
forwarding-adjacency holdtime 60
commit
```

Configure PCE: Example

The following configuration example illustrates a PCE configuration:

```
configure
```
mpls traffic-eng
  interface pos 0/6/0/0
  pce address ipv4 192.168.25.66
  router id loopback 0
  router ospf 1
  router-id 192.168.25.66
  area 0
  interface pos 0/6/0/0
  interface loopback 0
  mpls traffic-eng router-id 192.168.70.1
  mpls traffic-eng area 0
  rsvp
  interface pos 0/6/0/0
  bandwidth 100
  commit

The following configuration example illustrates PCC configuration:

configure
  interface tunnel-te 10
  ipv4 unnumbered loopback 0
  destination 1.2.3.4
  path-option 1 dynamic pce
  mpls traffic-eng
  interface pos 0/6/0/0
  router id loopback 0
  router ospf 1
  router-id 192.168.25.66
  area 0
  interface pos 0/6/0/0
  interface loopback 0
  mpls traffic-eng router-id 192.168.70.1
  mpls traffic-eng area 0
  rsvp
  interface pos 0/6/0/0
  bandwidth 100
  commit

Related Topics
  Configuring a Path Computation Client, on page 46
  Configuring a Path Computation Element Address, on page 47
  Configuring PCE Parameters, on page 48
  Path Computation Element, on page 13

Configure Policy-based Tunnel Selection: Example

The following configuration example illustrates a PBTS configuration:

configure
  interface tunnel-te0
  ipv4 unnumbered Loopback3
  signalled-bandwidth 50000
  autoroute announce
  destination 1.5.177.2
  policy-class 2
  path-option 1 dynamic
Configure Automatic Bandwidth: Example

The following configuration example illustrates an automatic bandwidth configuration:

```
configure
  interface tunnel-te6
  auto-bw
    bw-limit min 10000 max 500000
    overflow threshold 50 min 1000 limit 3
    adjustment-threshold 20 min 1000
    application 180
```

Related Topics
- Configuring the Collection Frequency, on page 52
- Configuring the Automatic Bandwidth Functions, on page 53
- MPLS-TE Automatic Bandwidth Overview, on page 15

Configure Entropy Labels for MPLS TE Networks

Most MPLS networks use load balancing techniques for traffic engineering. What causes latency in such widespread networks is the time taken to inspect the label stack at each transit Label Switching Router (LSR) to determine the next hop or path.

The latency can be reduced by inserting a label known as the entropy label on top of the label stack at the ingress LSR. The entropy label contains the keys required by the load balancing function, and thus eliminates the need for deep packet inspection at transit LSRs. The ingress LSR, which has all the information about incoming packets, extracts the load balancing keys from the entropy label and decides the optimum paths for the packets. The transit LSRs use the rest of the label stack to forward the packets along the pre-determined paths.

The advantages of using entropy labels in MPLS networks are:

- Ingress LSRs operate at lower bandwidths than transit LSRs, and are hence the ideal choice for load balancing.

- Transit LSRs do not need to perform deep packet inspection and can effectively load balance the packets as decided by the Ingress LSRs.

- Transit LSRs are spared from the problem of misinterpreting the protocol denoted in the label stack and thereby causing inequitable distribution of traffic across equal cost paths exiting from the LSR.

The following illustration shows the transit of a packet through the MPLS network. The entropy label is attached at the ingress router for load balancing. When the optimum path is determined for the packet, which contains the payload (data) and the label stack, the entropy label is no longer required.
Configuration

1. To configure an MPLS entropy label, use the following configuration.

   RP/0/RP0/CPU0:router(config)# mpls ldp
   RP/0/RP0/CPU0:router(config-ldp)# entropy-label
   RP/0/RP0/CPU0:router(config-ldp)# commit
   RP/0/RP0/CPU0:router(config-ldp)# end

2. Locate the route that needs to use the entropy label for load balancing.

   RP/0/RP0/CPU0:router# show cef exact-route 10.1.6.1 10.1.1.1

   10.1.1.1/32, version 40, internal 0x1000001 0x0 (ptr 0x8d42b4d8) [1], 0x0 (0x8d5c5020),
   0xa20 (0x8e1c0098)
   ...
   Prefix Len 32, traffic index 0, precedence n/a, priority 4
   via Bundle-Ether613
   via 11.1.5.1/32, Bundle-Ether613, 2 dependencies, weight 0, class 0 [flags 0x0]
   path-idx 2 NHID 0x0 [0x8dd02920 0x8dd02810]
   next hop 11.1.5.1/32
   local adjacency
   local label 24002 labels imposed {ImplNull}

3. Use the route to pass the entropy label for load balancing.

   You are prompted for the option of entering the entropy label for multiple source-destination pairs.

   RP/0/RP0/CPU0:router# bundle-hash bundle-Ether 613
   Specify load-balance configuration (L3/3-tuple or L4/7-tuple) (L3,L4): L3
   Single SA/DA pair (IPv4,IPv6) or range (IPv4 only) or Entropy Label (MPLS only): S/R/E
   [S]: E

   Enter Entropy Label (in network byte order): 14001

   Entropy Label 14001 -- Link hashed to is TenGigE0/1/0/8/8

   Another? [y]:

4. Verify if traffic is getting load balanced with the MPLS entropy label configuration.
You have successfully configured an MPLS entropy label in your network.

### Additional References

For additional information related to implementing MPLS-TE, refer to the following references:

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS-TE commands</td>
<td><em>MPLS Traffic Engineering Commands</em> module in <em>MPLS Command Reference for Cisco NCS 6000 Series Routers.</em></td>
</tr>
</tbody>
</table>

#### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

#### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

#### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Format: TXT=79265 bytes) (Status: PROPOSED STANDARD)</td>
</tr>
<tr>
<td></td>
<td>(Format: TXT=22585 bytes) (Status: EXPERIMENTAL)</td>
</tr>
<tr>
<td>RFCs</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

**Technical Assistance**

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>