Implementing MPLS Traffic Engineering on Cisco IOS XR Software

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.

Feature History for Implementing MPLS-TE on Cisco IOS XR Software

Release	Modification
Release 2.0	This feature was introduced on the Cisco CRS-1.
Release 3.0	No modification.
Release 3.2	Support was added for the Cisco XR 12000 Series Router.
Release 3.3.0	Support was added for Generalized MPLS.
Release 3.4.0	Support was added for Flexible Name-based Tunnel Constraints, Interarea MPLS-TE, MPLS-TE Forwarding Adjacency, and GMPLS Protection and Restoration, and GMPLS Path Protection.
Release 3.4.1	Support was added for MPLS-TE and fast reroute link bundling on the Cisco CRS-1.
Release 3.5.0	Support was added for Unequal Load Balancing, IS-IS IP Fast Reroute Loop-free Alternative routing functionality, and Path Computation Element (PCE).

Release 3.6.0	No modification.
Release 3.7.0	Support was added for the following features: • PBTS for L2VPN and IPv6 traffic on the Cisco XR 12000 Series Router. • Ignore Intermediate System-to-Intermediate System (IS-IS) overload bit setting in MPLS-TE. • MPLS-TE/Fast Reroute (FRR) over Virtual Local Area Network (VLAN) interfaces on the Cisco CRS-1.

Contents

Prerequisites for Implementing Cisco MPLS Traffic Engineering

The following 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 for MPLS-TE commands.
- A router that runs Cisco IOS XR software.
- An installed composite mini-image and the MPLS package, or a full composite image.
- IGP activated.

Information About Implementing MPLS Traffic Engineering

To implement MPLS-TE, you should understand the concepts that are described in the following sections:

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.

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 resource reservation protocol (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 the following 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.

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 diff-serv traffic engineering 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.

Diff-Serv TE can be 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. Both modes are described in further detail in the sections that follow.

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, Russian Doll Model (RDM) with two bandwidth pools, global-pool and sub-pool.

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

IETF DS-TE Mode

IETF Diff-Serv 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 the Russian Doll Model (RDM) and the Maximum Allocation Model (MAM) both with two bandwidth pools. Note that 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 Russian Dolls and Maximum Allocation bandwidth constraints models. Both models support up two bandwidth pools.

Cisco IOS XR provides global configuration for the switching between bandwidth constraint models. Both models can be configured on a single interface to pre-configure 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:

- It is easy to use and intuitive.
- It ensures isolation across class types.
- It simultaneously achieves isolation, bandwidth efficiency, and protection against QoS degradation.

Russian Doll Bandwidth Constraint Model

The RDM constraint model has the following characteristics:

- It allows greater sharing of bandwidth among different class types.

- It simultaneously ensures bandwidth efficiency and protection against QoS degradation of all class types.
- It can be 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. While RDM ensures bandwidth efficiency and protection against QoS degradation of class types, it does guarantee isolation across class types.

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 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.

Table 4 list the default TE class and attributes.

Table 4 TE Classes and Priority

TE Class	Class Type	Priority
0	0	7
1	1	7
2	Unused	
3	Unused	
4	0	0
5	1	0
6	Unused	
7	Unused	

Note The default mapping includes four class types.

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 the following events:

- The periodic timer expires (this does not depend on bandwidth pool type).
- The 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.

Note 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 the following requirements for the backup tunnel path:

- The backup tunnel must not pass through the element it protects.

- The 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.

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.

IS-IS IP Fast Reroute Loop-free Alternative

For bandwidth protection, there must be sufficient backup bandwidth available to carry primary tunnel traffic. Use the **ipfrr lfa** command to compute loop-free alternates for all links or neighbors in the event of a link or node failure. To enable node protection on broadcast links, IPRR and bidirectional forwarding detection (BFD) must be enabled on the interface under IS-IS.

Note MPLS FRR and IPFRR cannot be configured on the same interface at the same time.

For information about configuring BFD, see *Cisco IOS XR Interface and Hardware Configuration Guide*.

MPLS-TE and Fast Reroute over Link Bundles

MPLS Traffic Engineering (TE) and Fast Reroute (FRR) are supported over bundle interfaces on the Cisco CRS-1 router only. MPLS-TE/FRR over virtual local area network (VLAN) interfaces is supported on the Cisco CRS-1 router only. Bidirectional forwarding detection (BFD) over VLAN is used as an FRR trigger to obtain more than 50 milliseconds of switchover time on the Cisco CRS-1.

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

- Over POS link bundles
- 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, GigabitEthernet, TenGigE, FastEthernet, and so forth).

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.

Note 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.

Generalized MPLS

Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering consists of extensions to the MPLS-TE mechanisms to control a variety of device types, including optical switches. When GMPLS-TE is used to control an hierarchical optical network—a network with a core of optical switches surrounded by outer layers of routers—it can provide unified control of devices that have very different hardware capabilities. Other control-plane solutions for such network architectures typically use an overlay model, using separate control-planes to manage the optical core and the routed network, respectively, with little or no knowledge passing between them.

GMPLS-TE protocols and extensions include:

- Resource Reservation Protocol (RSVP) for signaling
- Interior Gateway Protocols (IGP) such as Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) for routing
- Link Management Protocol (LMP) for managing link information

The base protocol definitions for RSVP, OSPF, and IS-IS were previously extended for MPLS-TE to provide circuit mechanisms within packet IP networks. These protocols have been extended for GMPLS-TE.

LMP provides facilities similar to Asynchronous Transfer Mode (ATM) Integrated Local Management Interface (ILMI) and Frame Relay Local Management Interface (LMI). LMP also has features addressing the minimal to nonexistent framing support typical of data links on optical switches.

Optical switches differ from packet and cell devices, in that the data links of optical switches typically can carry only transit traffic. This means that traffic entering an optical switch via one data link is required to leave the switch via a different link. For this reason, a data link that connects two neighboring optical devices cannot exchange control frames between the two devices.

Therefore, optical switches typically have separate frame-capable interfaces for sending and receiving control and management traffic. This type of control is referred to as out-of-band. It contrasts with the in-band control of many non-optical networks where control frames and data frames are intermixed on the same link.

To address this characteristic, the GMPLS protocols have been extended to support out-of-band control.

GMPLS Benefits

GMPLS bridges the Internet Protocol (IP) and photonic layers, thereby making possible interoperable and scalable parallel growth in the IP and photonic dimensions.

This allows for rapid service deployment and operational efficiencies, as well as for increased revenue opportunities. A smooth transition becomes possible from a traditional segregated transport and service overlay model to a more unified peer model.

By streamlining support for multiplexing and switching in a hierarchical fashion, and by utilizing the flexible intelligence of MPLS-TE, optical switching GMPLS becomes very helpful for service providers wanting to manage large volumes of traffic in a cost-efficient manner.

GMPLS Support

GMPLS-TE provides support for:

- Open Shortest Path First (OSPF) for bidirectional TE tunnel
- Frame, lambda, and port (fiber) labels
- Numbered/Unnumbered links
- OSPF extensions–Route computation with optical constraints
- RSVP extensions–Graceful Restart
- Graceful deletion
- LSP hierarchy
- Peer model
- Border model Control plane separation
- Interarea/AS-Verbatim
- BGP4/MPLS
- Restoration–Dynamic path computation
- Control channel manager
- Link summary
- Protection and restoration

GMPLS Protection and Restoration

GMPLS provides protection against failed channels (or links) between two adjacent nodes (span protection) and end-to-end dedicated protection (path protection). After the route is computed, signaling to establish the backup paths is carried out through RSVP-TE or CR-LDP. For span protection, 1+1 or M:N protection schemes are provided by establishing secondary paths through the network. In addition, you can use signaling messages to switch from the failed primary path to the secondary path.

Note Only 1:1 end-to-end path protection is supported.

The restoration of a failed path refers to the dynamic establishment of a backup path. This process requires the dynamic allocation of resources and route calculation. The following restoration methods are described:

- Line restoration—Finds an alternate route at an intermediate node.
- Path restoration—Initiates at the source node to route around a failed path within the path for a specific LSP.

Restoration schemes provide more bandwidth usage, because they do not preallocate any resource for an LSP.

GMPLS combines MPLS-FRR and other types of protection, such as SONET/SDH, wavelength, and so forth.

In addition to SONET alarms in POS links, protection and restoration is also triggered by bidirectional forwarding detection (BFD).

1:1 LSP Protection

When one specific protecting LSP or span protects one specific working LSP or span, 1:1 protection scheme occurs. However, normal traffic is transmitted only over one LSP at a time for working or recovery.

1:1 protection with extra traffic refers to the scheme in which extra traffic is carried over a protecting LSP when the protecting LSP is not being used for the recovery of normal traffic. For example, the protecting LSP is in standby mode. When the protecting LSP is required to recover normal traffic from the failed working LSP, the extra traffic is preempted. Extra traffic is not protected, but it can be restored. Extra traffic is transported using the protected LSP resources.

Shared Mesh Restoration and M:N Path Protection

Both shared mesh restoration and M:N (1:N is more practical) path protection offers sharing for protection resources for multiple working LSPs. For 1:N protection, a specific protecting LSP is dedicated to the protection of up to N working LSPs and spans. Shared mesh is defined as preplanned LSP rerouting, which reduces the restoration resource requirements by allowing multiple restoration LSPs to be initiated from distinct ingress nodes to share common resources, such as links and nodes.

End-to-end Recovery

End-to-end recovery refers to an entire LSP from the source for an ingress router endpoint to the destination for an egress router endpoint.

GMPLS Protection Requirements

The GMPLS protection requirements are specific to the protection scheme that is enabled at the data plane. For example, SONET APS or MPLS-FRR are identified as the data level for GMPLS protection.

GMPLS Prerequisites

The following prerequisites are required to implement GMPLS on Cisco IOS XR software:

- You must be in a user group associated with a task group that includes the proper task IDs for GMPLS commands.
- A router that runs Cisco IOS XR software.
- Installation of the Cisco IOS XR software mini-image on the router.

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.

MPLS Traffic Engineering Interarea Tunneling

This section describes the following new extensions of MPLS-TE:

Interarea Support

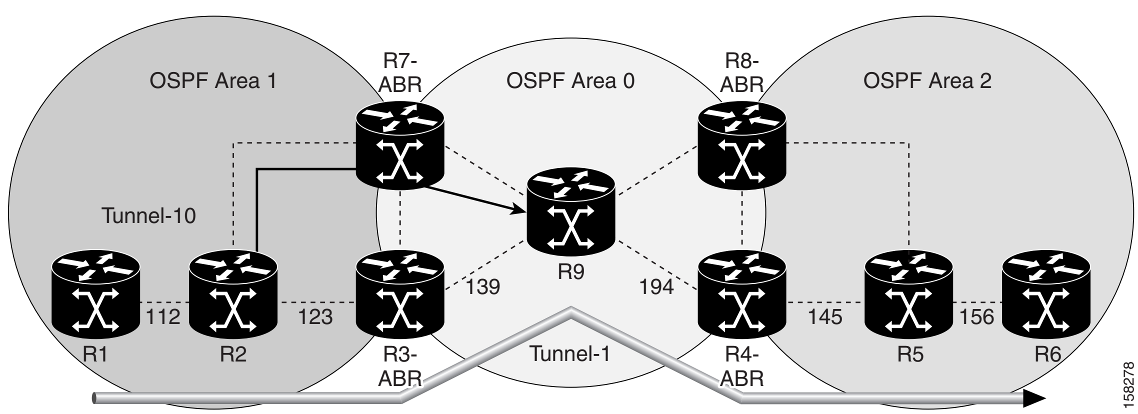
The MPLS-TE interarea tunneling feature allows you to establish TE tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thereby eliminating the requirement that headend and tailend routers reside in a single area.

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 10 shows a typical interarea TE network.

Figure 10 Interarea (OSPF) TE Network Diagram



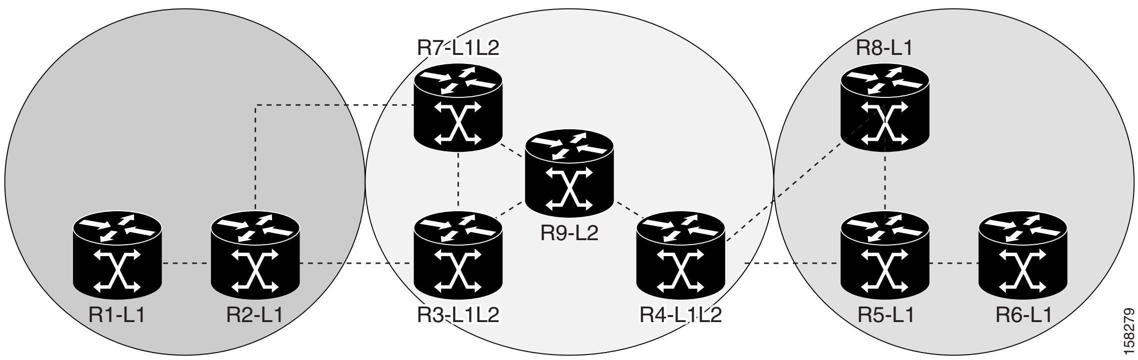
Multiarea Support

Multiarea support allows an ABR LSR to support MPLS-TE in more than one IGP area. A TE LSP will still be 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 11 Interlevel (IS-IS) TE Network



As shown in Figure 11, 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 configure an interarea TE 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 the following 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

Since 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.

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.

MPLS-TE Forwarding Adjacency Restrictions

The following restrictions are listed for the MPLS-TE Forwarding Adjacency feature:

- Using the MPLS-TE Forwarding Adjacency feature increases the size of the IGP database by advertising a TE tunnel as a link.
- The MPLS-TE Forwarding Adjacency feature is supported by Intermediate System-to-Intermediate System (IS-IS).
- When the MPLS-TE Forwarding Adjacency feature 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.

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)
- OSPF

Unequal Load Balancing

Unequal load balancing permits the routing of unequal proportions of traffic through tunnels to a common destination. Load shares on tunnels to the same destination are determined by TE from the tunnel configuration and passed via the MPLS Label Switching Database (LSD) to the Forwarding Information Base (FIB).

Note Load share values are renormalised by the FIB using values suitable for use by the forwarding code; the exact traffic ratios observed may not, therefore, exactly mirror the configured traffic ratios. This effect is more pronounced if there are many parallel tunnels to a destination, or if the load shares assigned to those tunnels are very different. The exact renormalization algorithm used is platform-dependent.

There are two ways to configure load balancing:

- Explicit configuration—Using this method, load shares are explicitly configured on each tunnel.
- Bandwidth configuration—If a tunnel is not configured with load-sharing parameters, the tunnel bandwidth and load-share values are considered equivalent for load-share calculations between tunnels, and a direct comparison between bandwidth and load-share configuration values is calculated.

Note Load shares are not dependent on any configuration other than the load share and bandwidth configured on the tunnel and the state of the global configuration switch.

Path Computation Element

Path Computation Element (PCE) solves the specific issue of inter-domain path computation for MPLS-TE 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).

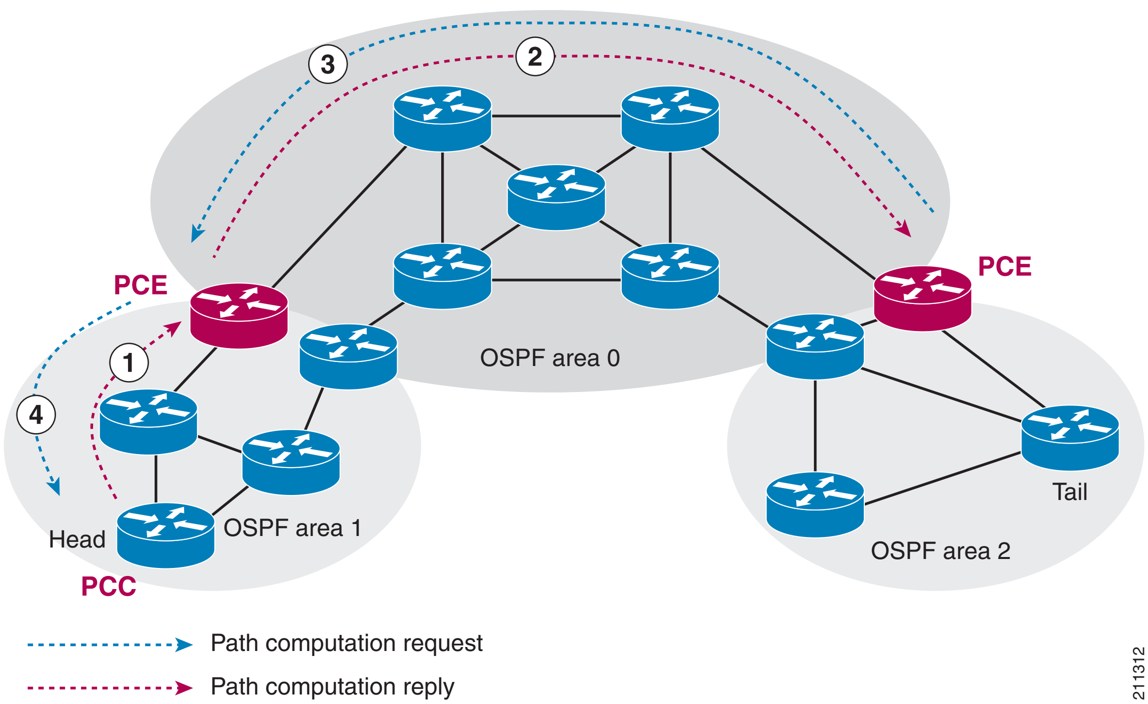
PCE uses area border routers (ABRs) to compute a TE LSP spanning multiple IGP areas as well as computation of Inter-AS TE LSP.

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 12 shows a typical PCE implementation.

Figure 12 Path Computation Element Network Diagram



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

- Message types: Open, PCReq, PCRep, PCErr, Close
- Objects: OPEN, CLOSE, RP, END-POINT, LSPA, BANDWIDTH, METRIC and NO-PATH

Policy-based Tunnel Selection

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

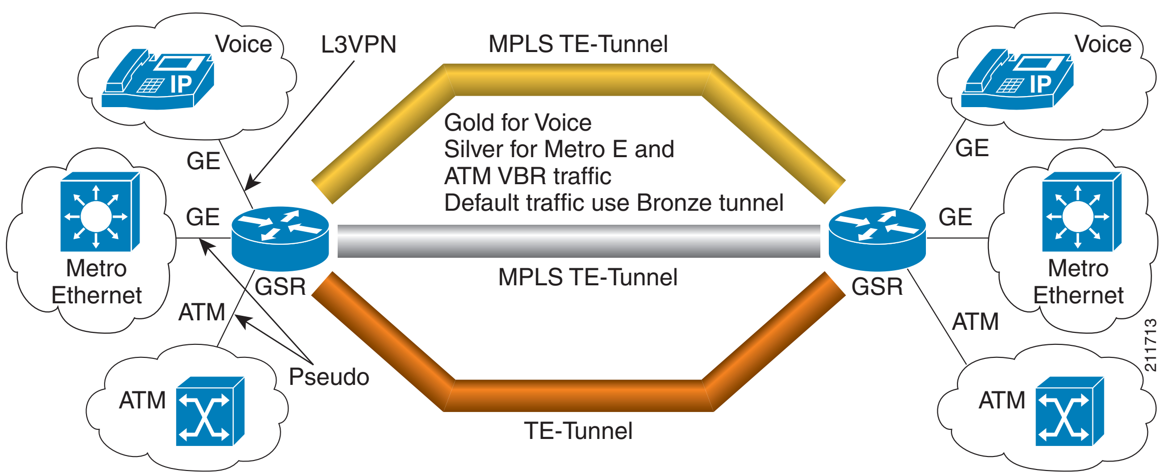
Policy-based Tunnel Selection Overview

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, EXP, or TOS field in the packet. When there are no paths with a default class configured, this traffic is forwarded using the paths with the lowest class value.

Figure 13 illustrates a PBTS implementation.

Figure 13 Policy-based Tunnel Selection Implementation



Policy-based Tunnel Selection Functions

The following PBTS functions are supported on the Cisco CRS-1 router and the Cisco XR 12000 Series Router:

- 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.
- The 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.

The following PBTS functions are supported only on the Cisco XR 12000 Series Router:

- L2VPN preferred path selection lets traffic be directed to a particular TE tunnel.
- 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.
- IPv6 traffic for both 6VPE and 6PE scenarios are supported.

PBTS with Dynamic Tunnel Selection

Note This feature is supported only on the Cisco XR 12000 Series Router.

Dynamic tunnel selection, which is based on class-of-service-based tunnel selection (CBTS), uses post-QoS EXP to select the tunnel. The TE tunnel contains a class attribute that is based on CoS or EXP. Traffic is forwarded on the TE tunnels based on the class attribute. For the balancing group, the traffic can be load-balanced among the tunnels of the same class. The default path is a LDP LSP or a default tunnel.

Restrictions

When implementing PBTS, the following restrictions are listed:

- When you enable QoS EXP remarking on an interface, 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.
- For information about the PBTS default path behavior and the **mpls traffic-eng igp-intact (OSPF)** command or **mpls traffic-eng igp-intact (IS-IS)** command, refer to *Cisco IOS XR Routing Command Reference*.

How to Implement Traffic Engineering on Cisco IOS XR Software

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.

This section explains the following procedures:

Building MPLS-TE Topology

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

Building the MPLS-TE topology is accomplished by performing the following basic steps:

- Enabling MPLS-TE on the port interface.
- Enabling RSVP on the port interface.
- Enabling an IGP such as OSPF or IS-IS for MPLS-TE.

Prerequisites

The following prerequisites are required to build the MPLS-TE topology:

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

- A 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. **router-id** {*interface-id* | *ip-address*}
3. **mpls traffic-eng**
4. **interface** *type interface-id*
5. **exit**
6. **router ospf** *process-name*
7. **router-id** {*interface-id* | *ip-address*}
8. **area** *area-id*
9. **interface** *type interface-id*
10. **interface** *interface-id*
11. **exit**
12. **mpls traffic-eng router-id**
13. **area** *area-id*
14. **exit**
15. **rsvp interface** *type interface-id*
16. **bandwidth** *bandwidth*
17. **end**
    or
    commit
18. **show mpls traffic topology**
19. **show mpls traffic-eng link-management advertisements**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters the configuration mode.

Step 2	**`router id`** {*`interface-id`* | *`ip-address`*} **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if)# router id loopback0`	Specifies the global router ID of the local node. • The router ID can be specified with an interface name or an IP address. By default, MPLS uses the global router ID.

	Command or Action	Purpose
Step 3	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters the MPLS-TE configuration mode.
Step 4	**interface** *type interface-id* Example: `RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0`	Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# exit`	Exits the current configuration mode.
Step 6	**router ospf** process-name **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 1`	Enters a name for the OSPF process.
Step 7	**router-id** {*interface-id* | *ip-address*} **Example:** `RP/0/RP0/CPU0:router(config-router)# router-id 192.168.25.66`	Configures a router ID for the OSPF process using an IP address.
Step 8	**area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-router)# area 0`	Configures an area for the OSPF process. • Backbone areas have an area ID of 0. • Non-backbone areas have a non-zero area ID.
Step 9	**interface** *type interface-id* ***Example:*** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0`	Configures one or more interfaces for the area configured in Step 8.
Step 10	**interface** *interface-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface loopback 0`	Enables IGP on the loopback0 MPLS router ID.
Step 11	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# exit`	Exits the current configuration mode.
Step 12	**mpls traffic-eng router-id loopback 0** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id loopback 0`	Sets the MPLS-TE loopback interface.

	Command or Action	Purpose
Step 13	**area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 0`	Sets the MPLS-TE area.
Step 14	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# exit`	Exits the current configuration mode.

Step 15	**rsvp interface** *type interface-id*
	
	`Example:`
	`RP/0/RP0/CPU0:router(config)# rsvp interface Bundle-POS 500`
		Enters RSVP interface configuration mode and enables RSVP on a particular interface on the originating node (in this case, on the Bundle-POS interface 500).

Step 16	**bandwidth** *bandwidth* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 100`	Sets the reserved RSVP bandwidth available on this interface. **Note** Physical interface bandwidth is not used by MPLS-TE.

Step 17	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-rsvp-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-rsvp-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

	Command or Action	Purpose
Step 18	**show mpls traffic-eng topology** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng topology`	(Optional) Verifies the traffic engineering topology.
Step 19	**show mpls traffic-eng link-management advertisements** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management advertisements`	(Optional) Displays all the link-management advertisements for the links on this node.

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 (see "Building MPLS-TE Topology" section on page 115).

Prerequisites

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

- You must have a router ID for the neighboring router.
- A 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** *number*
3. **destination** *ip-address*
4. **ipv4 unnumbered loopback** *number*
5. **path-option** *path-id* **dynamic**
6. **signaled bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*}
7. **end**
   or
   commit
8. **show mpls traffic-eng tunnels**
9. **show ipv4 interface brief**
10. **show mpls traffic-eng link-management admission-control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te 1`	Enters MPLS-TE interface configuration mode.
Step 3	**destination** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125`	Assigns a destination address on the new tunnel. • The destination address is the remote node's MPLS-TE router ID.
Step 4	**ipv4 unnumbered loopback** *number* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback 0`	Assigns a source address so that forwarding can be performed on the new tunnel.
Step 5	**path-option** *path-id* **dynamic** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic`	Sets the path option to dynamic and also assigns the path ID.
Step 6	**signaled bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*} **Example:** `RP/0/RP0/CPU0:router(config-if)# signaled bandwidth 100`	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).

	Command or Action	Purpose

Step 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 8	**show mpls traffic-eng tunnels** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels`	(Optional) Verifies that the tunnel is connected (in the UP state) and displays all configured TE tunnels.
Step 9	**show ipv4 interface brief** **Example:** `RP/0/RP0/CPU0:router# show ipv4 interface brief`	(Optional) Displays all TE tunnel interfaces.
Step 10	**show mpls traffic-eng link-management admission-control** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management admission-control`	(Optional) Displays all the tunnels on this node.

Configuring Forwarding over the MPLS-TE Tunnel

Perform this task to configure forwarding over the MPLS-TE tunnel created in the previous task (see "Creating an MPLS-TE Tunnel" section on page 119).

This procedure allows MPLS packets to be forwarded on the link between network neighbors.

Prerequisites

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

- You must have a router ID for the neighboring router.
- A 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** *number*
3. **ipv4 unnumbered loopback** *number*
4. **autoroute announce**
5. **exit**
6. **router static address-family ipv4 unicast** *prefix mask ip-address interface type*
7. **end**
   or
   commit
8. **ping** {*ip-address* | *hostname*}
9. **show mpls traffic-eng autoroute**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te 1`	Enters MPLS-TE interface configuration mode.
Step 3	**ipv4 unnumbered loopback** *number* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback 0`	Assigns a source address so that forwarding can be performed on the new tunnel.
Step 4	**autoroute announce** **Example:** `RP/0/RP0/CPU0:router(config-if)# autoroute announce`	Enables messages that notify the neighbor nodes about the routes that are forwarding.

	Command or Action	Purpose
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits the current configuration mode.
Step 6	**router static address-family ipv4 unicast** *prefix mask ip-address interface type* **Example:** `RP/0/RP0/CPU0:router(config)# router static address-family ipv4 unicast 2.2.2.2/32 tunnel-te 1`	(Optional) Enables a route using IP version 4 addressing, identifies the destination address and the tunnel where forwarding is enabled. • This configuration is used for static routes when autoroute announce config is not used.

Step 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 8	**ping** {*ip-address* | *hostname*} **Example:** `RP/0/RP0/CPU0:router# ping 192.168.12.52`	(Optional) Checks for connectivity to a particular IP address or host name.
Step 9	**show mpls traffic-eng autoroute** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng autoroute`	(Optional) Verifies forwarding by displaying what is advertised to IGP for the TE tunnel.

Protecting MPLS Tunnels with Fast Reroute

Perform this task to protect MPLS-TE tunnels, as created in the previous task (see "Configuring Forwarding over the MPLS-TE Tunnel" section on page 121).

Note 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.

Prerequisites

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

- You must have a router ID for the neighboring router.
- A 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 and a backup tunnel (see "Creating an MPLS-TE Tunnel" section on page 119).

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *number*
3. **fast-reroute**
4. **exit**
5. **mpls traffic-eng interface** *type interface-id*
6. **backup-path tunnel-te** *tunnel-number*
7. **exit**
8. **interface tunnel-te** *tunnel-id*
9. **backup-bw** {*bandwidth* | **sub-pool** {*bandwidth* | **unlimited**} | **global-pool** {*bandwidth* | **unlimited**}}
10. **ipv4 unnumbered loopback** *number*
11. **path-option** *path-id* **explicit name** *explicit-path-name*
12. **destination** *A.B.C.D*
13. **end**
    or
    commit
14. **show mpls traffic-eng tunnels backup**
15. **show mpls traffic-eng tunnels protection**
16. **show mpls traffic-eng fast-reroute database**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te1`	Enters MPLS-TE interface configuration mode.
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 interface** *type interface-id* ***Example:*** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng interface pos0/6/0/0`	Enters the MPLS-TE configuration mode, and enables traffic engineering on a particular interface on the originating node.
Step 6	**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 7	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits the current configuration mode.
Step 8	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te2`	Enters MPLS-TE interface configuration mode.
Step 9	**backup-bw** {*bandwidth* | **sub-pool** {*bandwidth* | **unlimited**} | **global-pool** {*bandwidth* | **unlimited**}} **Example:** `RP/0/RP0/CPU0:router(config-if)# 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.

	Command or Action	Purpose
Step 10	**ipv4 unnumbered loopback** *number* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback 0`	Assigns a source address to set up forwarding on the new tunnel.
Step 11	**path-option** *path-id* **explicit name** *explicit-path-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-path`	Sets the path option to explicit with a given name (previously configured) and assigns the path ID.
Step 12	**destination** *A.B.C.D* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125`	Assigns a destination address on the new tunnel. • The destination address is the remote node's MPLS-TE router ID. • The 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 13	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 14	**show mpls traffic-eng tunnels backup** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels backup`	(Optional) Displays the backup tunnel information.

	Command or Action	Purpose
Step 15	**show mpls traffic-eng tunnels protection** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels protection`	(Optional) Displays the tunnel protection information.
Step 16	**show mpls traffic-eng fast-reroute database** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng fast-reroute database`	(Optional) Displays the protected tunnel state (for example, the tunnel's current ready or active state).

Configuring a Prestandard Diff-Serv TE Tunnel

Perform this task to configure a Prestandard Diff-Serv TE tunnel.

Prerequisites

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

- You must have a router ID for the neighboring router.
- A 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-id*
3. **bandwidth** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}]
4. **exit**
5. **interface tunnel-te** *number*
6. **signaled bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*}
7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**rsvp interface** *type interface-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** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}] **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 100 150 sub-pool 50`	Sets the reserved RSVP bandwidth available on this interface. **Note** Physical interface bandwidth is not used by MPLS-TE.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# exit`	Exits the current configuration mode.
Step 5	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te2`	Enters MPLS-TE interface configuration mode.

	Command or Action	Purpose
Step 6	**signaled bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*} **Example:** `RP/0/RP0/CPU0:router(config-if)# 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 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring an IETF Diff-Serv TE Tunnel Using RDM

Perform this task to create an IETF mode differentiated services traffic engineering tunnel using RDM.

Prerequisites

The following prerequisites are required to create an IETF mode differentiated services traffic engineering tunnel using RDM:

- You must have a router ID for the neighboring router.
- A 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-id*
3. **bandwidth** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}]
4. **exit**

5. **mpls traffic-eng**
6. **ds-te mode ietf**
7. **exit**
8. **interface tunnel-te** *number*
9. **signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*}
10. **end**
    or
    commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**rsvp interface** *type interface-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** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}] **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth rdm 100 150`	Sets the reserved RSVP bandwidth available on this interface. **Note** Physical interface bandwidth is not used by MPLS-TE.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# exit`	Exits the current configuration mode.
Step 5	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.
Step 6	**ds-te mode ietf** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# ds-te mode ietf`	Enables IETF DS-TE mode and default TE class map. Configure IETF DS-TE mode on all network nodes.
Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# exit`	Exits the current configuration mode.

	Command or Action	Purpose
Step 8	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te4`	Enters MPLS-TE interface configuration mode.
Step 9	**signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*} **Example:** `RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10 class-type 1`	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).

Step 10	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring an IETF Diff-Serv 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.

Prerequisites

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.
- A 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-id*
3. **bandwidth** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}]
4. **exit**
5. **mpls traffic-eng**
6. **ds-te mode ietf**
7. **ds-te bc-model mam**
8. **exit**
9. **interface tunnel-te** *number*
10. **signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*}
11. **end**
    or
    commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**rsvp interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config)# rsvp interface pos0/6/0/0`	Enters RSVP configuration mode and selects the RSVP interface.
Step 3	**bandwidth** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}] **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth mam max-reservable-bw 400 bc0 300 bc1 200`	Sets the reserved RSVP bandwidth available on this interface. **Note** Physical interface bandwidth is not used by MPLS-TE.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# exit`	Exits the current configuration mode.
Step 5	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.

	Command or Action	Purpose
Step 6	**`ds-te mode ietf`** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# ds-te mode ietf`	Enables IETF DS-TE mode and default TE class map. Configure IETF DS-TE mode on all nodes in the network.
Step 7	**`ds-te bc-model mam`** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# ds-te bc-model mam`	Enables the MAM bandwidth constraint model globally.
Step 8	**`exit`** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# exit`	Exits the current configuration mode.
Step 9	**`interface tunnel-te`** *`number`* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te4`	Enters MPLS-TE interface configuration mode.
Step 10	**`signalled-bandwidth`** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*} **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 10 class-type 1`	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).

Step 11	**`end`**
	or
	`commit`
	
	Example:
	`RP/0/RP0/CPU0:router(config-rsvp-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-rsvp-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Ignore Integrated Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE

Perform this task to configure an overload node avoidance to 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 path-selection ignore overload**
3. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**mpls traffic-eng path-selection ignore overload** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng path-selection ignore overload`	Ignores the Intermediate System-to-Intermediate System (IS-IS) overload bit setting for MPLS-TE.

Step 3	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring GMPLS on Cisco IOS XR Software

To fully configure GMPLS, you must complete the following high-level tasks in order:

- Configuring IPCC Control Channel Information, page MPC-136
- Configuring Local and Remote TE Links, page MPC-139
- Configuring Numbered and Unnumbered Optical TE Tunnels, page MPC-154
- Configuring LSP Hierarchy, page MPC-159
- Configuring Border Control Model, page MPC-160
- Configuring Path Protection, page MPC-161

Note These high-level tasks are broken down into, in some cases, several subtasks.

Configuring IPCC Control Channel Information

This section includes the following subtasks:

- Configuring Router IDs, page MPC-136
- Configuring OSPF over IPCC, page MPC-138

Note You must configure each subtask on both the headend and tailend router.

Configuring Router IDs

Perform this task to configure the router ID for the headend and tailend routers.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-id*
3. **ipv4 address** *A.B.C.D/prefix*
4. **exit**
5. **configure**
6. **router-id** {*interface-id* | *ip-address*}
7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface POS0/6/0/0`	Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.

	Command or Action	Purpose
Step 3	**ipv4 address** *A.B.C.D/prefix* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0`	Specifies a primary or secondary IPv4 address for an interface. • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# exit`	Exits the current configuration mode.
Step 5	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Re-enters global configuration mode.
Step 6	**router id** {*interface-id* | *ip-address*} **Example:** `RP/0/RP0/CPU0:router(config-if)# router id loopback 0`	Specifies the global router ID of the local node. • The router ID can be specified with an interface name or an IP address. By default, MPLS uses the global router ID.

Step 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring OSPF over IPCC

Perform this task to configure OSPF over IPCC on both the headend and tailend routers.

The IGP interface ID is configured for control network, specifically for the signaling plane in the optical domain.

Note IPCC support is restricted to routed, out-of-fiber, and out-of-band.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. **area** *area-id*
4. **interface** *interface-id*
5. **exit**
6. **mpls traffic-eng router-id** {*interface-id* | *ip-address*}
7. **mpls traffic-eng area** *area-id*
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 1`	Configures OSPF routing and assigns a process name.
Step 3	**area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 0`	Configures an area ID for the OSPF process (either as a decimal value or IP address): • Backbone areas have an area ID of 0. • Non-backbone areas have a nonzero area ID.
Step 4	**interface** *interface-id* **Example:** `RP/0/RP0/CPU0:router((config-ospf-ar)# interface Loopback 0`	Enables IGP on the interface. **Note** Use this command to configure any interface included in the control network.

	Command or Action	Purpose
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit`	Exits the current configuration mode.
Step 6	**mpls traffic-eng router-id** {*interface-id* | *ip-address*} **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng router id 192.168.25.66`	Configures a router ID for the OSPF process using an IP address.
Step 7	**mpls traffic-eng area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng area 0`	Configures the MPLS-TE area.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Local and Remote TE Links

The subtasks in this section describe how to configure local and remote MPLS-TE link parameters for numbered and unnumbered TE links on both headend and tailend routers.

This section includes the following subtasks:

- Configuring Numbered and Unnumbered Links, page MPC-140
- Configuring Local Reservable Bandwidth, page MPC-142
- Configuring Local Switching Capability Descriptors, page MPC-143
- Configuring Persistent Interface Index, page MPC-145

- Enabling LMP Message Exchange, page MPC-146
- Configuring Remote TE Link Adjacency Information for Numbered Links, page MPC-150

Configuring Numbered and Unnumbered Links

Perform this task to configure numbered and unnumbered links.

Note Unnumbered TE links use the IP address of the associated interface.

SUMMARY OF STEPS

1. **configure**
2. **interface** *type interface-id*
3. **ipv4 address** *ipv4-address mask*
   or
   ipv4 unnumbered interface *type interface-id*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface POS0/6/0/0`	Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.
Step 3	**ipv4 address** *ipv4-address mask* **or** **ipv4 unnumbered interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0`	Specifies a primary or secondary IPv4 address for an interface. • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash. or • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface. **Note** If you configured a unnumbered GigE interface in Step 2 and selected the ipv4 unnumbered interface type option in this step, you must enter the **ipv4 point-to-point** command to configure point-to-point interface mode.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Local Reservable Bandwidth

Perform this task to configure the local reservable bandwidth for the data bearer channels.

SUMMARY STEPS

1. **configure**
2. **rsvp interface** *type interface-id*
3. **bandwidth** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}]
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**rsvp interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config)# rsvp interface POS0/6/0/0`	Enters RSVP configuration mode and selects an RSVP interface ID.

	Command or Action	Purpose
Step 3	**bandwidth** [*0 - 4294967295*] [**bc0**] [*global-pool*] [**mam** {*0-4294967295* | *max-reservable-bandwidth*}] [**rdm** {*0-4294967295* | **bc0** | *global-pool*}] **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 2488320 2488320`	Sets the reserved RSVP bandwidth available on this interface. **Note** MPLS-TE can use only the amount of bandwidth specified using this command on the configured interface.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-rsvp-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-rsvp-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Local Switching Capability Descriptors

Perform this task to configure the local switching capability descriptor.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **interface** *type interface-id*
4. **flooding-igp ospf** *instance-id* **area** *area-id*
5. **switching key** *cap*
6. **encoding {sonet/sdh | ethernet}**
7. **capability** {**psc1** | **lsc** | **fsc**}
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.
Step 3	**interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0`	Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.
Step 4	**flooding-igp ospf** *instance-id* **area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if)# flooding-igp ospf 0 1`	Specifies the IGP OSPF interface ID and area where the TE links are to be flooded.
Step 5	**switching key** *cap* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if)# switching key 0`	Specifies the switching configuration for the interface and enters switching key submode where you will configure encoding and capability. **Note** The recommended switch key value is 0.
Step 6	**encoding** {**sonet/sdh** | **ethernet**} **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# encoding ethernet`	Specifies the interface encoding type, as follows: • sonet/sdh, or POS • ethernet, or GigE

	Command or Action	Purpose
Step 7	**capability** {**psc1** | **lsc** | **fsc**} **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# capability psc1`	Specifies the interface switching capability type. The recommended switch capability type is **psc1**.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# end`
	or
	`RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Persistent Interface Index

Perform this task to preserve the LMP interface index across all interfaces on the router.

SUMMARY STEPS

1. **configure**
2. **snmp-server ifindex persist**
3. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server ifindex persist** **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server ifindex persist`	Enables ifindex persistence globally on all Simple Network Management Protocol (SNMP) interfaces.

Step 3	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling LMP Message Exchange

Perform the following task to enable LMP message exchange.

LMP is enabled by default. You can disable LMP on a per neighbor basis using the **lmp static** command in LMP protocol neighbor submode.

Note LMP is recommended unless the peer optical device does not support LMP (in which case it is necessary to disable it at both ends).

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **lmp neighbor** *name*

4. **ipcc routed**
5. **remote node-id** *node-id*
6. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.
Step 3	**lmp neighbor** *name* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# lmp neighbor OXC1`	Configures or updates a LMP neighbor and its associated parameters.
Step 4	**ipcc routed** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# ipcc routed`	Configures a routable Internet Protocol Control Channel (IPCC).

	Command or Action	Purpose
Step 5	`remote node-id` *node-id* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# remote node-id 2.2.2.2`	Configures the remote node ID for an LMP neighbor. **Note** The *node-id* value can also be an IPv4 address

Step 6	`end`
	or
	`commit`
	
	Example:
	`RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# end`
	or
	`RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Disabling LMP Message Exchange

Perform the following task to disable LMP message exchange.

LMP is enabled by default. You can disable LMP on a per neighbor basis using the **lmp static** command in LMP protocol neighbor submode.

Note LMP is recommended unless the peer optical device does not support LMP (in which case it is necessary to disable it at both ends).

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **lmp neighbor** *name*
4. **lmp static**
5. **ipcc routed**
6. **remote node-id** *node-id*
7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.
Step 3	**lmp neighbor** *name* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# lmp neighbor OXC1`	Configures or updates a LMP neighbor and its associated parameters.
Step 4	**lmp static** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# lmp static`	Disables dynamic LMP procedures for the specified neighbor, including LMP hello and LMP link summary. **Note** Use this command for neighbors that do not support dynamic lmp procedures.
Step 5	**ipcc routed** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# ipcc routed`	Configures a routable IPCC.

	Command or Action	Purpose
Step 6	**remote node-id** *node-id* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# remote node-id 2.2.2.2`	Configures the remote node ID for an LMP neighbor. **Note** The node ID value must be an IPv4 address.

Step 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# end`
	or
	`RP/0/RP0/CPU0:router(config-rsvp-if-sw-0x1)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Remote TE Link Adjacency Information for Numbered Links

Perform this task to configure remote TE link adjacency information for numbered links.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **interface** *type interface-id*
4. **lmp data-link adjacency**
5. **remote switching-capability** {**fsc** | **lsc** | **psc1**}
6. **remote interface-id unnum** *value*
7. **remote te-link ipv4 A.B.C.D**
8. **exit**
9. **lmp neighbor** *name*
10. **remote node-id** *A.B.C.D*
11. **end**
    or
    commit
12. **show mpls lmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.
Step 3	**interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0`	Enters MPLS-TE interface configuration mode and enables TE on a particular interface on the originating node.
Step 4	**lmp data-link adjacency** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if)# lmp data-link adjacency`	Configures LMP neighbor remote TE links.
Step 5	**remote switching-capability** {**fsc** | **lsc** | **psc1**} **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote switching-capability lsc`	Configures the remote LMP MPLS-TE interface switching capability.
Step 6	**remote interface-id unnum** *interface identifier* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote interface-id unnum 7`	Configures the unnumbered interface identifier. **Note** Identifiers you specify using this command are the values assigned by the neighbor at the remote side.
Step 7	**remote te-link ipv4 A.B.C.D** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote te-link ipv4 10.10.10.10`	Configures the remote LMP MPLS-TE link ID address.
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit`	Exits the current configuration mode.
Step 9	**lmp neighbor** *name* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# neighbor OXC1`	Configures or updates an LMP neighbor and its associated parameters.

	Command or Action	Purpose
Step 10	**remote node-id** ***A.B.C.D*** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote te-link-id ipv4 10.10.10.10`	Configures the remote LMP MPLS-TE link ID address.

Step 11	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# end`
	or
	`RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 12	**show mpls lmp** **Example:** `RP/0/RP0/CPU0:router# show mpls lmp`	Verifies the assigned value for the local interface identifiers.

Configuring Remote TE Link Adjacency Information for Unnumbered Links

Perform this task to configure remote TE link adjacency information for unnumbered links.

Note To display the assigned value for the local interface identifiers, use the **show mpls lmp** command.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **interface** *type interface-id*
4. **lmp data-link adjacency**
5. **neighbor** *name*
6. **remote te-link-id unnum**
7. **remote interface-id unnum**

8. **remote switching-capability**
9. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.
Step 3	**interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0`	Enters MPLS-TE interface configuration mode and enables TE on a particular interface on the originating node.
Step 4	**lmp data link adjacency** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if)# lmp data-link adjacency`	Configures LMP neighbor remote TE links.
Step 5	**neighbor** *name* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# neighbor OXC1`	Configures or updates a LMP neighbor and its associated parameters.
Step 6	**remote te-link-id unnum** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote te-link-id unnum 111`	Configures the unnumbered interface and identifier.
Step 7	**remote interface-id unnum** *interface identifier* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote interface-id unnum 7`	Configures the unnumbered interface identifier. **Note** Identifiers you specify using this command are the values assigned by the neighbor at the remote side.

	Command or Action	Purpose
Step 8	**remote switching-capability** {**fsc** | **lsc** | **psc1**} **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote switching-capability lsc`	Configures emote the LMP MPLS-TE interface switching capability.

Step 9	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# end`
	or
	`RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Numbered and Unnumbered Optical TE Tunnels

This section includes the following subtasks:

- Configuring an Optical TE Tunnel Using Dynamic Path Option, page MPC-154
- Configuring an Optical TE Tunnel Using Explicit Path Option, page MPC-157

Note Before you can successfully bring optical TE tunnels “up,” you must complete the procedures in the preceding sections.

The following characteristics can apply to the headend (or, signaling) router:

- Tunnels can be numbered or unnumbered.
- Tunnels can be dynamic or explicit.

The following characteristics can apply to the tailend (or, passive) router:

- Tunnels can be numbered or unnumbered.
- Tunnels must use the explicit path-option.

Configuring an Optical TE Tunnel Using Dynamic Path Option

Perform this task to configure a numbered or unnumbered optical tunnel on a router; in this example, the dynamic path option on the headend router.

The dynamic option does not require that you specify the different hops to be taken along the way. The hops are calculated automatically.

Note This section provides two examples that describe how to configure a optical tunnels. It does not include procedures for every option available on the headend and tailend routers.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *number*
3. **ipv4 address** *A.B.C.D/prefix*
   or
   ipv4 unnumbered interface *type interface-id*
4. **switching transit** *switching type* **encoding** *encoding type*
5. **priority** *setup-priority hold-priority*
6. **signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*}
7. **destination** *A.B.C.D*
8. **path-option** *path-id* **dynamic**
9. **direction** [**bidirectional]**
10. **end**
    or
    commit

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters global configuration mode.

Step 2	`interface tunnel-te` *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te1`	Enters MPLS-TE interface configuration mode.

	Command or Action	Purpose
Step 3	**ipv4 address** *A.B.C.D/prefix* or **ipv4 unnumbered interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0`	Specifies a primary or secondary IPv4 address for an interface. • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash. or • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.
Step 4	**switching transit** *switching type* **encoding** *encoding type* **Example:** `RP/0/RP0/CPU0:router(config-if)# switching transit lsc encoding sonetsdh`	Specifies the switching capability and encoding types for all transit TE links used to signal the optical tunnel.
Step 5	**priority** *setup-priority hold-priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# priority 1 1`	Configures setup and reservation priorities for MPLS-TE tunnels.
Step 6	**signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*} **Example:** `RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10 class-type 1`	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).
Step 7	**destination** *A.B.C.D* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125`	Assigns a destination address on the new tunnel. • The destination address is the remote node's MPLS-TE router ID. • The destination address is the merge point between backup and protected tunnels.
Step 8	**path-option** *path-id* **dynamic** **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option l dynamic`	Configures the dynamic path option and path ID.

	Command or Action	Purpose
Step 9	**direction** [**bidirectional**] **Example:** `RP/0/RP0/CPU0:router(config-if)# direction bidirection`	Configures a bidirectional optical tunnel for GMPLS.

Step 10	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring an Optical TE Tunnel Using Explicit Path Option

Perform this task to configure a numbered or unnumbered optical TE tunnel on a router.

This task can apply to both the headend and tailend router.

Note You cannot configure dynamic tunnels on the tailend router.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *number*
3. **ipv4 address** *ipv4-address mask*
   or
   ipv4 unnumbered interface *type interface-id*
4. **passive**
5. **match identifier**
6. **destination** *A.B.C.D*

7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router# interface POS9/0`	Moves configuration to the interface level, directing subsequent configuration commands to the specified interface.
Step 2	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router# interface POS9/0`	Enters MPLS-TE interface configuration mode.
Step 3	**ipv4 address** ipv4-address mask **or** **ipv4 unnumbered interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router# interface POS9/0`	Specifies a primary or secondary IPv4 address for an interface. • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash. or • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.
Step 4	**passive** **Example:** `RP/0/RP0/CPU0:router#` **`passive`**	Configures a passive interface. **Note** The tailend (passive) router does not signal the tunnel, it simply accepts a connection from the headend router. The tailend router supports the same configuration as the headend router.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 5</td><td><code>match identifier</code>

Example:
<code>RP/0/RP0/CPU0:router# match identifier</code></td><td>Configures the match identifier. You must enter the hostname for the head router then underscore _t, and the tunnel number for the head router. If tunnel-te1 is configured on the head router with a hostname of gmpls1, CLI is match identifier gmpls1_t1.

Note The match identifier must correspond to the tunnel-te number configured on the headend router. Together with the address specified using the destination keyword, this identifier uniquely identifies acceptable incoming tunnel requests.</td></tr>
<tr><td>Step 6</td><td><code>destination A.B.C.D</code>

Example:
<code>RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125</code></td><td>Assigns a destination address on the new tunnel.
The destination address is the remote node’s MPLS-TE router ID.
The destination address is the merge point between backup and protected tunnels.</td></tr>
<tr><td>Step 7</td><td><code>end</code>
or
<code>commit</code>

Example:
<code>RP/0/RP0/CPU0:router(config-if)# end</code>
or
<code>RP/0/RP0/CPU0:router(config-if)# commit</code></td><td>Saves configuration changes.
When you issue the end command, the system prompts you to commit changes:
<code>Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:</code>
Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td></tr>
</table>

Configuring LSP Hierarchy

This section describes the high-level steps required to configure LSP hierarchy.

LSP hierarchy allows standard MPLS-TE tunnels to be established over GMPLS-TE tunnels.

Consider the following information when configuring LSP hierarchy:

- LSP hierarchy supports numbered optical TE tunnels with IPv4 addresses only.
- LSP hierarchy supports numbered optical TE tunnels using numbered or unnumbered TE links.

Note Before you can successfully configure LSP hierarchy, you must first establish a numbered optical tunnel between the headend and tailend routers, as described in Configuring Numbered and Unnumbered Optical TE Tunnels, page MPC-154.

To configure LSP hierarchy, you must perform a series of tasks that have been previously described in this GMPLS configuration section. The tasks, which must be completed in the order presented, are as follows:

1. Establish an optical TE tunnel.
2. Configure an optical TE tunnel under IGP.
3. Configure the bandwidth on the optical TE tunnel.
4. Configure the optical TE tunnel as a TE link.
5. Configure an MPLS-TE tunnel.

Configuring Border Control Model

Border model lets you specify the optical core tunnels to be advertised to edge packet topologies. Using this model, the entire topology is stored in a separate packet instance, allowing packet networks where these optical tunnels are advertised to use LSP hierarchy to signal an MPLS tunnel over the optical tunnel.

Consider the following information when configuring protection and restoration:

- The GMPLS optical TE tunnel must be numbered and have a valid IPv4 address.
- The router ID, which is used for the IGP area and interface ID, must be consistent in all areas.
- The OSPF interface ID may be a numeric or alphanumeric.

Note Border model control functionality is provided for multiple IGP instances in one area or in multiple IGP areas.

To configure border control model functionality, you will perform a series of tasks that have been previously described in this GMPLS configuration section. The tasks, which must be completed in the order presented, are as follows:

1. Configure two optical tunnels on different interfaces.

Note When configuring IGP, you must keep the optical and packet topology information in separate routing tables.

2. Configure OSPF adjacency on each tunnel.
3. Configure bandwidth on each tunnel.
4. Configure packet tunnels.

Configuring Path Protection

This section provides the following sections to configure path protection:

- Configuring an LSP, page MPC-161
- Forcing Reversion of the LSP, page MPC-164

Configuring an LSP

Perform this task to configure an LSP for an explicit path.

Path protection is enabled on a tunnel by adding an additional path option configuration at the active end. The path can be configured either explicitly or dynamically.

Note When the dynamic option is used for both working and protecting LSPs, CSPF extensions are used to determine paths with different degrees of diversity. When the paths are computed, they are used over the lifetime of the LSPs. The nodes on the path of the LSP determine if the PSR is or is not for a given LSP. This determination is based on information that is obtained at signaling.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *number*
3. **ipv4 address** *ipv4-address mask*
   or
   ipv4 unnumbered interface *type interface-id*
4. **signalled-name** *name*
5. **switching transit** *capability switching type* **encoding** *encoding type*
6. **switching endpoint** *capability switching type* **encoding** *encoding type*
7. **priority** *setup-priority hold-priority*
8. **signalled-bandwidth** {*bandwidth* **[class-type** *ct*] | **sub-pool** *bandwidth*}
9. **destination** *A.B.C.D*
10. **direction** [**bidirectional]**
11. **path-option** *path-id* **explicit** {**name** *pathname* | *path-number*}
12. **path-option protecting** *path-id* **explicit** {**name** *pathname* | *path-number*}
13. **end**
    or
    commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te1`	Enters tunnel-te interface configuration mode.
Step 3	**ipv4 address** ipv4-address mask or **ipv4 unnumbered interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 99.99.99.2 255.255.255.254`	Specifies a primary or secondary IPv4 address for an interface. • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash. or • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.
Step 4	**signalled-name** *name* **Example:** `RP/0/RP0/CPU0:router(config-if)# signalled-name tunnel-te1`	Configures the name of the tunnel required for an MPLS TE tunnel. • Use the *name* argument to specify the signal for the tunnel.
Step 5	**switching transit** *capability switching type* **encoding** *encoding type* **Example:** `RP/0/RP0/CPU0:router(config-if)# switching transit lsc encoding sonetsdh`	Specifies the switching capability and encoding types for all transit TE links used to signal the optical tunnel to configure an optical LSP.
Step 6	**switching endpoint** *capability* switching type **encoding** *encoding type* **Example:** `RP/0/RP0/CPU0:router(config-if)# switching endpoint psc1 encoding sonetsdh`	Specifies the switching capability and encoding types for all endpoint TE links used to signal the optical tunnel that is mandatory to set up the GMPLS LSP.

	Command or Action	Purpose
Step 7	**priority** *setup-priority hold-priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# priority 2 2`	Configures setup and reservation priorities for MPLS-TE tunnels.
Step 8	**signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*} **Example:** `RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 2488320`	Configures the bandwidth required for an MPLS TE tunnel. The **signalled-bandwidth** command supports two bandwidth pools (class-types) for **Diff-Serv Aware TE** (DS-TE) feature.
Step 9	**destination** *A.B.C.D* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 24.24.24.24`	Assigns a destination address on the new tunnel. • The destination address is the remote node's MPLS-TE router ID. • The destination address is the merge point between backup and protected tunnels.
Step 10	**direction** [**bidirectional**] **Example:** `RP/0/RP0/CPU0:router(config-if)# direction bidirection`	Configures a bidirectional optical tunnel for GMPLS.
Step 11	**path-option** *path-id* **explicit** {**name** *pathname* | path-number} **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name po4`	Configures the explicit path option and path ID.

	Command or Action	Purpose
Step 12	**path-option protecting** *path-id* **explicit** {**name** *pathname* | *path-number*} **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option protecting 1 explicit name po6`	Configures the path setup option to protect a path.

Step 13	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Forcing Reversion of the LSP

Perform this task to allow a forced reversion of the LSPs, which is only applicable to 1:1 LSP protection.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng path-protection switchover** {*tunnel name* | *number*}
3. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng path-protection switchover** {*tunnel name* | *number*} **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng path-protection switchover 1`	Specifies a manual switchover for path protection for a GMPLS optical LSP. The tunnel ID is configured for a switchover. The **mpls traffic-eng path-protection switchover** command must be issued on both head and tail router of the GMPLS LSP to achieve the complete path switchover at both ends.

Step 3	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config)# end`
	or
	`RP/0/RP0/CPU0:router(config)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Flexible Name-based Tunnel Constraints

To fully configure MPLS-TE Flexible Name-based Tunnel Constraints, you must complete the following high-level tasks in order:

1. Assigning Color Names to Numeric Values, page MPC-166
2. Associating Affinity-Names with TE Links, page MPC-167
3. Associating Affinity Constraints for TE Tunnels, page MPC-168

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).

Note 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*}
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic engineering** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic eng`	Enters MPLS-TE mode.

	Command or Action	Purpose
Step 3	**affinity-map {*affinity name* | *affinity value*}** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# affinity-map red 1`	Enters an affinity name, or a map value, using a color name (repeat this command to assign multiple colors up to a maximum of 64 colors). An affinity color name cannot exceed 64 characters. The value you assign to a color name must be a single digit.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-mpls-te)# end`
	or
	`RP/0/RP0/CPU0:router(config-mpls-te)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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 as described in Assigning Color Names to Numeric Values, page MPC-166.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng interface** *type interface-id*
3. **attribute-names** *color1 color2*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng interface** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic eng interface tunnel-te2`	Enters MPLS-TE mode to configure an interface.
Step 3	**attribute-names** *color1 color2* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-if)# red`	Assigns colors to TE links over the selected interface.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-mpls-te-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-mpls-te-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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

Note 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** *index* {**include** | **include-strict** | **exclude** | **exclude-all**} *color*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure`** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**`interface tunnel-te`** *`tunnel-id`* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te 1`	Selects the a tunnel/interface.

	Command or Action	Purpose
Step 3	**affinity** *index* {**include** | **include-strict** | **exclude** | **exclude-all**} *color* **Example:** `RP/0/RP0/CPU0:router(config-if)# affinity 0 include red`	Enter link attributes for links comprising tunnel. UP TO TEN COLORS. There can be multiple include statements under tunnel configuration as in the above configuration. With the following configuration, a link is eligible for CSPF if it has at least red color OR has at least 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.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# interface POS9/0`	Enters global configuration mode.
Step 2	**router isis** *instance-id* Example: `RP/0/RP0/CPU0:router(config)# router is-is 1`	Enters an IS-IS instance.
Step 3	**net *network-entity-title*** **Example:** `RP/0/RP0/CPU0:router(config-isis)# net 47.0001.0000.0000.0002.00`	Enters an IS-IS network entity title (NET) for the routing process.
Step 4	**address-family {ipv4 | ipv6} {unicast}** **Example:** `RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast`	Enters address family configuration mode for configuring IS-IS routing that use IPv4 and IPv6 address prefixes.
Step 5	**metric-style wide** **Example:** `RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide`	Enter the new-style type, length, and value (TLV) objects.

	Command or Action	Purpose
Step 6	**`mpls traffic-eng`** *`level`* **Example:** `RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng level-1-2`	Enter the required MPLS-TE level or levels.

Step 7	**`end`**
	or
	`commit`
	
	Example:
	`RP/0/RP0/CPU0:router(config-isis-af)# end`
	or
	`RP/0/RP0/CPU0:router(config-isis-af)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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** *type-interface*
4. **area** *area-id*
5. **mpls traffic-eng**
6. **interface** *type interface-id*
7. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 100`	Enters a name that uniquely identifies an OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.
Step 3	**mpls traffic-eng router-id** *type interface-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id Loopback0`	Enters the MPLS interface type. For more information, use the question mark (?) online help function.
Step 4	**area** ***area-id*** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 0`	Enters an OSPF area identifier. The *area-id* argument can be specified as either a decimal value or an IP address.
Step 5	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# area 0`	Enters an OSPF area identifier. The *area-id* argument can be specified as either a decimal value or an IP address.

	Command or Action	Purpose

Step 6	`interface type interface-id`
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar)# interface POS 0/2/0/0`
		Enters an interface ID. For more information, use the question mark (?) online help function.

Step 7	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar)# end`
	or
	`RP/0/RP0/CPU0:router(config-ospf-ar)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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*
3. **index** *number* **next-address loose ipv4 unicast** *A.B.C.D*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# interface POS9/0`	Enters global configuration mode.
Step 2	**explicit-path** *name* **Example:** `RP/0/RP0/CPU0:router(config)# explicit-path interarea1`	Enters a name for the explicit path.
Step 3	**index** *number* **next-address loose ipv4 unicast** *A.B.C.D* **Example:** `RP/0/RP0/CPU0:router(config-expl-path)# index 1 next-address loose ipv4 unicast 10.10.10.10`	Includes a path entry at a specific index.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-expl-path)# end`
	or
	`RP/0/RP0/CPU0:router(config-expl-path)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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** *number*

3. **forwarding-adjacency holdtime** *value*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# interface POS9/0`	Enters global configuration mode.
Step 2	**interface tunnel-te** *number* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te 1`	Enters MPLS-TE interface configuration mode.
Step 3	**forwarding-adjacency holdtime** ***value*** **Example:** `RP/0/RP0/CPU0:router(config-if)# forwarding-adjacency holdtime 60`	Configures forwarding adjacency using an optional specific holdtime value. By default, this value is 0 (milliseconds).

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Unequal Load Balancing

Perform the following tasks to configure unequal load balancing:

- Setting Unequal Load Balancing Parameters, page MPC-177
- Enabling Unequal Load Balancing, page MPC-178

Setting Unequal Load Balancing Parameters

The first step you must take to configure unequal load balancing requires that you set the parameters on each specific interface.

The default load share for tunnels with no explicit configuration is the configured bandwidth.

Note Equal load-sharing occurs if there is no configured bandwidth.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-id*
3. **load-share** *value*
4. **end**
   or
   commit
5. **show mpls traffic-eng tunnels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure`** **Example:** `RP/0/RP0/CPU0:router# config`	Enters global configuration mode.
Step 2	**`interface`** *`type interface-id`* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# interface tunnel-te1.`	Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node. **Note** Only tunnel-te interfaces are permitted.
Step 3	**`load-share`** *`value`* **Example:** `RP/0/RP0/CPU0:router(config-if)# load-share 1000`	Configures the load-sharing parameters for the specified interface.

	Command or Action	Purpose

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 5	**show mpls traffic-eng tunnels** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels`	Verifies the state of unequal load balancing, including bandwidth and load-share values.

Enabling Unequal Load Balancing

This task describes how to enable unequal load balancing. (Quite simply, this is a global switch used to turn unequal load-balancing on or off.)

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. load-share **unequal**
4. **end**
   or
   commit
5. **show mpls traffic-eng tunnels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# config`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters the MPLS-TE configuration mode.
Step 3	**load-share unequal** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# load-share unequal`	Enables unequal load sharing across TE tunnels to the same destination.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-mpls-te)# end`
	or
	`RP/0/RP0/CPU0:router(config-mpls-te)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 5	**show mpls traffic-eng tunnels** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels`	Verifies the state of unequal load balancing, including bandwidth and load-share values.

Configuring a Path Computation Client and Element

Perform the following tasks to configure PCE:

- Configuring a Path Computation Client, page MPC-180
- Configuring a Path Computation Element Address, page MPC-181
- Configuring PCE Parameters, page MPC-182

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** {*number*} **dynamic pce** [*address]*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`configure`** **Example:** `RP/0/RP0/CPU0:router# config`	Enters global configuration mode.
Step 2	**`interface tunnel-te`** *`tunnel-id`* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te 6`	Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.

	Command or Action	Purpose
Step 3	**path-option** {*number*} **dynamic pce** [*address*] **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic pce`	Configures a TE tunnel as a PCC.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring a Path Computation Element Address

Perform this task to configure a PCE address.

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

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **pce address ipv4** *address*
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters the MPLS-TE configuration mode.
Step 3	**pce address ipv4** *address* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce address ipv4 10.1.1.1`	Configures a PCE IPv4 address.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-mpls-te)# end`
	or
	`RP/0/RP0/CPU0:router(config-mpls-te)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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** *address*
5. **pce keepalive** *interval*
6. **pce deadtimer** *value*
7. **pce reoptimize** *value*
8. **pce request-timeout** *value*
9. **pce tolerance keepalive** *value*
10. **end**
    or
    commit
11. **show mpls traffic pce peer** [*address* | **all**]
12. **show mpls traffic-eng pce tunnels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# config`	Enters global configuration mode.
Step 2	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS-TE configuration mode.
Step 3	**pce address ipv4** *address* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce address ipv4 10.1.1.1`	Configures a PCE IPv4 address.
Step 4	**pce peer address ipv4** *address* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce peer address ipv4 10.1.1.1`	(Optional) Configures a static PCE peer address. This step is optional; PCE peers are also discovered dynamically via OSPF/ISIS.
Step 5	**pce keepalive** *interval* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce keepalive 10`	Configures a PCEP keepalive interval. The range is 0 to 255 seconds. When the keepalive interval is 0, the LSR does not send keepalive messages.
Step 6	**pce deadtimer** *value* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce deadtimer 50`	Configures a PCE deadtimer value. The range is 0 to 255 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.

	Command or Action	Purpose
Step 7	**pce reoptimize** *value* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce reoptimize 200`	Configures a periodic reoptimization timer value. The range is 60 to 604800 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.
Step 8	**pce request-timeout** *value* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce request-timeout 10`	Configures a PCE request-timeout. Range is 5 to 100 seconds. PCC/PCE keeps a pending path request only for the request-timeout period.
Step 9	**pce tolerance keepalive** *value* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# pce tolerance keepalive 10`	(Optional) Configures a PCE tolerance keepalive value (which is the minimum acceptable peer proposed keepalive).

Step 10	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 11	**show mpls traffic pce peer [***address* **| all]** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng pce peer`	(Optional) Verifies the PCE peer address and state.
Step 12	**show mpls traffic-eng pce tunnels** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng pce tunnels`	(Optional) Verifies status PCE tunnels.

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 loopback** *number*
4. **signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*}
5. **autoroute announce**
6. **destination** *A.B.C.D*
7. **policy-class** *1 - 7*
8. **path-option** *path-id* **explicit name** *explicit-path-name*
9. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *tunnel-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-te 6`	Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.
Step 3	**ipv4 unnumbered loopback** *number* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback 0`	Assigns a source address so that forwarding can be performed on the new tunnel.
Step 4	**signalled-bandwidth** {*bandwidth* [**class-type** *ct*] | **sub-pool** *bandwidth*} **Example:** `RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10 class-type 1`	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).

	Command or Action	Purpose
Step 5	**autoroute announce** **Example:** `RP/0/RP0/CPU0:router(config-if)# autoroute announce`	Enables messages that notify the neighbor nodes about the routes that are forwarding.
Step 6	**destination** *A.B.C.D* **Example:** `RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125`	Assigns a destination address on the new tunnel. • The destination address is the remote node's MPLS-TE router ID. • The destination address is the merge point between backup and protected tunnels.
Step 7	**policy-class** *1 - 7* **Example:** `RP/0/RP0/CPU0:router(config-if)# policy-class 1`	Configures PBTS to direct traffic into specific TE tunnels.
Step 8	**path-option** *path-id* **explicit name** *explicit-path-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-path`	Sets the path option to explicit with a given name (previously configured) and assigns the path ID.

Step 9	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Cisco MPLS-TE

This section provides the following examples:

Configuring Fast Reroute and SONET APS: Example

When SONET Automatic Protection Switching (APS) is configured on a router, it does not offer protection for tunnels; because of this limitation, fast reroute (FRR) still remains the protection mechanism for MPLS-TE.

When APS is configured in a SONET core network, an alarm might be generated toward a router downstream. If this router is configured with FRR, the hold-off timer must be configured at the SONET level to prevent FRR from being triggered while the core network is performing a restoration. Enter the following commands to configure the delay:

```
RP/0/RP0/CPU0:Route-3(config)# controller sonet 0/6/0/0 delay trigger line 250
RP/0/RP0/CPU0:Route-3(config)# controller sonet 0/6/0/0 path delay trigger 300
```

Building 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 loopback 0
  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
!
(IS-IS)
...
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 Loopback 0
  !
  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 l 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
  route ipv4 192.168.12.52/32 tunnel-te1
  commit
ping 192.168.12.52
show mpls traffic autoroute
!
interface tunnel-te1
  fast-reroute
  mpls traffic-eng interface pos 0/6/0/0
  backup-path tunnel-te 2
  interface tunnel-te2
  backup-bw global-pool 5000
```

```
  ipv4 unnumbered loopback 0
  path-option 1 explicit name backup-path
  destination 192.168.92.125
 commit
show mpls traffic-eng tunnels backup
show mpls traffic-eng fast-reroute database
!
rsvp
  interface pos 0/6/0/0
  bandwidth 100 150 sub-pool 50
  interface tunnel-te1
  bandwidth sub-pool 10
commit
```

Configuring IETF Diff-Serv TE Tunnels: Example

The following example shows how to configure DiffServ-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 model mam
 interface tunnel-te 1bandwidth 10 class-type 1
 commit
```

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

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

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

Configuring GMPLS: Example

This example shows how to set up headend and tailend routers with bidirectional optical unnumbered tunnels using numbered TE links:

Headend Router

```
router ospf roswell
 router-id 11.11.11.11
 nsf cisco
 area 23
 !
```

```
 area 51
  interface Loopback 0
  !
  interface MgmtEth0/0/CPU0/1
  !
  interface POS0/4/0/1
  !
 !
 mpls traffic-eng router-id Loopback 0
 mpls traffic-eng area 51
!

rsvp
 interface POS0/2/0/3
  bandwidth 2000
 !
!
interface tunnel-te1
 ipv4 unnumbered Loopback 0
 switching transit fsc encoding sonetsdh
 switching endpoint psc1 encoding packet
 priority 3 3
 signalled-bandwidth 500
 destination 55.55.55.55
 direction bidirectional
 path-option 1 dynamic
!

mpls traffic-eng
 interface POS0/2/0/3
  flooding-igp ospf roswell area 51
  switching key 1
   encoding packet
   capability psc1
  !
  switching link
   encoding sonetsdh
   capability fsc
  !
  lmp data-link adjacency
   neighbor gmpls5
   remote te-link-id ipv4 10.0.0.5
   remote interface-id unnum 12
   remote switching-capability psc1
  !
 !
 lmp neighbor gmpls5
  ipcc routed
  remote node-id 55.55.55.55
 !
!
```

Tailend Router

```
router ospf roswell
 router-id 55.55.55.55
 nsf cisco
 area 23
 !
 area 51
  interface Loopback 0
  !
  interface MgmtEth0/0/CPU0/1
  !
```

```
  interface POS0/4/0/2
  !
 !
 mpls traffic-eng router-id Loopback 0
 mpls traffic-eng area 51
!

mpls traffic-eng
 interface POS0/2/0/3
  flooding-igp ospf roswell area 51
  switching key 1
   encoding packet
   capability psc1
  !
  switching link
   encoding sonetsdh
   capability fsc
  !
  lmp data-link adjacency
   neighbor gmpls1
   remote te-link-id ipv4 10.0.0.1
   remote interface-id unnum 12
   remote switching-capability psc1
  !
 !
 lmp neighbor gmpls1
  ipcc routed
  remote node-id 11.11.11.11
 !
!
rsvp
 interface POS0/2/0/3
  bandwidth 2000
 !
!
interface tunnel-te1
 ipv4 unnumbered Loopback 0
 passive
 match identifier head_router_hostname_t1
 destination 11.11.11.11
!
```

Configuring 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 Loopback0
 !
 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
 affinity-map purple 10
 affinity-map yellow 8
!
```

Configuring an Interarea Tunnel: Example

The following configuration example shows how to configure a traffic engineering interarea tunnel. Router R1 is the headend for tunnel1, and router R2 (20.0.0.20) is the tailend. Tunnel1 is configured with a path option that is loosely routed through Ra and Rb.

Note Specifying the tunnel tailend in the loosely router path is optional.

```
config
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
next-address loose 192.168.40.40
next-address loose 192.168.60.60
next-address loose 192.168.20.20
```

Note Generally for an interarea tunnel you should configure multiple loosely routed path options that specify different combinations of ABRs (for OSPF) or level-1-2 boundary routers (for IS-IS) to increase the likelihood that the tunnel is successfully signaled. In this simple topology there are no other loosely routed paths.

Configuring 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
```

Configuring Unequal Load Balancing: Example

The following configuration example illustrates unequal load balancing configuration:

```
configure
  interface tunnel-te0
```

```
    destination 1.1.1.1
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
  interface tunnel-te1
    destination 1.1.1.1
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    load-share 5
  interface tunnel-te2
    destination 1.1.1.1
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 5
  interface tunnel-te10
    destination 2.2.2.2
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 10
  interface tunnel-te11
    destination 2.2.2.2
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 10
  interface tunnel-te12
    destination 2.2.2.2
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 20
  interface tunnel-te20
    destination 3.3.3.3
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 10
  interface tunnel-te21
    destination 3.3.3.3
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 10
    load-share 20
  interface tunnel-te30
    destination 4.4.4.4
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 10
    load-share 5
  interface tunnel-te31
    destination 4.4.4.4
    path-option 1 dynamic
    ipv4 unnumbered Loopback0
    signalled-bandwidth 10
    load-share 20
  mpls traffic-eng
    load-share unequal
end
```

Configuring 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 loopback 0
  mpls traffic-eng area 0
  rsvp
  interface pos 0/6/0/0
  bandwidth 100
  commit
```

The following configuration example illustrates PCC configuration:

```
configure
  int 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 loopback 0
  mpls traffic-eng area 0
  rsvp
  interface pos 0/6/0/0
  bandwidth 100
  commit
```

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
```

Additional References

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

Related Documents

Related Topic	Document Title
MPLS-TE commands	*MPLS Traffic Engineering Commands on Cisco IOS XR Software* module in the *Cisco IOS XR MPLS Command Reference*
Cisco CRS-1 router getting started material	*Cisco IOS XR Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*

Standards

Standards[1]	Title
Technical Assistance Center (TAC) home page, containing 30,000 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.	—

1. Not all supported standards are listed.

MIBs

MIBs	MIBs Link
—	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: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
4124	*Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering.* F. Le Faucheur, Ed. June 2005. (Format: TXT=79265 bytes) (Status: PROPOSED STANDARD)
4125	*Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering.* F. Le Faucheur, W. Lai. June 2005. (Format: TXT=22585 bytes) (Status: EXPERIMENTAL)
4127	*Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering.* F. Le Faucheur, Ed. June 2005. (Format: TXT=23694 bytes) (Status: EXPERIMENTAL)

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport