



Implementing MPLS Traffic Engineering

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

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

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



Note

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

Feature History for Implementing MPLS-TE

Release	Modification
Release 2.0	This feature was introduced.
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, GMPLS Protection and Restoration, and GMPLS Path Protection.
Release 3.4.1	Support was added for MPLS-TE and fast reroute link bundling.
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	Modification
Release 3.7.0	Support was added for the following features: <ul style="list-style-type: none"> • 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.
Release 3.8.0	Support was added for the following features: <ul style="list-style-type: none"> • MPLS-TE Automatic Bandwidth. • SRLG (Shared Risk Link Groups). • Policy Based Tunnel Selection (PBTS) IPv6 that includes the Interior Gateway Protocol (IGP) default path.
Release 3.9.0	The Point-to-Multipoint TE feature was added.
Release 4.0.0	Support was added for the following features: <ul style="list-style-type: none"> • AutoTunnel Backup • SRLG (Shared Risk Link Groups)
Release 4.1.0	Support was added for the following features: <ul style="list-style-type: none"> • Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE
Release 4.1.1	The Auto-Tunnel Mesh feature was added.
Release 4.2.0	Support was added for the following features: <ul style="list-style-type: none"> • Soft-Preemption • Path Option Attributes
Release 4.2.1	The Auto-Tunnel Attribute-set feature was added for auto-backup tunnels.
Release 4.2.3	Support was added for the following features: <ul style="list-style-type: none"> • End-to-End TE Path Protection Enhancements — Explicit Path Protection and Co-existence of Path Protection with Fast Reroute • P2MP-TE Inter-area Enhancements
Release 5.2.2	Make-Before-Break feature was added.

Release	Modification
Release 6.1.2	Named Tunnel feature was added.

- [Prerequisites for Implementing Cisco MPLS Traffic Engineering, page 3](#)
- [Information About Implementing MPLS Traffic Engineering, page 3](#)
- [How to Implement Traffic Engineering, page 50](#)
- [Configuration Examples for Cisco MPLS-TE, page 165](#)
- [Additional References, page 191](#)

Prerequisites for Implementing Cisco MPLS Traffic Engineering

These prerequisites are required to implement MPLS TE:

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

Information About Implementing MPLS Traffic Engineering

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

Overview of MPLS Traffic Engineering

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

MPLS-TE is essential for service provider and Internet service provider (ISP) backbones. Such backbones must support a high use of transmission capacity, and the networks must be very resilient so that they can withstand link or node failures. MPLS-TE provides an integrated approach to traffic engineering. With MPLS, traffic engineering capabilities are integrated into Layer 3, which optimizes the routing of IP traffic, given the constraints imposed by backbone capacity and topology.

Related Topics

[Configuring Forwarding over the MPLS-TE Tunnel, on page 55](#)

Benefits of MPLS Traffic Engineering

MPLS-TE enables ISPs to route network traffic to offer the best service to their users in terms of throughput and delay. By making the service provider more efficient, traffic engineering reduces the cost of the network.

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

How MPLS-TE Works

MPLS-TE automatically establishes and maintains label switched paths (LSPs) across the backbone by using RSVP. The path that an LSP uses is determined by the LSP resource requirements and network resources, such as bandwidth. Available resources are flooded by means of extensions to a link-state-based Interior Gateway Protocol (IGP).

MPLS-TE tunnels are calculated at the LSP headend router, based on a fit between the required and available resources (constraint-based routing). The IGP automatically routes the traffic to these LSPs.

Typically, a packet crossing the MPLS-TE backbone travels on a single LSP that connects the ingress point to the egress point. MPLS-TE is built on these mechanisms:

Tunnel interfaces

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

MPLS-TE path calculation module

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

RSVP with TE extensions

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

MPLS-TE link management module

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

Link-state IGP (Intermediate System-to-Intermediate System [IS-IS] or Open Shortest Path First [OSPF]—each with traffic engineering extensions)

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

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

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

Label switching forwarding

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

One approach to engineering a backbone is to define a mesh of tunnels from every ingress device to every egress device. The MPLS-TE path calculation and signaling modules determine the path taken by the LSPs for these tunnels, subject to resource availability and the dynamic state of the network.

The IGP (operating at an ingress device) determines which traffic should go to which egress device, and steers that traffic into the tunnel from ingress to egress. A flow from an ingress device to an egress device might be so large that it cannot fit over a single link, so it cannot be carried by a single tunnel. In this case, multiple tunnels between a given ingress and egress can be configured, and the flow is distributed using load sharing among the tunnels.

Related Topics

[Building MPLS-TE Topology, on page 50](#)

[Creating an MPLS-TE Tunnel, on page 53](#)

[Build MPLS-TE Topology and Tunnels: Example, on page 166](#)

MPLS Traffic Engineering

Multiprotocol Label Switching (MPLS) is an Internet Engineering Task Force (IETF)-specified framework that provides efficient designation, routing, forwarding, and switching of traffic flows through the network.

TE is the process of adjusting bandwidth allocations to ensure that enough bandwidth is available for high-priority traffic.

In MPLS TE, the upstream router creates a network tunnel for a particular traffic stream and sets the bandwidth available for that tunnel.

Backup AutoTunnels

The MPLS Traffic Engineering AutoTunnel Backup feature enables a router to dynamically build backup tunnels on the interfaces that are configured with MPLS TE tunnels. This feature enables a router to dynamically build backup tunnels when they are needed. This prevents you from having to build MPLS TE tunnels **statically**.

The MPLS Traffic Engineering (TE)—AutoTunnel Backup feature has these benefits:

- Backup tunnels are built automatically, eliminating the need for users to preconfigure each backup tunnel and then assign the backup tunnel to the protected interface.
- Protection is expanded—FRR does not protect IP traffic that is not using the TE tunnel or Label Distribution Protocol (LDP) labels that are not using the TE tunnel.

This feature protects against these failures:

- **P2P Tunnel NHOP protection**—Protects against link failure for the associated P2P protected tunnel

- **P2P Tunnel NNHOP protection**—Protects against node failure for the associated P2P protected tunnel
- **P2MP Tunnel NHOP protection**—Protects against link failure for the associated P2MP protected tunnel

Related Topics

[Enabling an AutoTunnel Backup, on page 60](#)

[Removing an AutoTunnel Backup, on page 61](#)

[Establishing MPLS Backup AutoTunnels to Protect Fast Reroutable TE LSPs, on page 62](#)

[Establishing Next-Hop Tunnels with Link Protection, on page 63](#)

[Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 179](#)

AutoTunnel Attribute-set

This feature supports auto-tunnels configuration using attribute templates, known as attribute-set. The TE attribute-set template that specifies a set of TE tunnel attributes, is locally configured at the head-end of auto-tunnels. The control plane triggers the automatic provisioning of a corresponding TE tunnel, whose characteristics are specified in the respective attribute-set.

Currently, auto-tunnel backups are created with the default values of all tunnel attributes. To support configurable attributes for auto-tunnel backup, it is required to configure attribute-set and assign it to the backup tunnels. The attribute-set consists of a set of tunnel attributes such as priority, affinity, signaled bandwidth, logging, policy-class, record-route and so on.

The following rules (consistent across all auto-tunnels) apply while configuring the attribute-set:

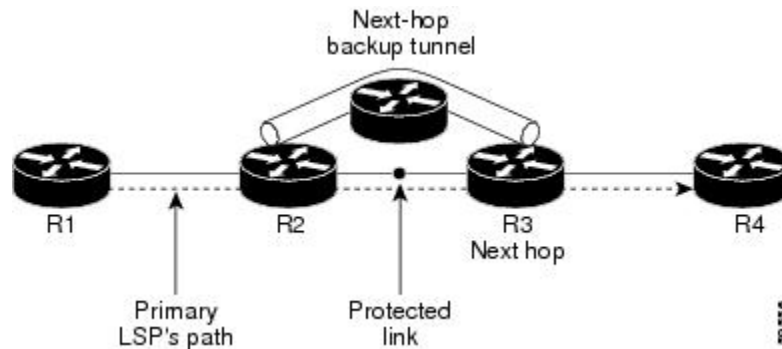
- If no attribute-set template is defined, the auto-tunnels is created using default attribute values.
- If an attribute-set is defined and the attribute-set template is already configured, the auto-tunnel is created using the attributes specified in the associated attribute-set.
- If an attribute-set is assigned, but it is not defined or configured, auto-tunnel is not created.
- Any number of attribute-sets can be configured with same attribute settings.
- Empty tunnel attribute implies all parameters have default values.
- When specific attribute is not specified in the attribute-set, a default value for that attribute is used.

Link Protection

The backup tunnels that bypass only a single link of the LSP path provide link protection. They protect LSPs, if a link along their path fails, by rerouting the LSP traffic to the next hop, thereby bypassing the failed link. These are referred to as NHOP backup tunnels because they terminate at the LSP's next hop beyond the point of failure.

This figure illustrates link protection.

Figure 1: Link Protection

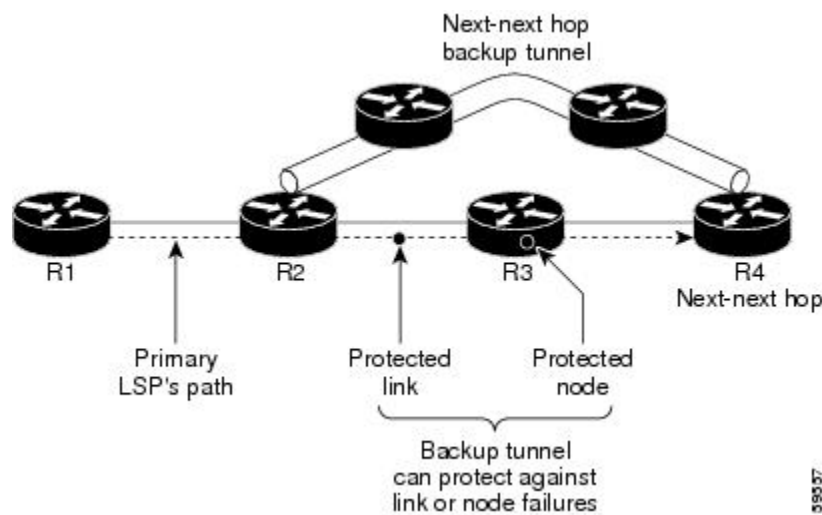


Node Protection

The backup tunnels that bypass next-hop nodes along LSP paths are called NNHOP backup tunnels because they terminate at the node following the next-hop node of the LSPs, thereby bypassing the next-hop node. They protect LSPs by enabling the node upstream of a link or node failure to reroute the LSPs and their traffic around a node failure to the next-hop node. NNHOP backup tunnels also provide protection from link failures because they bypass the failed link and the node.

This figure illustrates node protection.

Figure 2: Node Protection



Backup AutoTunnel Assignment

At the head or mid points of a tunnel, the backup assignment finds an appropriate backup to protect a given primary tunnel for FRR protection.

The backup assignment logic is performed differently based on the type of backup configured on the output interface used by the primary tunnel. Configured backup types are:

- Static Backup
- AutoTunnel Backup
- No Backup (In this case no backup assignment is performed and the tunnels is unprotected.)



Note Static backup and Backup AutoTunnel cannot exist together on the same interface or link.



Note Node protection is always preferred over link protection in the Backup AutoTunnel assignment.

In order that the Backup AutoTunnel feature operates successfully, the following configuration must be applied at global configuration level:

```
ipv4 unnumbered mpls traffic-eng Loopback 0
```



Note The Loopback 0 is used as router ID.

Explicit Paths

Explicit paths are used to create backup autotunnels as follows:

For NHOP Backup Autotunnels:

- NHOP excludes the protected link's local IP address.
- NHOP excludes the protected link's remote IP address.
- The explicit-path name is `_autob_nhop_tunnelxxx`, where xxx matches the dynamically created backup tunnel ID.

For NNHOP Backup Autotunnels:

- NNHOP excludes the protected link's local IP address.
- NNHOP excludes the protected link's remote IP address (link address on next hop).
- NNHOP excludes the NHOP router ID of the protected primary tunnel next hop.
- The explicit-path name is `_autob_nnhop_tunnelxxx`, where xxx matches the dynamically created backup tunnel ID.

Periodic Backup Promotion

The periodic backup promotion attempts to find and assign a better backup for primary tunnels that are already protected.

With AutoTunnel Backup, the only scenario where two backups can protect the same primary tunnel is when both an NHOP and NNHOP AutoTunnel Backups get created. The backup assignment takes place as soon as the NHOP and NNHOP backup tunnels come up. So, there is no need to wait for the periodic promotion.

Although there is no exception for AutoTunnel Backups, periodic backup promotion has no impact on primary tunnels protected by AutoTunnel Backup.

One exception is when a manual promotion is triggered by the user using the **mpls traffic-eng fast-reroute timers promotion** command, where backup assignment or promotion is triggered on all FRR protected primary tunnels—even unprotected ones. This may trigger the immediate creation of some AutoTunnel Backup, if the command is entered within the time window when a required AutoTunnel Backup has not been yet created.

You can configure the periodic promotion timer using the global configuration **mpls traffic-eng fast-reroute timers promotion** *sec* command. The range is 0 to 604800 seconds.

**Note**

A value of 0 for the periodic promotion timer disables the periodic promotion.

Protocol-Based CLI

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

Differentiated Services Traffic Engineering

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

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

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

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

Related Topics

- [Confirming DiffServ-TE Bandwidth](#)
- [Bandwidth Configuration \(MAM\): Example](#)
- [Bandwidth Configuration \(RDM\): Example](#)

Prestandard DS-TE Mode

Prestandard DS-TE uses the Cisco proprietary mechanisms for RSVP signaling and IGP advertisements. This DS-TE mode does not interoperate with third-party vendor equipment. Note that prestandard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces.

Prestandard Diff-Serve TE mode supports a single bandwidth constraint model a Russian Doll Model (RDM) with two bandwidth pools: global-pool and sub-pool.

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

Related Topics

[Configuring a Prestandard DS-TE Tunnel, on page 65](#)

[Configure IETF DS-TE Tunnels: Example, on page 167](#)

IETF DS-TE Mode

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

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

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

Bandwidth Constraint Models

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

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



Note

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

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

Maximum Allocation Bandwidth Constraint Model

The MAM constraint model has the following characteristics:

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

Related Topics

[Configuring an IETF DS-TE Tunnel Using MAM, on page 69](#)

Russian Doll Bandwidth Constraint Model

The RDM constraint model has these characteristics:

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

**Note**

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

Related Topics

[Configuring an IETF DS-TE Tunnel Using RDM, on page 67](#)

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.

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

Table 1: TE Classes and Priority

TE Class	Class Type	Priority
0	0	7
1	1	7
2	Unused	—
3	Unused	—
4	0	0

TE Class	Class Type	Priority
5	1	0
6	Unused	—
7	Unused	—

Flooding

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

Flooding Triggers

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

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

Flooding Thresholds

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

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

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



Note

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

Fast Reroute

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

FRR (link or node) is supported over sub-pool tunnels the same way as for regular TE tunnels. In particular, when link protection is activated for a given link, TE tunnels eligible for FRR are redirected into the protection LSP, regardless of whether they are sub-pool or global pool tunnels.

**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 these requirements for the backup tunnel path:

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

**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.

Related Topics

[Protecting MPLS Tunnels with Fast Reroute, on page 57](#)

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 for the Cisco CRS-1 Router*.

MPLS-TE and Fast Reroute over Link Bundles

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

These 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, and FastEthernet, 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 avoidance feature allows network administrators to prevent RSVP-TE label switched paths (LSPs) from being disabled, when a router in that path has its Intermediate System-to-Intermediate System (IS-IS) overload bit set.

The IS-IS overload bit avoidance feature is activated using this command:

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

The IS-IS overload bit avoidance feature is deactivated using the **no** form of this command:

```
no mpls traffic-eng path-selection ignore overload
```

When the IS-IS overload bit avoidance feature is activated, all nodes, including head nodes, mid nodes, and tail nodes, with the overload bit set, are ignored. This means that they are still available for use with RSVP-TE label switched paths (LSPs). This feature enables you to include an overloaded node in CSPF.

Enhancement Options of IS-IS OLA

You can restrict configuring IS-IS overload bit avoidance with the following enhancement options:

- **path-selection ignore overload head**

The tunnels stay up if **set-overload-bit** is set by IS-IS on the head router. Ignores overload during CSPF for LSPs originating from an overloaded node. In all other cases (mid, tail, or both), the tunnel stays down.

- **path-selection ignore overload mid**

The tunnels stay up if **set-overload-bit** is set by IS-IS on the mid router. Ignores overload during CSPF for LSPs transiting from an overloaded node. In all other cases (head, tail, or both), the tunnel stays down.

- **path-selection ignore overload tail**

The tunnels stay up if **set-overload-bit** is set by IS-IS on the tail router. Ignores overload during CSPF for LSPs terminating at an overloaded node. In all other cases (head, mid, or both), the tunnel stays down.

- **path-selection ignore overload**

The tunnels stay up irrespective of on which router the **set-overload-bit** is set by IS-IS.



Note When you do not select any of the options, including head nodes, mid nodes, and tail nodes, you get a behavior that is applicable to all nodes. This behavior is backward compatible in nature.

For more information related to IS-IS overload avoidance related commands, see *Cisco IOS XR MPLS Command Reference for the Cisco CRS Router*.

Related Topics

[Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE, on page 73](#)

[Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 168](#)

DWDM Transponder Integration

A GMPLS UNI based solution preserves all the advantages of the integration of the DWDM transponder into the router blade. These advantages include:

- improved CAPEX and OPEX models
- component, space and power savings
- improved IP availability through pro-active protection.

GMPLS Benefits

GMPLS bridges the 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 or 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 or AS-Verbatim
- BGP4 or MPLS
- Restoration—Dynamic path computation
- Control channel manager
- Link summary
- Protection and restoration

Related Topics

[Configuring Router IDs, on page 74](#)

[Configuring OSPF over IPCC, on page 76](#)

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 and wavelength. 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.
- Router that runs Cisco IOS XR software.
- Installation of the Cisco IOS XR softwaremini-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.

Related Topics

- [Assigning Color Names to Numeric Values, on page 97](#)
- [Associating Affinity-Names with TE Links, on page 98](#)
- [Associating Affinity Constraints for TE Tunnels, on page 99](#)
- [Configure Flexible Name-based Tunnel Constraints: Example, on page 170](#)

MPLS Traffic Engineering Interarea Tunneling

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

- [Interarea Support, on page 18](#)
- [Multiarea Support, on page 19](#)
- [Loose Hop Expansion, on page 20](#)
- [Loose Hop Reoptimization, on page 20](#)
- [Fast Reroute Node Protection, on page 20](#)

Interarea Support

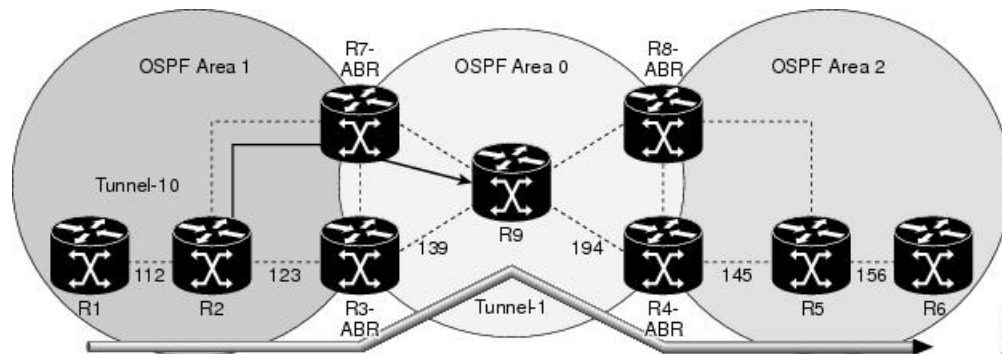
The MPLS-TE interarea tunneling feature allows you to establish P2P and P2MP TE tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thereby eliminating the requirement that headend and tailend routers reside in a single area.

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.

This figure shows a typical interarea TE network.

Figure 3: Interarea (OSPF) TE Network Diagram



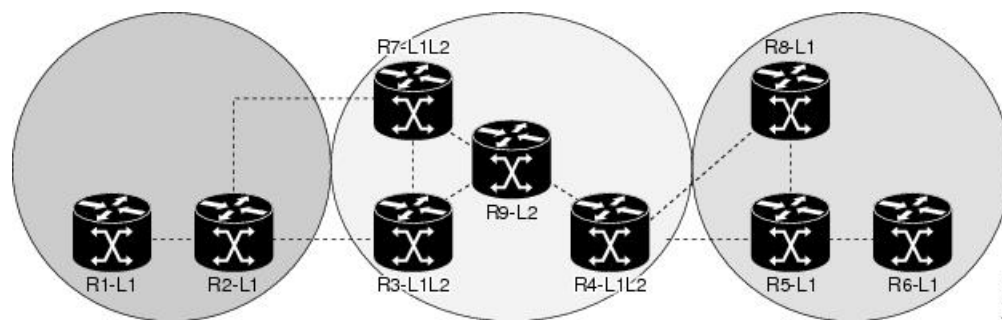
Multiarea Support

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

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

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

Figure 4: Interlevel (IS-IS) TE Network



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



Note

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

Loose Hop Expansion

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

Interarea MPLS-TE allows you to configure an interarea traffic engineering (TE) label switched path (LSP) by specifying a loose source route of ABRs along the path. It is 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.

For P2MP-TE tunnels, ABRs support loose hop ERO expansion to find path to the next ABR until it reaches to the tail-end LSR, without introducing remerge.

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

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

Loose Hop Reoptimization

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

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

ABR Node Protection

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

Fast Reroute Node Protection

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

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

Related Topics

[Protecting MPLS Tunnels with Fast Reroute, on page 57](#)

Make-Before-Break

The MPLS TE Make-Before-Break (MBB) explicit path and path option feature allows tunnels whose explicit paths or path options are modified to be reoptimized without losing any data. An explicit path or a path option modification is entirely configuration driven. Any change to an in-use path option or an in-use explicit path of a tunnel triggers the MBB procedure.

MBB lets the LSP hold on to the existing resources until the new path is successfully established and traffic has been directed over to the new LSP before the original LSP is torn down. This ensures that no data packets are lost during the transition to the new LSP.

With this feature the flapping of tunnels whose explicit paths or path options are modified, is avoided. This feature is enabled by default.

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 IGP to compute the SPF even if they are not the head end of any TE tunnels.

Related Topics

[Configuring MPLS-TE Forwarding Adjacency, on page 103](#)

[Configure Forwarding Adjacency: Example, on page 173](#)

MPLS-TE Forwarding Adjacency Restrictions

The MPLS-TE Forwarding Adjacency feature has these restrictions:

- Using the MPLS-TE Forwarding Adjacency increases the size of the IGP database by advertising a TE tunnel as a link.
- The MPLS-TE Forwarding Adjacency is supported by Intermediate System-to-Intermediate System (IS-IS).
- When the MPLS-TE Forwarding Adjacency is enabled on a TE tunnel, the link is advertised in the IGP network as a Type-Length-Value (TLV) 22 without any TE sub-TLV.

- MPLS-TE forwarding adjacency tunnels must be configured bidirectionally.
- Multicast intact is not supported with MPLS-TE Forwarding Adjacency.

MPLS-TE Forwarding Adjacency Prerequisites

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

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

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 through the MPLS Label Switching Database (LSD) to the Forwarding Information Base (FIB).



Note

Load share values are renormalized 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.

Related Topics

[Setting Unequal Load Balancing Parameters, on page 104](#)

[Enabling Unequal Load Balancing, on page 105](#)

[Configure Unequal Load Balancing: Example, on page 173](#)

Path Computation Element

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

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

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

Path Computation Element (PCE)

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

Path Computation Client (PCC)

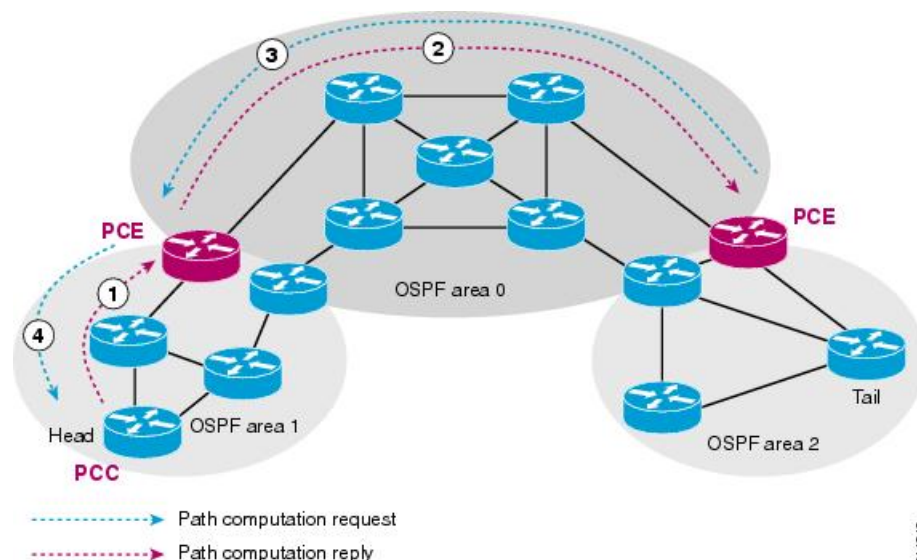
Represents a software module running on a router that is capable of sending and receiving path computation requests and responses to and from PCEs. The PCC is typically an LSR (Label Switching Router).

PCC-PCE communication protocol (PCEP)

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

This figure shows a typical PCE implementation.

Figure 5: 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

Related Topics

- [Configuring a Path Computation Client, on page 106](#)
- [Configuring a Path Computation Element Address, on page 107](#)
- [Configuring PCE Parameters, on page 108](#)
- [Configure PCE: Example, on page 174](#)

Policy-Based Tunnel Selection

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

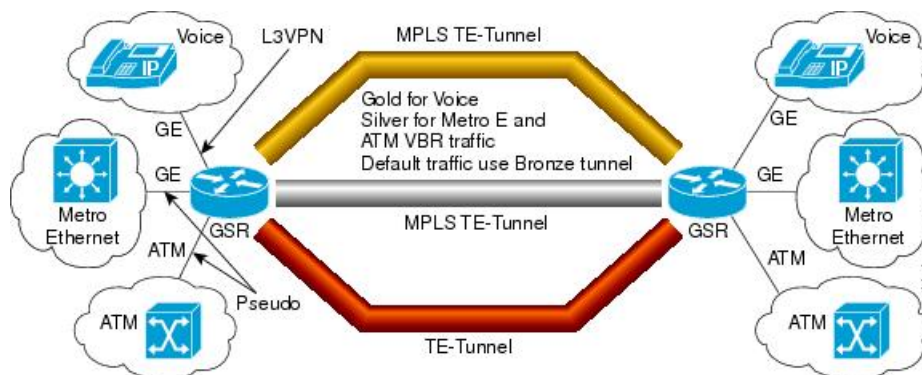
Policy-Based Tunnel Selection

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

PBTS works by selecting tunnels based on the classification criteria of the incoming packets, which are based on the IP precedence, experimental (EXP), or type of service (ToS) field in the packet. When there are no paths with a default class configured, this traffic is forwarded using the paths with the lowest class value. PBTS supports up to seven (exp 1 - 7) EXP values associated with a single TE tunnel.

This figure illustrates a PBTS implementation.

Figure 6: Policy-Based Tunnel Selection Implementation



Related Topics

- [Configuring Policy-based Tunnel Selection, on page 111](#)
- [Configure Policy-based Tunnel Selection: Example, on page 175](#)

Policy-Based Tunnel Selection Functions

The following PBTS functions are supported:

- IPv4 traffic arrives unlabeled on the VRF interface and the non-VRF interface.
- MPLS traffic is supported on the VRF interface and the non-VRF interface.
- Load balancing across multiple TE tunnels with the same traffic class attribute is supported.
- Selected TE tunnels are used to service the lowest tunnel class as default tunnels.
- LDP over TE tunnel and single-hop TE tunnel are supported.
- Both Interior Gateway Protocol (IGP) and Label Distribution Protocol (LDP) paths are used as the default path for all traffic that belongs to a class that is not configured on the TE tunnels.
- According to the quality-of-service (QoS) policy, tunnel selection is based on the outgoing experimental (EXP) value and the remarked EXP value.
- L2VPN preferred path selection lets traffic be directed to a particular TE tunnel.
- IPv6 traffic for both 6VPE and 6PE scenarios are supported.

Related Topics

[Configuring Policy-based Tunnel Selection, on page 111](#)

[Configure Policy-based Tunnel Selection: Example, on page 175](#)

PBTS Restrictions

When implementing PBTS, the following restrictions are listed:

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

Path Protection

Path protection provides an end-to-end failure recovery mechanism (that is, a full path protection) for MPLS-TE tunnels. A secondary Label Switched Path (LSP) is established, in advance, to provide failure protection for the protected LSP that is carrying a tunnel's TE traffic. When there is a failure on the protected LSP, the source router immediately enables the secondary LSP to temporarily carry the tunnel's traffic. If there is a failure on the secondary LSP, the tunnel no longer has path protection until the failure along the secondary path is cleared. Path protection can be used within a single area (OSPF or IS-IS), external BGP [eBGP], and static routes.

The failure detection mechanisms triggers a switchover to a secondary tunnel by:

- Path error or resv-tear from Resource Reservation Protocol (RSVP) signaling
- Notification from the Bidirectional Forwarding Detection (BFD) protocol that a neighbor is lost

- Notification from the Interior Gateway Protocol (IGP) that the adjacency is down
- Local teardown of the protected tunnel's LSP due to preemption in order to signal higher priority LSPs, a Packet over SONET (POS) alarm, online insertion and removal (OIR), and so on

An alternate recovery mechanism is Fast Reroute (FRR), which protects MPLS-TE LSPs only from link and node failures, by locally repairing the LSPs at the point of failure. Co-existence of FRR and path protection is supported; this means FRR and path-protection can be configured on the same tunnel at the same time.

Although not as fast as link or node protection, presignaling a secondary LSP is faster than configuring a secondary primary path option, or allowing the tunnel's source router to dynamically recalculate a path. The actual recovery time is topology-dependent, and affected by delay factors such as propagation delay or switch fabric latency.

Related Topics

[Configure Tunnels for Path Protection: Example, on page 175](#)

Pre-requisites for Path Protection

These are the pre-requisites for enabling path protection:

- Ensure that your network supports MPLS-TE, Cisco Express Forwarding, and Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).
- Enable MPLS.
- Configure TE on the routers.
- Configure a TE tunnel with a dynamic path option by using the **path-option** command with the **dynamic** keyword.

Related Topics

[Configure Tunnels for Path Protection: Example, on page 175](#)

Restrictions for Path Protection

- Only Point-to-Point (P2P) tunnels are supported.
- Point-to-Multipoint (P2MP) TE tunnels are not supported.
- A maximum of one standby LSP is supported.
- There can be only one secondary path for each dynamic path option.
- Explicit path option can be configured for the path protected TE with the secondary path option as dynamic.
- A maximum number of path protected tunnel TE heads is 2000.
- A maximum number of TE tunnel heads is equal to 4000.
- When path protection is enabled for a tunnel, and the primary label switched path (LSP) is not assigned a backup tunnel, but the standby LSP is assigned fast-reroute (FRR), the MPLS TE FRR protected value displayed is different from the Cisco express forwarding (CEF) fast-reroute value.

- Inter-area is not supported for path protection.

Related Topics

[Configure Tunnels for Path Protection: Example, on page 175](#)

Restrictions for Explicit Path Protection

Explicit paths are used to create backup autotunnels. Explicit path protection provides a recovery mechanism to protect explicit paths for MPLS-TE tunnels. These restrictions are listed to protect an explicit path:

- Only one explicit protecting path is supported per path-option.
- Link or node path diversity is not ensured for explicit protecting paths.
- An explicit protecting path cannot protect a dynamic path option.
- All options such as **verbatim**, **lockdown** are supported for the protecting path as long as it's explicit.
- An explicit path cannot be protected by its own path option level.
- An explicit path can be protected by a path option level that references the same explicit path name or identifier, because it is considered another path-option.
- Enhanced path protection is not supported.

Related Topics

[Configure Tunnels for Path Protection: Example, on page 175](#)

Co-existence of Path Protection with Fast Reroute

Path protection and FRR can be configured on the same tunnel at the same time. The co-existence of path protection and FRR on the same tunnel provides these benefits:

- Protection is expanded — having an FRR protected tunnel that is also path-protected ensures that failures of non-protected links on the primary path are handled more efficiently by a quick switch-over to the pre-sigaled standby LSP.
- Quick and effective re-optimization — having a pre-computed standby LSP allows the system to minimize re-optimization LSP path calculation and signaling, by simply switching over to the pre-sigaled standby LSP. Effectively, path protection switch over replaces the post-FRR LSP down event re-optimization.
- Total time on backup is reduced — handling FRR failure using a path protection switch over reduces total time on backup because the traffic is diverted from the backup to the standby, as soon as the head-end receives the FRR LSP down notification, without having to wait for a re-optimization LSP.

MPLS-TE Automatic Bandwidth

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

These topics provide information about MPLS-TE automatic bandwidth:

MPLS-TE Automatic Bandwidth Overview

MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every head-end. MPLS-TE monitors the traffic rate on a tunnel interface. Periodically, MPLS-TE resizes the bandwidth on the tunnel interface to align it closely with the traffic in the tunnel. MPLS-TE automatic bandwidth can perform these functions:

- Monitors periodic polling of the tunnel output rate
- Resizes the tunnel bandwidth by adjusting the highest rate observed during a given period

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

This table lists the automatic bandwidth functions.

Table 2: Automatic Bandwidth Variables

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

The output rate on a tunnel is collected at regular intervals that are configured by using the **application** command in MPLS-TE auto bandwidth interface configuration mode. When the application period timer

expires, and when the difference between the measured and the current bandwidth exceeds the adjustment threshold, the tunnel is reoptimized. Then, the bandwidth samples are cleared to record the new largest output rate at the next interval.

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

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

**Note**

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

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

Related Topics

[Configuring the Collection Frequency, on page 112](#)

[Configuring the Automatic Bandwidth Functions, on page 114](#)

[Configure Automatic Bandwidth: Example, on page 176](#)

Adjustment Threshold

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

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

Overflow Detection

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

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

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

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

Underflow Detection

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

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

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

Restrictions for MPLS-TE Automatic Bandwidth

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

- Tunnel is in a fast reroute (FRR) backup, active, or path protect active state. This occurs because of the assumption that protection is a temporary state, and there is no need to reserve the bandwidth on a backup tunnel. You should prevent taking away the bandwidth from other primary or backup tunnels.
- Reoptimization fails to occur during a lockdown. In this case, the automatic bandwidth does not update the bandwidth unless the bandwidth application is manually triggered by using the **mpls traffic-eng auto-bw apply** command in EXEC mode.

Related Topics

[Forcing the Current Application Period to Expire Immediately, on page 114](#)

Point-to-Multipoint Traffic-Engineering

Point-to-Multipoint Traffic-Engineering Overview

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

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

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

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

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

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

Source

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

Mid-point

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

Receiver, Leaf, and Destination

Specifies the node in which LSP signaling ends.

Branch Point

Specifies the node in which packet replication is performed.

Source-to-Leaf (S2L) SubLSP

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

Point-to-Multipoint Traffic-Engineering Features

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



Note Per-S2L reoptimization is not supported.

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

- Static mapping for multicast groups to the P2MP-TE tunnel is required .

Point-to-Multipoint Traffic-Engineering Benefits

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

Related Topics

[Enabling Multicast Routing on the Router, on page 128](#)
[Configure Point-to-Multipoint for the Source: Example, on page 186](#)
[Configure the Point-to-Multipoint Solution: Example, on page 188](#)
[Disabling Destinations, on page 134](#)
[Disable a Destination: Example, on page 187](#)
[Logging Per Destinations for Point-to-Multipoint , on page 136](#)
[Configure the Point-to-Multipoint Tunnel: Example, on page 187](#)
[Configure the Point-to-Multipoint Solution: Example, on page 188](#)
[Point-to-Multipoint RSVP-TE , on page 32](#)

Point-to-Multipoint RSVP-TE

RSVP-TE signals a P2MP tunnel base that is based on a manual configuration. If all Source-to-Leaf (S2L)s use an explicit path, the P2MP tunnel creates a static tree that follows a predefined path based on a constraint such as a deterministic Label Switched Path (LSP). If the S2L uses a dynamic path, RSVP-TE creates a P2MP tunnel base on the best path in the RSVP-TE topology. RSVP-TE supports bandwidth reservation for constraint-based routing.

When an explicit path option is used, specify both the local and peer IP addresses in the explicit path option, provided the link is a GigabitEthernet or a TenGigE based interface. For point-to-point links like POS or bundle POS, it is sufficient to mention the remote or peer IP address in the explicit path option.

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

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

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

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

Related Topics

[Configuring the Static Group for the Point-to-Multipoint Interface, on page 130](#)

[Configure Point-to-Multipoint for the Source: Example, on page 186](#)

[Configure the Point-to-Multipoint Solution: Example, on page 188](#)

[Point-to-Multipoint Fast Reroute, on page 33](#)

Point-to-Multipoint Fast Reroute

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

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

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

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

Related Topics

[Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)

[Point-to-Multipoint RSVP-TE, on page 32](#)

Point-to-Multipoint Label Switch Path

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



Note

Grafting and pruning operate on a per destination basis.

The Opaque Value contains the stream information that uniquely identifies the tree to the root. To receive label switched multicast packets, the Egress Provider Edge (PE) indicates to the upstream router (the next hop closest to the root) which label it uses for the multicast source by applying the label mapping message.

The upstream router does not need to have any knowledge of the source; it needs only the received FEC to identify the correct P2MP LSP. If the upstream router does not have any FEC state, it creates it and installs the assigned downstream outgoing label into the label forwarding table. If the upstream router is not the root of the tree, it must forward the label mapping message to the next hop upstream. This process is repeated hop-by-hop until the root is reached.

By using downstream allocation, the router that wants to receive the multicast traffic assigns the label for it. The label request, which is sent to the upstream router, is similar to an unsolicited label mapping (that is, the upstream does not request it). The upstream router that receives that label mapping uses the specific label to send multicast packets downstream to the receiver. The advantage is that the router, which allocates the labels,

does not get into a situation where it has the same label for two different multicast sources. This is because it manages its own label space allocation locally.

Path Option for Point-to-Multipoint RSVP-TE

P2MP tunnels are signaled by using the dynamic and explicit path-options in an IGP intra area. InterArea cases for P2MP tunnels are signaled by the verbatim path option.

Path options for P2MP tunnels are individually configured for each sub-LSP. Only one path option per sub-LSP (destination) is allowed. You can choose whether the corresponding sub-LSP is dynamically or explicitly routed. For the explicit option, you can configure the verbatim path option to bypass the topology database lookup and verification for the specified destination.

Both dynamic and explicit path options are supported on a per destination basis by using the **path-option (P2MP-TE)** command. In addition, you can combine both path options.

Explicit Path Option

Configures the intermediate hops that are traversed by a sub-LSP going from the TE source to the egress MPLS node. Although an explicit path configuration enables granular control sub-LSP paths in an MPLS network, multiple explicit paths are configured for specific network topologies with a limited number of (equal cost) links or paths.

Dynamic Path Option

Computes the IGP path of a P2MP tree sub-LSP that is based on the OSPF and ISIS algorithm. The TE source is dynamically calculated based on the IGP topology.



Note

Dynamic path option can only compute fully-diverse standby paths. While, explicit path option supports partially diverse standby paths as well.

Dynamic Path Calculation Requirements

Dynamic path calculation for each sub-LSP uses the same path parameters as those for the path calculation of regular point-to-point TE tunnels. As part of the sub-LSP path calculation, the link resource (bandwidth) is included, which is flooded throughout the MPLS network through the existing RSVP-TE extensions to OSPF and ISIS. Instead of dynamic calculated paths, explicit paths are also configured for one or more sub-LSPs that are associated with the P2MP-TE tunnel.

- OSPF or ISIS are used for each destination.
- TE topology and tunnel constraints are used to input the path calculation.
- Tunnel constraints such as affinity, bandwidth, and priorities are used for all destinations in a tunnel.
- Path calculation yields an explicit route to each destination.

Static Path Calculation Requirements

The static path calculation does not require any new extensions to IGP to advertise link availability.

- Explicit path is required for every destination.

- Offline path calculation is used.
- TE topology database is not needed.
- If the topology changes, reoptimization is not required.

Related Topics

[Configuring Destinations for the Tunnel Interface](#), on page 131

[Configure the Point-to-Multipoint Tunnel: Example](#), on page 187

[Configure the Point-to-Multipoint Solution: Example](#), on page 188

[Point-to-Multipoint Traffic-Engineering Overview](#), on page 30

[Point-to-Multipoint RSVP-TE](#), on page 32

MPLS Traffic Engineering Shared Risk Link Groups

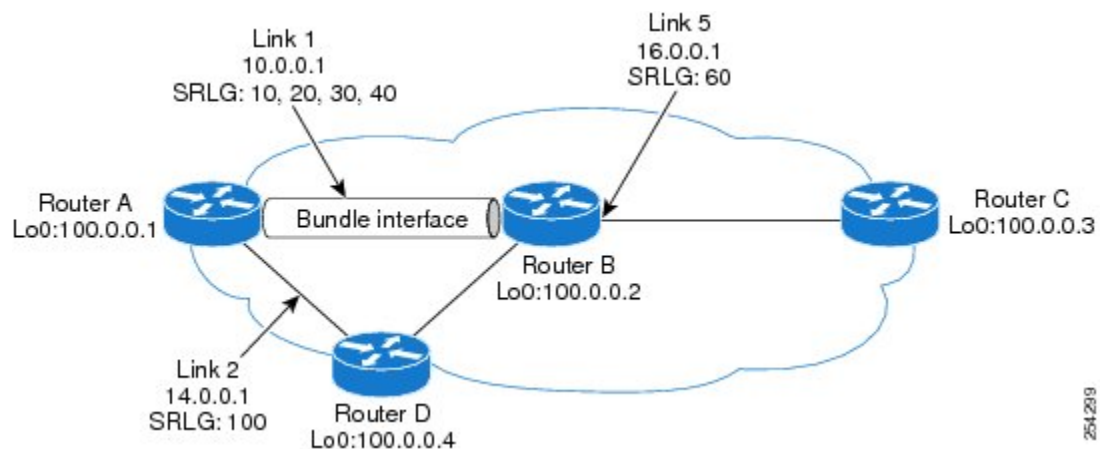
Shared Risk Link Groups (SRLG) in MPLS traffic engineering refer to situations in which links in a network share a common fiber (or a common physical attribute). These links have a shared risk, and that is when one link fails, other links in the group might fail too.

OSPF and Intermediate System-to-Intermediate System (IS-IS) flood the SRLG value information (including other TE link attributes such as bandwidth availability and affinity) using a sub-type length value (sub-TLV), so that all routers in the network have the SRLG information for each link.

To activate the SRLG feature, configure the SRLG value of each link that has a shared risk with another link. A maximum of 30 SRLGs per interface is allowed. You can configure this feature on multiple interfaces including the bundle interface.

[Figure 7: Shared Risk Link Group](#) illustrates the MPLS TE SRLG values configured on the bundle interface.

Figure 7: Shared Risk Link Group



Related Topics

[Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link](#), on page 117

[Creating an Explicit Path With Exclude SRLG](#), on page 119

[Using Explicit Path With Exclude SRLG, on page 120](#)
[Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)
[Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)
[Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Explicit Path

The Explicit Path configuration allows you to configure the explicit path. An IP explicit path is a list of IP addresses, each representing a node or link in the explicit path.

The MPLS Traffic Engineering (TE)—IP Explicit Address Exclusion feature provides a means to exclude a link or node from the path for an Multiprotocol Label Switching (MPLS) TE label-switched path (LSP).

This feature is enabled through the **explicit-path** command that allows you to create an IP explicit path and enter a configuration submode for specifying the path. The feature adds to the submode commands of the **exclude-address** command for specifying addresses to exclude from the path.

The feature also adds to the submode commands of the **exclude-srlg** command that allows you to specify the IP address to get SRLGs to be excluded from the explicit path.

If the excluded address or excluded srlg for an MPLS TE LSP identifies a flooded link, the constraint-based shortest path first (CSPF) routing algorithm does not consider that link when computing paths for the LSP. If the excluded address specifies a flooded MPLS TE router ID, the CSPF routing algorithm does not allow paths for the LSP to traverse the node identified by the router ID.

Related Topics

[Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 117](#)
[Creating an Explicit Path With Exclude SRLG, on page 119](#)
[Using Explicit Path With Exclude SRLG, on page 120](#)
[Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)
[Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)
[Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

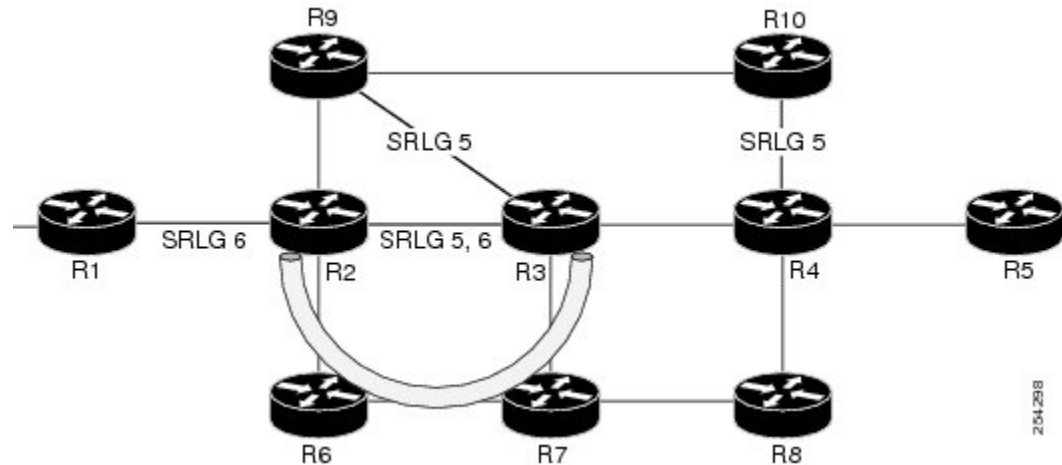
Fast ReRoute with SRLG Constraints

Fast ReRoute (FRR) protects MPLS TE Label Switch Paths (LSPs) from link and node failures by locally repairing the LSPs at the point of failure. This protection allows data to continue to flow on LSPs, while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

Backup tunnels that bypass only a single link of the LSP's path provide Link Protection. They protect LSPs by specifying the protected link IP addresses to extract SRLG values that are to be excluded from the explicit path, thereby bypassing the failed link. These are referred to as **next-hop (NHOP) backup tunnels** because

they terminate at the LSP's next hop beyond the point of failure. [Figure 8: NHOP Backup Tunnel with SRLG constraint](#) illustrates an NHOP backup tunnel.

Figure 8: NHOP Backup Tunnel with SRLG constraint



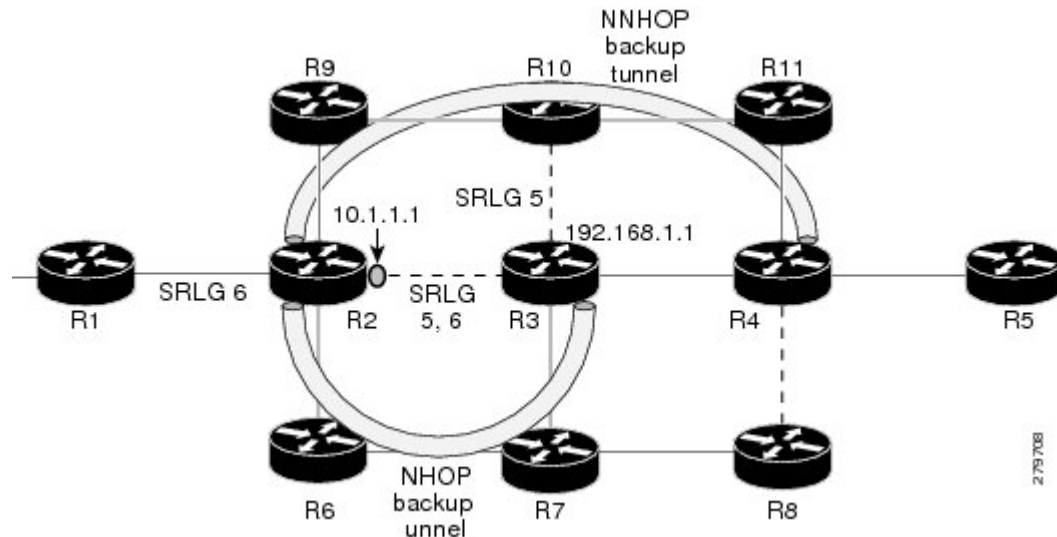
In the topology shown in the above figure, the backup tunnel path computation can be performed in this manner:

- Get all SRLG values from the exclude-SRLG link (SRLG values 5 and 6)
- Mark all the links with the same SRLG value to be excluded from SPF
- Path computation as CSPF R2->R6->R7->R3

FRR provides Node Protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called **NNHOP backup tunnels** because they terminate at the node following the next-hop node of the LSP paths, thereby bypassing the next-hop node. They protect LSPs when a node along their path fails, by enabling the node upstream to the point of failure to reroute the LSPs and their traffic, around the failed node to the next-next hop. They also protect LSPs by specifying the protected link IP addresses that are to be excluded from the explicit path, and the SRLG values associated with the IP addresses excluded from the explicit path.

NNHOP backup tunnels also provide protection from link failures by bypassing the failed link as well as the node. [Figure 9: NNHOP Backup Tunnel with SRLG constraint](#) illustrates an NNHOP backup tunnel.

Figure 9: NNHOP Backup Tunnel with SRLG constraint



In the topology shown in the above figure, the backup tunnel path computation can be performed in this manner:

- Get all SRLG values from the exclude-SRLG link (SRLG values 5 and 6)
- Mark all links with the same SRLG value to be excluded from SPF
- Verify path with SRLG constraint
- Path computation as CSPF R2->R9->R10->R4

Related Topics

[Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 117](#)

[Creating an Explicit Path With Exclude SRLG, on page 119](#)

[Using Explicit Path With Exclude SRLG, on page 120](#)

[Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)

[Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)

[Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Importance of Protection

This section describes the following:

- Delivery of Packets During a Failure
- Multiple Backup Tunnels Protecting the Same Interface

Related Topics

- [Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 117](#)
- [Creating an Explicit Path With Exclude SRLG, on page 119](#)
- [Using Explicit Path With Exclude SRLG, on page 120](#)
- [Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)
- [Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)
- [Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Delivery of Packets During a Failure

Backup tunnels that terminate at the NNHOP protect both the downstream link and node. This provides protection for link and node failures.

Related Topics

- [Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 117](#)
- [Creating an Explicit Path With Exclude SRLG, on page 119](#)
- [Using Explicit Path With Exclude SRLG, on page 120](#)
- [Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)
- [Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)
- [Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Multiple Backup Tunnels Protecting the Same Interface

- Redundancy—If one backup tunnel is down, other backup tunnels protect LSPs.
- Increased backup capacity—If the protected interface is a high-capacity link and no single backup path exists with an equal capacity, multiple backup tunnels can protect that one high-capacity link. The LSPs using this link falls over to different backup tunnels, allowing all of the LSPs to have adequate bandwidth protection during failure (rerouting). If bandwidth protection is not desired, the router spreads LSPs across all available backup tunnels (that is, there is load balancing across backup tunnels).

Related Topics

- [Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 117](#)
- [Creating an Explicit Path With Exclude SRLG, on page 119](#)
- [Using Explicit Path With Exclude SRLG, on page 120](#)
- [Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)
- [Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)
- [Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

SRLG Limitations

There are few limitations to the configured SRLG feature:

- The **exclude-address** and **exclude-srlg** options are not allowed in the IP **explicit path strict-address** network.

- Whenever SRLG values are modified after tunnels are signalled, they are verified dynamically in the next path verification cycle.

Related Topics

- [Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 117](#)
- [Creating an Explicit Path With Exclude SRLG, on page 119](#)
- [Using Explicit Path With Exclude SRLG, on page 120](#)
- [Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)
- [Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)
- [Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

MPLS TE SRLG Scale Enhancements

MPLS Traffic Engineering Shared Risk Link Groups (SRLG) feature has been enhanced to support:

- Increase from 32 to 64 (59 for ISIS) groups.
- Increase from 250 to 500 interfaces.

Related Topics

- [Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 117](#)
- [Creating an Explicit Path With Exclude SRLG, on page 119](#)
- [Using Explicit Path With Exclude SRLG, on page 120](#)
- [Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 122](#)
- [Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 125](#)
- [Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Soft-Preemption

MPLS-TE preemption consists of freeing the resources of an established LSP, and assigning them to a new LSP. The freeing of resources causes a traffic disruption to the LSP that is being preempted. Soft preemption is an extension to the RSVP-TE protocol to minimize and even eliminate such traffic disruption over the preempted LSP.

The soft-preemption feature attempts to preempt the LSPs in a graceful manner to minimize or eliminate traffic loss. However, the link might be over-subscribed for a period of time.

In a network that implements soft preemption, zero traffic loss is achieved in this manner:

- When signaling a new LSP, the ingress router indicates to all the intermediate nodes that the existing LSP is to be softly preempted, in case its resources are needed and is to be reassigned.
- When a given intermediate node needs to soft-preempt the existing LSP, it sends a new or special path error (preemption pending) to the ingress router. The intermediate node does not dismantle the LSP and maintains its state.
- When the ingress router receives the path error (preemption pending) from the intermediate node, it immediately starts a re-optimization that avoids the link that caused the preemption.
- When the re-optimization is complete, the ingress router tears down the soft-preempted LSP.

Related Topics

[Enabling Soft-Preemption on a Node, on page 138](#)

[Enabling Soft-Preemption on a Tunnel, on page 139](#)

Path Option Attributes

The path option attributes are configurable through a template configuration. This template, named **attribute-set**, is configured globally in the MPLS traffic-engineering mode.

You can apply an **attribute-set** to a path option on a per-LSP basis. The path option configuration is extended to take a path option attribute name. LSPs computed with a particular path option uses the attributes as specified by the attribute-set under that path option.

These prerequisites are required to implement path option attributes:

- Path option type attribute-set is configured in the MPLS TE mode
- Path option CLI extended to accept an attribute-set name

**Note**

The **signalled-bandwidth** and **affinity** attributes are supported under the attribute-set template.

Related Topics

[Configuring Attributes within a Path-Option Attribute, on page 140](#)

Configuration Hierarchy of Path Option Attributes

You can specify a value for an attribute within a path option **attribute-set** template. This does not prevent the configuring of the same attribute at a tunnel level. However, it is important to note that only one level is taken into account. So, the configuration at the LSP level is considered more specific than the one at the level of the tunnel, and it is used from this point onwards.

Attributes that are not specified within an attribute-set take their values as usual--configuration at the tunnel level, configuration at the global MPLS level, or default values. Here is an example:

```
attribute-set path-option MYSET
  affinity 0xBEEF mask 0xBEEF

interface tunnel-te 10
  affinity 0xCAFE mask 0xCAFE
  signalled-bandwidth 1000
  path-option 1 dynamic attribute-set name MYSET
  path-option 2 dynamic
```

In this example, the attribute-set named **MYSET** is specifying affinity as 0xBEEF. The signalled bandwidth has not been configured in this **MYSET**. The **tunnel 10**, meanwhile, has affinity 0xCAFE configured. LSPs computed from path-option 1 uses the affinity 0xBEEF/0xBEEF, while LSPs computed from path-option 2 uses the affinity 0xCAFE/0xCAFE. All LSPs computed using any of these path-options use **signalled-bandwidth** as 1000, as this is the only value that is specified only at the tunnel level.

**Note**

The attributes configured in a path option **attribute-set** template takes precedence over the same attribute configured under a tunnel. An attribute configured under a tunnel is used only if the equivalent attribute is **not** specified by the in-use path option **attribute-set** template.

Related Topics

[Configuring Attributes within a Path-Option Attribute, on page 140](#)

Traffic Engineering Bandwidth and Bandwidth Pools

MPLS traffic engineering allows constraint-based routing (CBR) of IP traffic. One of the constraints satisfied by CBR is the availability of required bandwidth over a selected path. Regular TE tunnel bandwidth is called the **global pool**. The **subpool bandwidth** is a portion of the global pool. If it is not in use, the subpool bandwidth is not reserved from the global pool. Therefore, subpool tunnels require a priority higher than that of non-subpool tunnels.

You can configure the signalled-bandwidth path option attribute to use either the global pool (default) or the subpool bandwidth. The signalled-bandwidth value for the path option may be any valid value and the pool does not have to be the same as that which is configured on the tunnel.

**Note**

When you configure signalled-bandwidth for path options with the **signalled-bandwidth bandwidth [sub-pool | global] kbps** command, use either all subpool bandwidths or all global-pool bandwidth values.

Related Topics

[Configuring Attributes within a Path-Option Attribute, on page 140](#)

Path Option Switchover

Reoptimization to a particular path option is not possible if the in-use path option and the new path option do not share the same bandwidth class. The path option switchover operation would fail in such a scenario. Use this command at the EXEC configuration mode to switchover to a newer path option :

mpls traffic-eng switchover *tunnel-xx ID path-option index*

The switchover to a newer path option is achieved, in these instances:

- when a lower index path option is available
- when any signalling message or topology update causes the primary LSP to go down
- when a local interface fails on the primary LSP or a path error is received on the primary LSP

**Note**

Path option switchover between various path options with different bandwidth classes is not allowed.

Related Topics

[Configuring Attributes within a Path-Option Attribute, on page 140](#)

Path Option and Path Protection

When path-protection is enabled, a standby LSP is established to protect traffic going over the tunnel. The standby LSP may be established using either the same path option as the primary LSP, or a different one.

The standby LSP is computed to be diverse from the primary LSP, so bandwidth class differences does not matter. This is true in all cases of diversity except node-diversity. With node diversity, it is possible for the standby LSP to share up to two links with the primary LSP, the link exiting the head node, and the link entering the tail node.

If you want to switchover from one path option to another path option and these path options have different classes, the path option switchover is rejected. However, the path option switchover can not be blocked in the path-protection feature. When the standby LSP becomes active using another path option of a different class type, the path option switchover cannot be rejected at the head end. It might get rejected by the downstream node.

Node-diversity is only possible under limited conditions. The conditions that must be met are:

- there is no second path that is both node and link diverse
- the current LSP uses a shared-media link at the head egress or tail ingress
- the shared-media link used by the current LSP permits computation of a node-diverse path

In Cisco IOS XR, reoptimization between different class types would actually be rejected by the next hop. This rejection will occur by an admission failure.

Related Topics

[Configuring Attributes within a Path-Option Attribute, on page 140](#)

Auto-Tunnel Mesh

The MPLS traffic engineering auto-tunnel mesh (Auto-mesh) feature allows you to set up full mesh of TE P2P tunnels automatically with a minimal set of MPLS traffic engineering configurations. You may configure one or more mesh-groups. Each mesh-group requires a destination-list (IPv4 prefix-list) listing destinations, which are used as destinations for creating tunnels for that mesh-group.

You may configure MPLS TE auto-mesh type attribute-sets (templates) and associate them to mesh-groups. LSR creates tunnels using the tunnel properties defined in the attribute-set.

Auto-Tunnel mesh provides benefits:

- Minimizes the initial configuration of the network.

You may configure tunnel properties template and mesh-groups or destination-lists on each TE LSRs that further creates full mesh of TE tunnels between those LSRs.

- Minimizes future configurations resulting due to network growth.

It eliminates the need to reconfigure each existing TE LSR in order to establish a full mesh of TE tunnels whenever a new TE LSR is added in the network.

Related Topics

- [Configuring Auto-Tunnel Mesh Tunnel ID, on page 141](#)
- [Configuring Auto-tunnel Mesh Unused Timeout, on page 142](#)
- [Configuring Auto-Tunnel Mesh Group, on page 143](#)
- [Configuring Tunnel Attribute-Set Templates, on page 145](#)
- [Enabling LDP on Auto-Tunnel Mesh, on page 146](#)

Destination List (Prefix-List)

Auto-mesh tunnels can be automatically created using prefix-list. Each TE enabled router in the network learns about the TE router IDs through a existing IGP extension.

You can view the router IDs on the router using this command:

```
show mpls traffic-eng topology | include TE Id
IGP Id: 0001.0000.0010.00, MPLS TE Id:100.1.1.1 Router Node (ISIS 1 level-2)
IGP Id: 0001.0000.0011.00, MPLS TE Id:100.2.2.2 Router Node (ISIS 1 level-2)
IGP Id: 0001.0000.0012.00, MPLS TE Id:100.3.3.3 Router Node (ISIS 1 level-2)
```

A prefix-list may be configured on each TE router to match a desired set of router IDs (MPLS TE ID as shown in the above output). For example, if a prefix-list is configured to match addresses of 100.0.0.0 with wildcard 0.255.255.255, then all 100.x.x.x router IDs are included in the auto-mesh group.

When a new TE router is added in the network and its router ID is also in the block of addresses described by the prefix-list, for example, 100.x.x.x, then it is added in the auto-mesh group on each existing TE router without having to explicitly modify the prefix-list or perform any additional configuration.

Auto-mesh does not create tunnels to its own (local) TE router IDs.

**Note**

When prefix-list configurations on all routers are not identical, it can result in non- symmetrical mesh of tunnels between those routers.

Related Topics

- [Configuring Auto-Tunnel Mesh Tunnel ID, on page 141](#)
- [Configuring Auto-tunnel Mesh Unused Timeout, on page 142](#)
- [Configuring Auto-Tunnel Mesh Group, on page 143](#)
- [Configuring Tunnel Attribute-Set Templates, on page 145](#)
- [Enabling LDP on Auto-Tunnel Mesh, on page 146](#)

VRF Redirection to MPLS TE Tunnels

The VRF redirection to MPLS TE tunnels feature adds automatic route MPLS TE tunnels through autoroute destination configuration. The VRF redirection to MPLS TE tunnels maps VRF prefixes over TE tunnels in the core to reach the same egress provider edge (PE). This enables to load-balance prefix traffic on multiple tunnels based on equal cost multi-path (ECMP). The ECMP is used to load-share the flow(s) on multiple available paths towards the destination PE. The route added by autoroute destination inherits the same IGP

computed metric to the tunnel endpoint. Any changes to the IGP route metric to the tunnel endpoint is automatically reflected on the autoroute destination route too.

In a typical VPN deployment over a TE core network, an operator creates a mesh of TE tunnels between PE routers and then configures autoroute announce to these tunnels. This leads to a mix of default VRF and VPNv4 traffic on the same tunnel connecting the PE routers. An operator may want to segregate their VPNv4 traffic on different tunnels. This can be achieved by creating multiple tunnels to the egress PE(s). The limitation of this approach is that the static routes are added with zero metrics. The VRF Redirection to MPLS TE Tunnels feature is a solution to resolve this limitation. Multiple VRFs can be mapped on the same tunnel by adding multiple autoroute destination addresses (BGP next-hops) to the same tunnel.

Routes added by static route are always added with zero cost metric. This results in traffic that is mapped on multiple tunnels to always load-balance due to ECMP. This may be undesirable when some of those tunnels have sub-optimal paths (have higher underlying cost to the endpoint). With autoroute destination, only the tunnel whose IGP cost to its endpoint is lowest will be considered for carrying traffic.

VRF redirection over TE tunnels feature supports:

- Automatic redirection of VRF traffic over TE tunnels.
- Multiple autoroute destinations under one tunnel to aggregate VRF traffic. If two VRFs are to be mapped on same tunnel, then two autoroute destination prefixes (BGP next-hops) will be configured under the tunnel.
- One autoroute destination under multiple tunnels to enable ECMP load-balance of VRF traffic.
- Implicit /32 mask for each route. Only host addresses residing on the tunnel endpoint are supported.
- High availability, RP failover, and non-stop forwarding (NSF) scenarios by proving hitless to traffic mechanisms.

MPLS TE Extended Admin Groups

The MPLS TE extended admin groups (EAG) configuration assigns EAG/AG name to bit-position and associates affinity-names with TE links. The configuration extends to assign names, up to 256, to TE links over the selected interface and assigns 32 names per attribute-set and index.

Use the **affinity-map** *map-name bit-position value* command to assign EAG/AG name to bit-position. Use the **attribute-names** *attribute-name1 attribute-name2 ...* and **attribute-names index index-number** *attribute-name1 attribute-name2 ...* commands to assign up to 32 names per attribute-set and index value.

Stateful Path Computation Element

The stateful path computation element (PCE) describes a set of procedures by which a path computation client (PCC) can report and delegate control of head-end tunnels sourced from the PCC to a PCE peer. The PCE peer can request the PCC to update and modify parameters of label switched paths (LSPs) it controls. The stateful model also enables a PCC to allow the PCE to initiate computations allowing the PCE to perform network-wide orchestration.

The transfer of LSP state and computation constraints is independent from the computation request, such that a PCE may see how state changes over time, without a computation request ever taking place. This allows the PCE to have better visibility into network state, as well as improve the efficiency of computation requests, as these can rely on state present on the PCE.

- Both PCE/PCC functionality runs on routers
- PCE function router need special image or official image with SMU installed
- PCE server could be external third party PCE server, such as Cariden

Stateful PCE provides support for these following request types and objects:

- Request types
 - PCReq—requests used by current stateless PCE implementation
 - PCCreate—LSP instantiation requests
 - PCUpd—LSP update requests
- LSP Objects
 - Operational flag
 - Delegation flag
 - Remove flag
 - Symbolic path name
 - LSP Identifiers
- Path list
 - ERO

Stateful PCE State Reporting

State reporting refers to the PCC sending information to PCEs about the state of LSPs. This is done as state changes occur and is used to keep PCEs informed of changes to the LSP as they occur. State reporting is also used as part of *state synchronization* and *delegation*.

A state report is a message sent by a PCC to a PCE reporting on the state of one or more TE tunnels. This allows the PCE to stay abreast of changes as they occur. Reports are triggered when the PCE needs to be informed of state. These occur when:

- State synchronization happens
- The PCC attempts to delegate control of a tunnel to a PCE
- The PCC revokes control of a tunnel from a PCE
- The PCC deletes a tunnel
- A signalling error occurs on a tunnel
- Reportable information about a tunnel changes

Stateful PCE State Synchronization

Synchronization refers to a procedure that occurs after a PCEP session is established between a PCE and a PCC. The purpose of state synchronization is to download the current LSP database of the PCC to a PCE. This is done through a set of state reports which are marked as *synchronizations*. This is the first communication to occur after the session is brought up. A full re-send of state reports can also be avoided when the PCE already has an up-to-date version of the LSP database as the version number can be indicated by the PCE during PCEP session establishment.

Stateful PCE Delegation

Delegation is the action by which control of a state is granted to a PCE by the PCC. A PCE to which control was delegated can alter attributes of the LSP. Control is only delegated to one PCE at a time.

- Delegation of control can be revoked from a PCE by the PCC.
- Delegation of control can also be returned to the PCC by the PCE.

Stateful PCE State Updating

State updating refers to the PCE sending information to a PCC to alter the attributes of an LSP. A state update is a message sent by a PCE to a PCC to alter the state of one or more TE tunnels. State updating is allowed only if the PCE has previously been delegated control of the LSP. State updating is also used to return delegated control.

Stateful PCE Creation of LSPs

Creation (or instantiation) of an LSP is a procedure by which a PCE instructs a PCC to create an LSP respecting certain attributes. For LSPs created in this manner, the PCE is delegated control automatically. Stateful PCE procedures enable a PCE to instruct a PCC to create a locally sourced tunnel.

MPLS TE Usability Enhancements

MPLS traffic engineering command line interface and logging output messages are enhanced as follows:

- The **show mpls traffic engineering** commands display **signaled-name** and supports **signaled-name** filter.
- Ability to allow immediate teardown of all labelled switched paths (LSPs) of the specified tunnel and to create new LSPs.
- Default behavior when affinity check fails at head-end is to reoptimize all LSP types.
- Logging output messages include MPLS TE tunnel signaled name.
- Logging of path change events and available bandwidth on the new for all auto-bandwidth operations.
- Auto-bandwidth logging output includes signaled name.

MPLS TE IPv6 Autoroute

The MPLS TE IPv6 Autoroute feature enables the use of IPv4 MPLS TE tunnels for IPv6 routing. The routing protocol IGP (IS-IS) considers the IPv4 MPLS TE tunnel for IPv6 routing path calculation only if the tunnel is advertised to carry IPv6 traffic. To advertise the tunnel, either IPv6 autoroute announce (AA) configuration or IPv6 forwarding adjacency (FA) configuration should be made on the tunnel. Also, the IPv6 has to be enabled on the tunnel so that the tunnel can handle IPv6 traffic.

To configure IPv6 routing on an MPLS TEv4 tunnel, see [Configuring IPv6 Routing Over IPv4 MPLS-TE Tunnels](#), on page 149.

MPLS TE IPv6 Autoroute Restrictions

- IGP support is only for IS-IS.
- IS-IS IPv4 and IPv6 must be configured under the same IS-IS instance.
- Unequal load balancing (UELB) does not apply to IPv6 traffic. While it may still be configured and used for IPv4 traffic, IPv6 traffic does not acknowledge the UELB configuration. However, equal loadsharing works for IPv6.
- Policy-based tunnel selection (PBTS) does not apply for IPv6 traffic. While it may still be configured and used for IPv4 traffic, IPv6 traffic does not acknowledge the PBTS configuration.
- MPLS auto tunnels do not support IPv6 autoroute announce and IPv6 forwarding adjacency configurations.

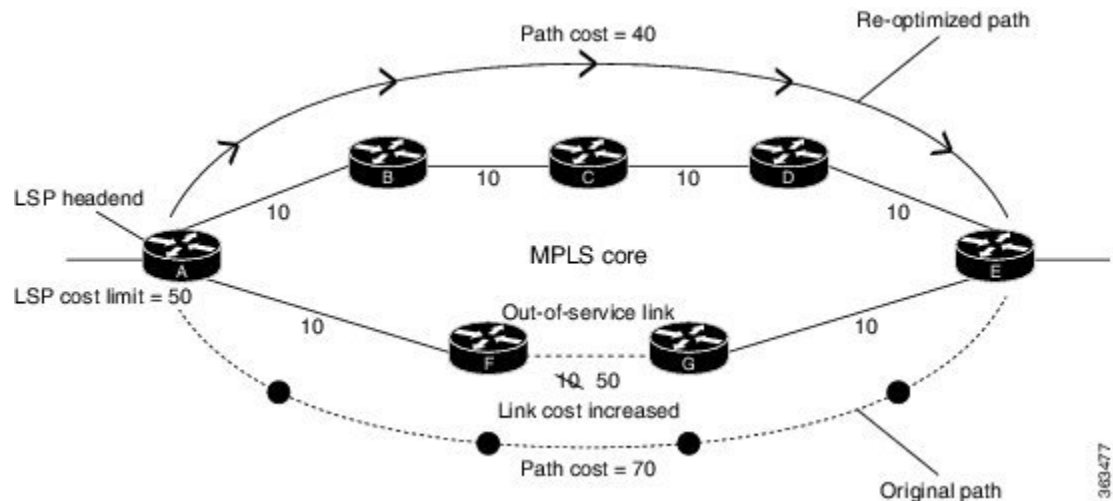
MPLS TE Path Cost Limit

The MPLS TE path cost limit feature enables graceful migration of TE label switched paths (LSPs) away from a link without affecting the traffic. This is useful when a link is scheduled to be decommissioned or brought down for maintenance.

In order to take a link out of service and gracefully migrate the LSPs away from it, the cost assigned to the link is to be set higher than the path cost limit (path aggregate admin-weight) assigned at the LSP headend. The cost of the tunnel is equal to the aggregate cost of the links through which the tunnel passes. The headend routers recalculate the total path costs at the time of periodic path verification. At this stage, the headend routers automatically check if the path limit is crossed and reroute the LSPs away from the out-of-service link.

This sample illustration explains the TE path cost limit application:

Figure 10: MPLS TE path cost limit application



Here, the path cost limit for the LSP is set at 50. To move the LSP away from the link between F and G, the link cost is increased to 50.

The total path cost is the aggregate of individual costs assigned to the links through which the LSP traverses. The effect of specifying a limit to the path cost (admin-weight) are:

- For new LSPs, if the path cost limit is crossed, the LSP is considered invalid and does not get signaled across its calculated path. However, if an alternate path that is below the cost limit is available, then that path is signaled.
- For existing LSPs, if the path cost limit is crossed, the LSP is considered as 'failed'. If the current LSP fails (for both FRR and non-FRR LSPs), the standby LSP will be activated if it exists. If there is no standby LSP, the tunnel will be re-optimized. If there is no standby LSP and no path is found for a re-optimized tunnel then the tunnel is put in 'reroute pending' state and re-optimization is attempted periodically.
- To recover from a cost limit failure, re-optimization will be triggered using any available path option.

Soft-preemption over FRR Backup Tunnels

The soft-preemption over FRR backup tunnels feature enables to move LSP traffic over the backup tunnels when the LSP is soft- preempted. MPLS TE tunnel soft-preemption allows removal of extra TE traffic in a graceful manner, by giving the preempted LSP a grace period to move away from the link. Though this mechanism saves the traffic of the preempted LSP from being dropped, this might cause traffic drops due to congestion as more bandwidth is reserved on the link than what is available. When the soft-preemption over FRR backup tunnel is enabled, the traffic of the preempted LSP is moved onto the FRR backup, if it is available and ready. This way, the capacity of the backup tunnel is used to remove the potential congestion that might be caused by soft-preemption.

MPLS TE Auto-tunnel Mesh One-hop

The MPLS TE Auto-tunnel primary one-hop feature allows automatic creation of tunnels over TE enabled interfaces to next hop neighbors. The Auto-tunnel primary one-hop is configurable under the MPLS TE Auto-tunnel mesh group mode and for each mesh group. The Auto-tunnel primary one-hop configuration automatically creates one-hop tunnels to next hop neighbors. A router that becomes a next hop neighbor will have a set of one-hop tunnels created automatically.

Inter-area Traffic Engineering with Dynamic ABR Discovery

The inter-area traffic engineering with dynamic ABR discovery feature adds support for inter-area point-to-point (P2P) and point-to-multi-point (P2MP) traffic engineering with dynamic ABR discovery. With this feature, there is no need to specify transit ABR addresses in the explicit paths to allow for dynamic/best path computation for inter-area tunnels.

How to Implement Traffic Engineering

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

These procedures are used to implement MPLS-TE:

Building MPLS-TE Topology

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

Before You Begin

Before you start to build the MPLS-TE topology, you must have enabled:

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng RP/0/RP0/CPU0:router(config-mpls-te)#	Enters MPLS-TE configuration mode.
Step 3	interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0 RP/0/RP0/CPU0:router(config-mpls-te-if)#	Enables traffic engineering on a particular interface on the originating node and enters MPLS-TE interface configuration mode.
Step 4	exit Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# exit RP/0/RP0/CPU0:router(config-mpls-te)#	Exits the current configuration mode.
Step 5	exit Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit RP/0/RP0/CPU0:router(config)#	Exits the current configuration mode.

	Command or Action	Purpose
Step 6	router ospf <i>process-name</i> Example: RP/0/RP0/CPU0:router(config)# router ospf 1	Enters a name for the OSPF process.
Step 7	area <i>area-id</i> Example: RP/0/RP0/CPU0:router(config-router)# area 0	Configures an area for the OSPF process. <ul style="list-style-type: none"> • Backbone areas have an area ID of 0. • Non-backbone areas have a non-zero area ID.
Step 8	exit Example: RP/0/RP0/CPU0:router(config-ospf-ar)# exit RP/0/RP0/CPU0:router(config-ospf)#	Exits the current configuration mode.
Step 9	mpls traffic-eng router-id <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1	Sets the MPLS-TE loopback interface.
Step 10	commit	
Step 11	show mpls traffic-eng topology Example: RP/0/RP0/CPU0:router# show mpls traffic-eng topology	(Optional) Verifies the traffic engineering topology.
Step 12	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.

Related Topics

[How MPLS-TE Works, on page 4](#)

[Build MPLS-TE Topology and Tunnels: Example, on page 166](#)

Creating an MPLS-TE Tunnel

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

Before You Begin

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router# interface tunnel-te 1	Configures an MPLS-TE tunnel interface.
Step 3	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)# destination	Assigns a destination address on the new tunnel. The destination address is the remote node's MPLS-TE router ID.

	Command or Action	Purpose
	192.168.92.125	
Step 4	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.
Step 5	path-option <i>preference - priority dynamic</i> Example: RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic	Sets the path option to dynamic and assigns the path ID.
Step 6	signalled-bandwidth { <i>bandwidth [class-type ct] sub-pool bandwidth</i> } Example: RP/0/RP0/CPU0:router(config-if)# signalled-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).
Step 7	commit	
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.

Related Topics

[How MPLS-TE Works, on page 4](#)

[Build MPLS-TE Topology and Tunnels: Example, on page 166](#)

[Building MPLS-TE Topology, on page 50](#)

Configuring Forwarding over the MPLS-TE Tunnel

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

Before You Begin

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 1	Enters MPLS-TE interface configuration mode.

	Command or Action	Purpose
Step 3	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if) # ipv4 unnumbered Loopback0	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.
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 <i>prefix mask ip-address interface type</i> Example: RP/0/RP0/CPU0:router(config) # router static address-family ipv4 unicast 2.2.2.2/32 tunnel-te 1	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 the autoroute announce command is not used.
Step 7	commit	
Step 8	ping { <i>ip-address hostname</i> } 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.

Related Topics

[Overview of MPLS Traffic Engineering, on page 3](#)

[Creating an MPLS-TE Tunnel, on page 53](#)

Protecting MPLS Tunnels with Fast Reroute

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

**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.

Before You Begin

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 9	exit Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit RP/0/RP0/CPU0:router(config)#	Exits the current configuration mode.
Step 10	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 2	Configures an MPLS-TE tunnel interface.
Step 11	backup-bw {<i>backup bandwidth</i> sub-pool {<i>bandwidth</i> unlimited} global-pool {<i>bandwidth</i> 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.
Step 12	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address to set up forwarding on the new tunnel.
Step 13	path-option <i>preference-priority</i> {explicit name <i>explicit-path-name</i>} 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 14	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125	Assigns a destination address on the new tunnel. <ul style="list-style-type: none"> • Destination address is the remote node's MPLS-TE router ID. • Destination address is the merge point between backup and protected tunnels. Note When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.
Step 15	commit	

	Command or Action	Purpose
Step 16	show mpls traffic-eng tunnels backup Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels backup	(Optional) Displays the backup tunnel information.
Step 17	show mpls traffic-eng tunnels protection frr Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels protection frr	(Optional) Displays the tunnel protection information for Fast-Reroute (FRR).
Step 18	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).

Related Topics

- [Fast Reroute, on page 13](#)
- [Fast Reroute Node Protection, on page 20](#)
- [Creating an MPLS-TE Tunnel, on page 53](#)
- [Configuring Forwarding over the MPLS-TE Tunnel, on page 55](#)

Enabling an AutoTunnel Backup

Perform this task to configure the AutoTunnel Backup feature. By default, this feature is disabled. You can configure the AutoTunnel Backup feature for each interface. It has to be explicitly enabled for each interface or link.

SUMMARY STEPS

1. **configure**
2. **ipv4 unnumbered mpls traffic-eng Loopback 0**
3. **mpls traffic-eng**
4. **auto-tunnel backup timers removal unused *frequency***
5. **auto-tunnel backup tunnel-id min *min* max *max***
6. **commit**
7. **show mpls traffic-eng auto-tunnel backup summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	ipv4 unnumbered mpls traffic-eng Loopback 0 Example: RP/0/RP0/CPU0:router(config)#ipv4 unnumbered mpls traffic-eng Loopback 0	Configures the globally configured IPv4 address that can be used by the AutoTunnel Backup Tunnels. Note Loopback 0 is the router ID. The AutoTunnel Backup tunnels will not come up until a global IPv4 address is configured.
Step 3	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 4	auto-tunnel backup timers removal unused frequency Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup timers removal unused 20	Configures how frequently a timer scans the backup automatic tunnels and removes tunnels that are not in use. <ul style="list-style-type: none"> Use the frequency argument to scan the backup automatic tunnel. Range is 0 to 10080. Note You can also configure the auto-tunnel backup command at mpls traffic-eng interface mode.
Step 5	auto-tunnel backup tunnel-id min minmax max Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500	Configures the range of tunnel interface numbers to be used for automatic backup tunnels. Range is 0 to 65535.
Step 6	commit	
Step 7	show mpls traffic-eng auto-tunnel backup summary Example: RP/0/RP0/CPU0:router# show mpls traffic-eng auto-tunnel backup summary	Displays information about configured MPLS-TE backup autotunnels.

Related Topics

[Backup AutoTunnels, on page 5](#)

[Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 179](#)

Removing an AutoTunnel Backup

To remove all the backup autotunnels, perform this task to remove the AutoTunnel Backup feature.

SUMMARY STEPS

1. `clear mpls traffic-eng auto-tunnel backup unused { all | tunnel-tenumber }`
2. `commit`
3. `show mpls traffic-eng auto-tunnel summary`

DETAILED STEPS

	Command or Action	Purpose
Step 1	clear mpls traffic-eng auto-tunnel backup unused { all tunnel-tenumber } Example: RP/0/RP0/CPU0:router# clear mpls traffic-eng auto-tunnel backup unused all	Clears all MPLS-TE automatic backup tunnels from the EXEC mode. You can also remove the automatic backup tunnel marked with specific tunnel-te, provided it is currently unused.
Step 2	commit	
Step 3	show mpls traffic-eng auto-tunnel summary Example: RP/0/RP0/CPU0:router# show mpls traffic-eng auto-tunnel summary	Displays information about MPLS-TE autotunnels including the ones removed.

Related Topics

[Backup AutoTunnels, on page 5](#)

[Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 179](#)

Establishing MPLS Backup AutoTunnels to Protect Fast Reroutable TE LSPs

To establish an MPLS backup autotunnel to protect fast reroutable TE LSPs, perform these steps:

SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng`
3. `interface type interface-path-id`
4. `auto-tunnel backup`
5. `attribute-set attribute-set-name`
6. `commit`
7. `show mpls traffic-eng auto-tunnel backup summary`

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
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-path-id Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0	Enables traffic engineering on a specific interface on the originating node.
Step 4	auto-tunnel backup Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup	Enables an auto-tunnel backup feature for the specified interface. Note You cannot configure the static backup on the similar link.
Step 5	attribute-set attribute-set-name Example: RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)#attribute-set ab	Configures attribute-set template for auto-tunnel backup tunnels.
Step 6	commit	
Step 7	show mpls traffic-eng auto-tunnel backup summary Example: RP/0/RP0/CPU0:router# show mpls traffic auto-tunnel backup summary	Displays information about configured MPLS-TE backup autotunnels.

Related Topics

[Backup AutoTunnels, on page 5](#)

[Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 179](#)

Establishing Next-Hop Tunnels with Link Protection

To establish a next-hop tunnel and link protection on the primary tunnel, perform these steps:

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **interface type interface-path-id**
4. **auto-tunnel backup nhop-only**
5. **auto-tunnel backup exclude srlg [preferred]**
6. **attribute-set attribute-set-name**
7. **commit**
8. **show mpls traffic-eng tunnels number detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
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-path-id Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0	Enables traffic engineering on a specific interface on the originating node.
Step 4	auto-tunnel backup nhop-only Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup nhop-only	Enables the creation of dynamic NHOP backup tunnels. By default, both NHOP and NNHOP protection are enabled. Note Using this nhop-only option, only link protection is provided.
Step 5	auto-tunnel backup exclude srlg [preferred] Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup exclude srlg preferred	Enables the exclusion of SRLG values on a given link for the AutoTunnel backup associated with a given interface. The preferred option allows the AutoTunnel Backup tunnels to come up even if no path excluding all SRLG is found.
Step 6	attribute-set attribute-set-name Example: RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)#attribute-set ab	Configures attribute-set template for auto-tunnel backup tunnels.
Step 7	commit	

	Command or Action	Purpose
Step 8	show mpls traffic-eng tunnels <i>number</i> detail Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 1 detail	Displays information about configured NHOP tunnels and SRLG information.

Related Topics

[Backup AutoTunnels, on page 5](#)

[Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 179](#)

Configuring a Prestandard DS-TE Tunnel

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

Before You Begin

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

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

Related Topics

[Configuring Traffic Engineering Tunnel Bandwidth](#)

[Prestandard DS-TE Mode, on page 10](#)

[Configure IETF DS-TE Tunnels: Example, on page 167](#)

Configuring an IETF DS-TE Tunnel Using RDM

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

Before You Begin

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	rsvp interface <i>type interface-path-id</i> 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 rdm <i>{total-reservable-bw bc0 global-pool} {sub-pool bc1 reservable-bw}</i>	Sets the reserved RSVP bandwidth available on this interface by using the Russian Doll Model (RDM) bandwidth constraints model. The range for the <i>total reserve bandwidth</i> argument is 0 to 4294967295.

	Command or Action	Purpose
	Example: <pre>RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth rdm 100 150</pre>	Note Physical interface bandwidth is not used by MPLS-TE.
Step 4	exit Example: <pre>RP/0/RP0/CPU0:router(config-rsvp-if)# exit RP/0/RP0/CPU0:router(config-rsvp)</pre>	Exits the current configuration mode.
Step 5	exit Example: <pre>RP/0/RP0/CPU0:router(config-rsvp) exit RP/0/RP0/CPU0:router(config)</pre>	Exits the current configuration mode.
Step 6	mpls traffic-eng Example: <pre>RP/0/RP0/CPU0:router(config)# mpls traffic-eng RP/0/RP0/CPU0:router(config-mpls-te)#</pre>	Enters MPLS-TE configuration mode.
Step 7	ds-te mode ietf Example: <pre>RP/0/RP0/CPU0:router(config-mpls-te)# ds-te mode ietf</pre>	Enables IETF DS-TE mode and default TE class map. IETF DS-TE mode is configured on all network nodes.
Step 8	exit Example: <pre>RP/0/RP0/CPU0:router(config-mpls-te)# exit</pre>	Exits the current configuration mode.
Step 9	interface tunnel-te <i>tunnel-id</i> Example: <pre>RP/0/RP0/CPU0:router(config)# interface tunnel-te 4 RP/0/RP0/CPU0:router(config-if)#</pre>	Configures an MPLS-TE tunnel interface.
Step 10	signalled-bandwidth {<i>bandwidth</i> [class-type <i>ct</i>] sub-pool <i>bandwidth</i>}	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
	Example: RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10 class-type 1	
Step 11	commit	

Related Topics

[Configuring Traffic Engineering Tunnel Bandwidth](#)
[Russian Doll Bandwidth Constraint Model, on page 11](#)

Configuring an IETF DS-TE Tunnel Using MAM

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

Before You Begin

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	rsvp interface <i>type interface-path-id</i> 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 mam { <i>total reservable bandwidth</i> max-reservable-bw <i>maximum-reservable-bw</i> } [bc0 <i>reservable bandwidth</i>] [bc1 <i>reservable bandwidth</i>] 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-rsvp-if)# exit RP/0/RP0/CPU0:router(config-rsvp)#	Exits the current configuration mode.
Step 5	exit Example: RP/0/RP0/CPU0:router(config-rsvp)# exit RP/0/RP0/CPU0:router(config)#	Exits the current configuration mode.
Step 6	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng RP/0/RP0/CPU0:router(config-mpls-te)#	Enters MPLS-TE configuration mode.
Step 7	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.

	Command or Action	Purpose
Step 8	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 9	exit Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit	Exits the current configuration mode.
Step 10	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 4 RP/0/RP0/CPU0:router(config-if)#	Configures an MPLS-TE tunnel interface.
Step 11	signalled-bandwidth {<i>bandwidth</i> [class-type <i>ct</i>] sub-pool <i>bandwidth</i>} Example: RP/0/RP0/CPU0:router(config-rsvp-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 12	commit	

Related Topics

[Configuring Traffic Engineering Tunnel Bandwidth](#)
[Maximum Allocation Bandwidth Constraint Model, on page 10](#)

Configuring MPLS -TE and Fast-Reroute on OSPF

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

Before You Begin



Note

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 1 RP/0/RP0/CPU0:router(config-if)#	Configures an MPLS-TE tunnel interface. The range for the tunnel ID number is 0 to 65535.
Step 3	path-option [protecting] <i>preference-priority</i> { dynamic [pce [address ipv4 address] explicit { name <i>pathname</i> identifier <i>path-number</i> } } [isis instance name { level level }] [ospf instance name { area area ID }]] [verbatim] [lockdown] Example: RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit identifier 6 ospf green area 0	Configures an explicit path option for an MPLS-TE tunnel. OSPF is limited to a single OSPF instance and area.
Step 4	Repeat Step 3 as many times as needed. Example: RP/0/RP0/CPU0:router(config-if)# path-option 2 explicit	Configures another explicit path option.

	Command or Action	Purpose
	<code>name 234 ospf 3 area 7 verbatim</code>	
Step 5	<code>commit</code>	
Step 6	show mpls traffic-eng tunnels [<i>tunnel-number</i>] Example: RP/0/RP0/CPU0:router# <code>show mpls traffic-eng tunnels 1</code>	Displays information about MPLS-TE tunnels.

Related Topics

[Configure MPLS-TE and Fast-Reroute on OSPF: Example, on page 168](#)

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

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

SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng`
3. `path-selection ignore overload {head | mid | tail}`
4. `commit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>configure</code>	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# <code>mpls traffic-eng</code> RP/0/RP0/CPU0:router(config-mpls-te)#	Enters MPLS-TE configuration mode.
Step 3	path-selection ignore overload {head mid tail} Example: RP/0/RP0/CPU0:router(config-mpls-te)# <code>path-selection ignore overload head</code>	Ignores the Intermediate System-to-Intermediate System (IS-IS) overload bit setting for MPLS-TE. If set-overload-bit is set by IS-IS on the head router, the tunnels stay up.

	Command or Action	Purpose
Step 4	commit	

Related Topics

[Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE, on page 14](#)
[Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 168](#)

Configuring GMPLS

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

- [Configuring IPCC Control Channel Information, on page 74](#)
- [Configuring Local and Remote TE Links, on page 78](#)
- [Configuring Numbered and Unnumbered Optical TE Tunnels, on page 88](#)
- [Configuring LSP Hierarchy, on page 92](#)
- [Configuring Border Control Model, on page 93](#)
- [Configuring Path Protection, on page 93](#)



Note

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

Configuring IPCC Control Channel Information

To configure IPCC control channel information, complete these subtasks:

- [Configuring Router IDs, on page 74](#)
- [Configuring OSPF over IPCC, on page 76](#)



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-path-id*
3. **ipv4 address** *ipv4-address mask*
4. **exit**
5. **router ospf** *process-name*
6. **mpls traffic-eng router-id** *type interface-path-id*
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface <i>type interface-path-id</i> 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 <i>ipv4-address mask</i> 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. <ul style="list-style-type: none"> • 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. • 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-if)# exit RP/0/RP0/CPU0:router(config)#	Exits the current configuration mode.
Step 5	router ospf <i>process-name</i> Example: RP/0/RP0/CPU0:router(config)# router ospf 1 RP/0/RP0/CPU0:router(config-ospf)#	Configures an Open Shortest Path First (OSPF) routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

	Command or Action	Purpose
Step 6	mpls traffic-eng router-id <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router id Loopback0	Specifies that the TE router identifier for the node is the IP address that is associated with a given interface. The router ID is specified with an interface name or an IP address. By default, MPLS uses the global router ID.
Step 7	commit	

Related Topics

[GMPLS Support](#) , on page 16

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** *type interface-path-id*
5. **exit**
6. **exit**
7. **mpls traffic-eng router-id** {*type interface-path-id* | *ip-address* }
8. **area** *area-id*
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
Step 2	router ospf <i>process-name</i> Example: RP/0/RP0/CPU0:router(config) # router ospf 1	Configures OSPF routing and assigns a process name.
Step 3	area <i>area-id</i> 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): <ul style="list-style-type: none"> • Backbone areas have an area ID of 0. • Non-backbone areas have a nonzero area ID.
Step 4	interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-ospf-ar) # interface Loopback0	Enables IGP on the interface. This command is used to configure any interface included in the control network.
Step 5	exit Example: RP/0/RP0/CPU0:router(config-ospf-ar-if) # exit RP/0/RP0/CPU0:router(config-ospf-ar) #	Exits the current configuration mode.
Step 6	exit Example: RP/0/RP0/CPU0:router(config-ospf-ar) # exit RP/0/RP0/CPU0:router(config-ospf) #	Exits the current configuration mode.
Step 7	mpls traffic-eng router-id { <i>type interface-path-id</i> <i>ip-address</i> } Example: RP/0/RP0/CPU0:router(config-ospf) # mpls traffic-eng router-id 192.168.25.66	Configures a router ID for the OSPF process using an IP address.
Step 8	area <i>area-id</i> Example: RP/0/RP0/CPU0:router(config-ospf) # area 0 RP/0/RP0/CPU0:router(config-ospf-ar) #	Configures the MPLS-TE area.
Step 9	commit	

Related Topics

[GMPLS Support](#) , on page 16

Configuring Local and Remote TE Links

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

- [Configuring Numbered and Unnumbered Links](#), on page 78
- [Configuring Local Reservable Bandwidth](#), on page 79
- [Configuring Local Switching Capability Descriptors](#), on page 80
- [Configuring Persistent Interface Index](#), on page 81
- [Enabling LMP Message Exchange](#), on page 82
- [Disabling LMP Message Exchange](#), on page 83
- [Configuring Remote TE Link Adjacency Information for Numbered Links](#), on page 84
- [Configuring Remote TE Link Adjacency Information for Unnumbered Links](#), on page 86

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 STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. Do one of the following:
 - **ipv4 address** *ipv4-address mask*
 - **ipv4 unnumbered interface** *type interface-path-id*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
Step 2	interface <i>type interface-path-id</i> 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	Do one of the following: <ul style="list-style-type: none"> • ipv4 address <i>ipv4-address mask</i> • ipv4 unnumbered interface <i>type interface-path-id</i> 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. <ul style="list-style-type: none"> • Network mask is 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. • Network mask is 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 <ul style="list-style-type: none"> • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface. Note If you configured a unnumbered GigabitEthernet interface in Step 2 and selected the ipv4 unnumbered interface command type option in this step, you must enter the ipv4 point-to-point command to configure point-to-point interface mode.
Step 4	commit	

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-path-id*
3. **bandwidth** [*total reservable bandwidth*] [**bc0** *bandwidth*] [**global-pool** *bandwidth*] [**sub-pool** *reservable-bw*]
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	rsvp interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config)# rsvp interface POS0/6/0/0	Enters RSVP configuration mode and selects an RSVP interface ID.
Step 3	bandwidth [<i>total reservable bandwidth</i>] [bc0 <i>bandwidth</i>] [global-pool <i>bandwidth</i>] [sub-pool <i>reservable-bw</i>] 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	commit	

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-path-id*
4. **floodig-igp ospf** *instance-id area area-id*
5. **switching key** *value* [**encoding** *encoding type*]
6. **switching key** *value* [**capability** {**psc1** | **lsc** | **fsc**}]
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
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-path-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	floodig-igp ospf instance-id area area-id Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# floodig-igp ospf 0 area 1	Specifies the IGP OSPF interface ID and area where the TE links are to be flooded.
Step 5	switching key value [encoding encoding type] Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# switching key 1 encoding ethernet	Specifies the switching configuration for the interface and enters switching key mode where you will configure encoding and capability. Note The recommended switch key value is 0.
Step 6	switching key value [capability {psc1 lsc fsc}] Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# switching key 1 capability psc1	Specifies the interface switching capability type. The recommended switch capability type is psc1 .
Step 7	commit	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
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	commit	

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 mode.

**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. **commit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	Imp neighbor <i>name</i> 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).
Step 5	remote node-id <i>node-id</i> Example: RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1) # remote node-id 2.2.2.2	Configures the remote node ID for an LMP neighbor. In addition, the <i>node-id</i> value can be an IPv4 address.
Step 6	commit	

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 mode.



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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	lmp neighbor <i>name</i> 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-nbr-OXC1)# lmp static	Disables dynamic LMP procedures for the specified neighbor, including LMP hello and LMP link summary. This command is used 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.
Step 6	remote node-id <i>node-id</i> Example: RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# remote node-id 2.2.2.2	Configures the remote node ID for an LMP neighbor. The node ID value must be an IPv4 address.
Step 7	commit	

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-path-id*
4. **lmp data-link adjacency**
5. **remote switching-capability** {fsc | lsc | psc1}
6. **remote interface-id** unnum *value*
7. **remote node-id** *node-id*
8. **neighbor** *name*
9. **remote node-id** *address*
10. **commit**
11. **show mpls lmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	interface <i>type interface-path-id</i> 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.

	Command or Action	Purpose
Step 6	remote interface-id unnum <i>value</i> Example: <pre>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote interface-id unnum 7</pre>	Configures the unnumbered interface identifier. Identifiers, which you specify by using this command, are the values assigned by the neighbor at the remote side.
Step 7	remote node-id <i>node-id</i> Example: <pre>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote node-id 10.10.10.10</pre>	Configures the remote node ID.
Step 8	neighbor <i>name</i> Example: <pre>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# neighbor OXC1</pre>	Configures or updates an LMP neighbor and its associated parameters.
Step 9	remote node-id <i>address</i> Example: <pre>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote node-id 10.10.10.10</pre>	Configures the remote node ID.
Step 10	commit	
Step 11	show mpls lmp Example: <pre>RP/0/RP0/CPU0:router# show mpls lmp</pre>	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-path-id*
4. **lmp data link adjacency**
5. **neighbor** *name*
6. **remote te-link-id unnum**
7. **remote interface-id unnum** *interface-identifier*
8. **remote switching-capability** {fsc | lsc | psc1}
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	interface <i>type interface-path-id</i> 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 <i>name</i> Example: RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# neighbor OXCL	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	Configures the unnumbered interface and identifier.

	Command or Action	Purpose
	<code>te-link-id unnum 111</code>	
Step 7	remote interface-id unnum <i>interface-identifier</i> Example: RP/0/RP0/CPU0:router(config-mpls-te-if-adj) # remote interface-id unnum 7	Configures the unnumbered interface identifier. Identifiers, which you specify by using this command, are the values assigned by the neighbor at the remote side.
Step 8	remote switching-capability {fsc lsc psc1} Example: RP/0/RP0/CPU0:router(config-mpls-te-if-adj) # remote switching-capability lsc	Configures remote the LMP MPLS-TE interface switching capability.
Step 9	commit	

Configuring Numbered and Unnumbered Optical TE Tunnels

These subtasks are included:

- [Configuring an Optical TE Tunnel Using Dynamic Path Option, on page 88](#)
- [Configuring an Optical TE Tunnel Using Explicit Path Option, on page 91](#)



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**

The examples describe how to configure optical tunnels. It does not include procedures for every option available on the headend and tailend routers.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-gte** *tunnel-id*
3. **ipv4 address** *ip-address/prefix* or **ipv4 unnumbered** *type interface-path-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** *ip-address*
8. **path-option** *path-id* **dynamic**
9. **direction** [*bidirectional*]
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-gte <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-gte1	Configures an MPLS-TE tunnel for GMPLS interfaces.
Step 3	ipv4 address <i>ip-address/prefix</i> or ipv4 unnumbered <i>type interface-path-id</i> 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. <ul style="list-style-type: none"> • 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. • 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 <ul style="list-style-type: none"> • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.

	Command or Action	Purpose
Step 4	switching transit <i>switching type encoding encoding type</i> 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 <i>priority hold-priority</i> Example: RP/0/RP0/CPU0:router(config-if) # priority 1 1	Configures setup and reservation priorities for MPLS-TE tunnels.
Step 6	signalled-bandwidth { <i>bandwidth [class-type ct] sub-pool bandwidth</i> } 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 <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if) # destination 192.168.92.125	Assigns a destination address on the new tunnel. <ul style="list-style-type: none"> • Destination address is the remote node's MPLS-TE router ID. • Destination address is the merge point between backup and protected tunnels.
Step 8	path-option <i>path-id dynamic</i> Example: RP/0/RP0/CPU0:router(config-if) # path-option 1 dynamic	Configures the dynamic path option and path ID.
Step 9	direction [<i>bidirectional</i>] Example: RP/0/RP0/CPU0:router(config-if) # direction bidirection	Configures a bidirectional optical tunnel for GMPLS.
Step 10	commit	

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 be applied to both the headend and tailend router.



Note

You cannot configure dynamic tunnels on the tailend router.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-gte** *tunnel-id*
3. **ipv4 address** *ipv4-address mask* or **ipv4 unnumbered** *type interface-path-id*
4. **passive**
5. **match identifier** *tunnel number*
6. **destination** *ip-address*
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-gte <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-gte 1 RP/0/RP0/CPU0:router(config-if)#	Configures an MPLS-TE tunnel interface for GMPLS interfaces.
Step 3	ipv4 address <i>ipv4-address mask</i> or ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 address 127.0.0.1 255.0.0.0	Specifies a primary or secondary IPv4 address for an interface. <ul style="list-style-type: none"> • 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. • 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 <ul style="list-style-type: none"> • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.
Step 4	passive	Configures a passive interface.

	Command or Action	Purpose
	Example: <pre>RP/0/RP0/CPU0:router(config-if)# passive</pre>	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.
Step 5	match identifier <i>tunnel number</i> Example: <pre>RP/0/RP0/CPU0:router(config-if)# match identifier gmpls1_t1</pre>	Configures the match identifier. You must enter the hostname for the head router then underscore <code>_t</code> , 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-gte number configured on the headend router. Together with the address specified using the destination command, this identifier uniquely identifies acceptable incoming tunnel requests.
Step 6	destination <i>ip-address</i> Example: <pre>RP/0/RP0/CPU0:router(config-if)# destination 10.1.1.1</pre>	Assigns a destination address on the new tunnel. <ul style="list-style-type: none"> • Destination address is the remote node's MPLS-TE router ID. • Destination address is the merge point between backup and protected tunnels.
Step 7	commit	

Configuring LSP Hierarchy

These tasks describe the high-level steps that are 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.

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.

Related Topics

[Configuring Numbered and Unnumbered Optical TE Tunnels, on page 88](#)

Configuring Border Control Model

Border control 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:

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



Note

Border control model 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

These tasks describe how to configure path protection:

- [Configuring an LSP, on page 94](#)
- [Forcing Reversion of the LSP, on page 96](#)

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-gte** *number*
3. **ipv4 address** *ipv4-address mask* or **ipv4 unnumbered** *type interface-path-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** *ip-address*
10. **path-option** *path-id explicit* {**name** *pathname* | *path-number* }
11. **path-option protecting** *path-id explicit* {**name** *pathname* | *path-number*}
12. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-gte <i>number</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-gte 1	Configures an MPLS-TE tunnel interface for GMPLS interfaces.
Step 3	ipv4 address <i>ipv4-address mask</i> or ipv4 unnumbered <i>type interface-path-id</i> 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. <ul style="list-style-type: none"> • 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.

	Command or Action	Purpose
		<ul style="list-style-type: none"> 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. <p>or</p> <ul style="list-style-type: none"> Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.
Step 4	signalled-name <i>name</i> Example: RP/0/RP0/CPU0:router(config-if) # signalled-name tunnel-gt1	Configures the name of the tunnel required for an MPLS TE tunnel. The <i>name</i> argument specifies the signal for the tunnel.
Step 5	switching transit <i>capability-switching-type encoding encoding-type</i> 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 <i>capability-switching -ype encoding encoding-type</i> Example: RP/0/RP0/CPU0:router(config-if) # switching endpoint pscl 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.
Step 7	priority <i>setup-priority hold-priority</i> Example: RP/0/RP0/CPU0:router(config-if) # priority 2 2	Configures setup and reservation priorities for MPLS-TE tunnels.
Step 8	signalled-bandwidth { <i>bandwidth [class-type ct] sub-pool bandwidth</i> } 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 the Diff-Serv Aware TE (DS-TE) feature.
Step 9	destination <i>ip-address</i>	Assigns a destination address on the new tunnel.

	Command or Action	Purpose
	Example: <pre>RP/0/RP0/CPU0:router(config-if)# destination 24.24.24.24</pre>	<ul style="list-style-type: none"> • Destination address is the remote node's MPLS-TE router ID. • Destination address is the merge point between backup and protected tunnels.
Step 10	path-option path-id explicit {name pathname path-number } Example: <pre>RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name po4</pre>	Configures the explicit path option and path ID.
Step 11	path-option protecting path-id explicit {name pathname path-number } Example: <pre>RP/0/RP0/CPU0:router(config-if)# path-option protecting 1 explicit name po6</pre>	Configures the path setup option to protect a path.
Step 12	commit	

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. **mpls traffic-eng path-protection switchover {gmpls tunnel-name | tunnel-te tunnel-id }**
2. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	mpls traffic-eng path-protection switchover {gmpls tunnel-name tunnel-te tunnel-id } Example: <pre>RP/0/RP0/CPU0:router# mpls traffic-eng path-protection switchover tunnel-te 1</pre>	<p>Specifies a manual switchover for path protection for a GMPLS optical LSP. The tunnel ID is configured for a switchover.</p> <p>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.</p>
Step 2	commit	

Configuring Flexible Name-based Tunnel Constraints

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

- 1 [Assigning Color Names to Numeric Values](#), on page 97
- 2 [Associating Affinity-Names with TE Links](#), on page 98
- 3 [Associating Affinity Constraints for TE Tunnels](#), on page 99

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* | **bit-position value**}
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng RP/0/RP0/CPU0:router(config-mpls-te)#	Enters MPLS-TE configuration mode.
Step 3	affinity-map <i>affinity name</i> { <i>affinity value</i> bit-position value }	Enters an affinity name and a map value by using a color name (repeat this command to assign multiple colors up to a maximum of 64 colors). An affinity color name cannot

	Command or Action	Purpose
	Example: RP/0/RP0/CPU0:router(config-mpls-te) # affinity-map red 1	exceed 64 characters. The value you assign to a color name must be a single digit.
Step 4	commit	

Related Topics

[Flexible Name-based Tunnel Constraints, on page 17](#)

[Configure Flexible Name-based Tunnel Constraints: Example, on page 170](#)

Associating Affinity-Names with TE Links

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng RP/0/RP0/CPU0:router(config-mpls-te)#	Enters MPLS-TE configuration mode.
Step 3	interface type interface-path-id Example: RP/0/RP0/CPU0:router(config-mpls-te) # interface tunnel-te 2	Enables MPLS-TE on an interface and enters MPLS-TE interface configuration mode.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-mpls-te-if)#	
Step 4	attribute-names <i>attribute name</i> Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# attribute-names red	Assigns colors to TE links over the selected interface.
Step 5	commit	

Related Topics

[Flexible Name-based Tunnel Constraints, on page 17](#)

[Configure Flexible Name-based Tunnel Constraints: Example, on page 170](#)

[Assigning Color Names to Numeric Values, on page 97](#)

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** {*affinity-value* **mask** *mask-value* | **exclude** *name* | **exclude -all** | **include** *name* | **include-strict** *name*}
4. **commit**

DETAILED STEPS

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

Related Topics

[Flexible Name-based Tunnel Constraints, on page 17](#)

[Configure Flexible Name-based Tunnel Constraints: Example, on page 170](#)

Configuring IS-IS to Flood MPLS-TE Link Information

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	router isis <i>instance-id</i> Example: RP/0/RP0/CPU0:router(config)# router isis 1	Enters an IS-IS instance.
Step 3	net <i>network-entity-title</i> 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 { <i>ipv4</i> <i>ipv6</i> } { <i>unicast</i> } Example: RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast	Enters address family configuration mode for configuring IS-IS routing that uses IPv4 and IPv6 address prefixes.
Step 5	metric-style <i>wide</i> Example: RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide	Enters the new-style type, length, and value (TLV) objects.
Step 6	mpls traffic-eng <i>level</i> Example: RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng level-1-2	Enters the required MPLS-TE level or levels.
Step 7	commit	

Configuring an OSPF Area of MPLS-TE

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	router ospf <i>process-name</i> Example: RP/0/RP0/CPU0:router(config)# router ospf 100	Enters a name that uniquely identifies an OSPF routing process. <i>process-name</i> Any alphanumeric string no longer than 40 characters without spaces.
Step 3	mpls traffic-eng router-id <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1	Enters the MPLS interface type. For more information, use the question mark (?) online help function.
Step 4	area <i>area-id</i> Example: RP/0/RP0/CPU0:router(config-ospf)# area 0	Enters an OSPF area identifier. <i>area-id</i> Either a decimal value or an IP address.
Step 5	interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-ospf-ar)# interface POS 0/2/0/0	Identifies an interface ID. For more information, use the question mark (?) online help function.
Step 6	commit	

Configuring Explicit Paths with ABRs Configured as Loose Addresses

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

SUMMARY STEPS

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

DETAILED STEPS

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

Configuring MPLS-TE Forwarding Adjacency

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 1	Enters MPLS-TE interface configuration mode.
Step 3	forwarding-adjacency holdtime <i>value</i> 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	commit	

Related Topics

- [MPLS-TE Forwarding Adjacency Benefits, on page 21](#)
- [Configure Forwarding Adjacency: Example, on page 173](#)

Configuring Unequal Load Balancing

Perform these tasks to configure unequal load balancing:

- [Setting Unequal Load Balancing Parameters, on page 104](#)
- [Enabling Unequal Load Balancing, on page 105](#)

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 tunnel-te** *tunnel-id*
3. **load-share** *value*
4. **commit**
5. **show mpls traffic-eng tunnels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 1	Configures an MPLS-TE tunnel 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 <i>value</i> Example: RP/0/RP0/CPU0:router(config-if)# load-share 1000	Configures the load-sharing parameters for the specified interface.
Step 4	commit	
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.

Related Topics

[Unequal Load Balancing, on page 22](#)

[Configure Unequal Load Balancing: Example, on page 173](#)

Enabling Unequal Load Balancing

This task describes how to enable unequal load balancing. (For example, 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. **commit**
5. **show mpls traffic-eng tunnels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
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	commit	
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.

Related Topics

[Unequal Load Balancing, on page 22](#)

[Configure Unequal Load Balancing: Example, on page 173](#)

Configuring a Path Computation Client and Element

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

- [Configuring a Path Computation Client, on page 106](#)
- [Configuring a Path Computation Element Address, on page 107](#)
- [Configuring PCE Parameters, on page 108](#)

Configuring a Path Computation Client

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

**Note**

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> 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	path-option <i>preference-priority</i> dynamic pce Example: RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic pce	Configures a TE tunnel as a PCC.
Step 4	commit	

Related Topics

[Path Computation Element, on page 23](#)

[Configure PCE: Example, on page 174](#)

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. `commit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>configure</code>	
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	<code>commit</code>	

Related Topics

[Path Computation Element, on page 23](#)

[Configure PCE: Example, on page 174](#)

Configuring PCE Parameters

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
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 <i>address</i> 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 ipv4 <i>address</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# pce peer address ipv4 10.1.1.1	Configures a static PCE peer address. PCE peers are also discovered dynamically through OSPF or ISIS.
Step 5	pce keepalive <i>interval</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# pce keepalive 10	Configures a PCEP keepalive interval. The range is from 0 to 255 seconds. When the keepalive interval is 0, the LSR does not send keepalive messages.

	Command or Action	Purpose
Step 6	<p>pce deadtimer <i>value</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpls-te)# pce deadtimer 50</pre>	Configures a PCE deadtimer value. The range is from 0 to 255 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.
Step 7	<p>pce reoptimize <i>value</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpls-te)# pce reoptimize 200</pre>	Configures a periodic reoptimization timer value. The range is from 60 to 604800 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.
Step 8	<p>pce request-timeout <i>value</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpls-te)# pce request-timeout 10</pre>	Configures a PCE request-timeout. Range is from 5 to 100 seconds. PCC or PCE keeps a pending path request only for the request-timeout period.
Step 9	<p>pce tolerance keepalive <i>value</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-mpls-te)# pce tolerance keepalive 10</pre>	Configures a PCE tolerance keepalive value (which is the minimum acceptable peer proposed keepalive).
Step 10	commit	
Step 11	<p>show mpls traffic-eng pce peer [<i>address</i> all]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show mpls traffic-eng pce peer</pre>	Displays the PCE peer address and state.
Step 12	<p>show mpls traffic-eng pce tunnels</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show mpls traffic-eng pce tunnels</pre>	Displays the status of the PCE tunnels.

Related Topics

[Path Computation Element, on page 23](#)

[Configure PCE: Example, on page 174](#)

Configuring Policy-based Tunnel Selection

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 6	Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.
Step 3	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address so that forwarding can be performed on the new tunnel.
Step 4	signalled-bandwidth { <i>bandwidth</i> [<i>class-type</i> <i>ct</i>] <i>sub-pool bandwidth</i> } 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 5	autoroute announce Example: RP/0/RP0/CPU0:router(config-if)# autoroute	Enables messages that notify the neighbor nodes about the routes that are forwarding.

	Command or Action	Purpose
	announce	
Step 6	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if) # destination 10.1.1.1	Assigns a destination address on the new tunnel. <ul style="list-style-type: none"> • Destination address is the remote node's MPLS-TE router ID. • Destination address is the merge point between backup and protected tunnels.
Step 7	policy-class { <i>1 - 7</i> } {default} Example: RP/0/RP0/CPU0:router(config-if) # policy-class 1	Configures PBTS to direct traffic into specific TE tunnels or default class. Multiple EXP values can be specified as part of a policy-class, separated by spaces. The EXP values configured to a TE tunnel effectively form a monolithic policy-class, which should not overlap with other policy-classes. Once an EXP value is used in a policy-class configuration, it can only be reused if the subsequent policy-class configurations containing that EXP value are identical. For example, if the configuration policy-class 1 2 3 is applied to one or more tunnels, configurations such as policy-class 1 , policy-class 2 3 , or policy-class 3 4 5 become invalid.
Step 8	path-option <i>preference-priority</i> { explicit name <i>explicit-path-name</i> } 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	commit	

Related Topics

[Policy-Based Tunnel Selection Functions, on page 25](#)

[Policy-Based Tunnel Selection, on page 24](#)

[Configure Policy-based Tunnel Selection: Example, on page 175](#)

Configuring the Automatic Bandwidth

Perform these tasks to configure the automatic bandwidth:

Configuring the Collection Frequency

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng RP/0/RP0/CPU0:router(config-mpls-te)#	Enters MPLS-TE configuration mode.
Step 3	auto-bw collect frequency <i>minutes</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-bw collect frequency 1	Configures the automatic bandwidth collection frequency, and controls the manner in which the bandwidth for a tunnel collects output rate information; but does not adjust the tunnel bandwidth. <i>minutes</i> Configures the interval between automatic bandwidth adjustments in minutes. Range is from 1 to 10080.
Step 4	commit	
Step 5	show mpls traffic-eng tunnels [auto-bw] Example: RP/0/RP0/CPU0:router# show mpls traffic tunnels auto-bw	Displays information about MPLS-TE tunnels for the automatic bandwidth. The globally configured collection frequency is displayed.

Related Topics

[MPLS-TE Automatic Bandwidth Overview, on page 28](#)
[Configure Automatic Bandwidth: Example, on page 176](#)

Forcing the Current Application Period to Expire Immediately

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	mpls traffic-eng auto-bw apply {all tunnel-te <i>tunnel-number</i>} Example: RP/0/RP0/CPU0:router# mpls traffic-eng auto-bw apply tunnel-te 1	Configures the highest bandwidth available on a tunnel without waiting for the current application period to end. all Configures the highest bandwidth available instantly on all the tunnels. tunnel-te Configures the highest bandwidth instantly to the specified tunnel. Range is from 0 to 65535.
Step 2	commit	
Step 3	show mpls traffic-eng tunnels [auto-bw] Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-bw	Displays information about MPLS-TE tunnels for the automatic bandwidth.

Related Topics

[Restrictions for MPLS-TE Automatic Bandwidth, on page 30](#)

Configuring the Automatic Bandwidth Functions

Perform this task to configure the following automatic bandwidth functions:

Application frequency

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

Bandwidth collection

Configures only the bandwidth collection.

Bandwidth parameters

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

Adjustment threshold

Configures the adjustment threshold for each tunnel.

Overflow detection

Configures the overflow detection for each tunnel.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 6 RP/0/RP0/CPU0:router(config-if)#	Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.
Step 3	auto-bw Example: RP/0/RP0/CPU0:router(config-if)# auto-bw RP/0/RP0/CPU0:router(config-if-tunte-autobw)#	Configures automatic bandwidth on a tunnel interface and enters MPLS-TE automatic bandwidth interface configuration mode.

	Command or Action	Purpose
Step 4	application <i>minutes</i> Example: RP/0/RP0/CPU0:router(config-if-tunte-autobw) # application 1000	Configures the application frequency in minutes for the applicable tunnel. minutes Frequency in minutes for the automatic bandwidth application. Range is from 5 to 10080 (7 days). The default value is 1440 (24 hours).
Step 5	bw-limit { <i>min bandwidth</i> } { <i>max bandwidth</i> } Example: RP/0/RP0/CPU0:router(config-if-tunte-autobw) # bw-limit min 30 max 80	Configures the minimum and maximum automatic bandwidth set on a tunnel. min Applies the minimum automatic bandwidth in kbps on a tunnel. Range is from 0 to 4294967295. max Applies the maximum automatic bandwidth in kbps on a tunnel. Range is from 0 to 4294967295.
Step 6	adjustment-threshold <i>percentage</i> [<i>min minimum-bandwidth</i>] Example: RP/0/RP0/CPU0:router(config-if-tunte-autobw) # adjustment-threshold 50 min 800	Configures the tunnel bandwidth change threshold to trigger an adjustment. percentage Bandwidth change percent threshold to trigger an adjustment if the largest sample percentage is higher or lower than the current tunnel bandwidth. Range is from 1 to 100 percent. The default value is 5 percent. min Configures the bandwidth change value to trigger an adjustment. The tunnel bandwidth is changed only if the largest sample is higher or lower than the current tunnel bandwidth. Range is from 10 to 4294967295 kilobits per second (kbps). The default value is 10 kbps.
Step 7	overflow threshold <i>percentage</i> [<i>min bandwidth</i>] limit <i>limit</i> Example: RP/0/RP0/CPU0:router(config-if-tunte-autobw) # overflow threshold 100 limit 1	Configures the tunnel overflow detection. percentage Bandwidth change percent to trigger an overflow. Range is from 1 to 100 percent.

	Command or Action	Purpose
		limit Configures the number of consecutive collection intervals that exceeds the threshold. The bandwidth overflow triggers an early tunnel bandwidth update. Range is from 1 to 10 collection periods. The default value is none. min Configures the bandwidth change value in kbps to trigger an overflow. Range is from 10 to 4294967295. The default value is 10.
Step 8	commit	
Step 9	show mpls traffic-eng tunnels [auto-bw] Example: <pre>RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-bw</pre>	Displays the MPLS-TE tunnel information only for tunnels in which the automatic bandwidth is enabled.

Related Topics

[MPLS-TE Automatic Bandwidth Overview, on page 28](#)
[Configure Automatic Bandwidth: Example, on page 176](#)

Configuring the Shared Risk Link Groups

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

Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link

Perform this task to configure the SRLG value for each link that has a shared risk with another link.



Note

You can configure up to 30 SRLGs per interface.

SUMMARY STEPS

1. **configure**
2. **srlg**
3. **interface** *type interface-path-id*
4. **value** *value*
5. **commit**
6. **show srlg interface** *type interface-path-id*
7. **show srlg**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	srlg Example: RP/0/RP0/CPU0:router(config)# srlg	Configures SRLG configuration commands on a specific interface configuration mode and assigns this SRLG a value.
Step 3	interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-srlg)# interface POS 0/6/0/0	Configures an interface type and path ID to be associated with an SRLG and enters SRLG interface configuration mode.
Step 4	value <i>value</i> Example: RP/0/RP0/CPU0:router(config-srlg-if)# value 100 RP/0/RP0/CPU0:router (config-srlg-if)# value 200 RP/0/RP0/CPU0:router (config-srlg-if)# value 300	Configures SRLG network values for a specific interface. Range is 0 to 4294967295. Note You can also set SRLG values on multiple interfaces including bundle interface.
Step 5	commit	
Step 6	show srlg interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router# show srlg interface POS 0/6/0/0	(Optional) Displays the SRLG values configured for a specific interface.
Step 7	show srlg Example: RP/0/RP0/CPU0:router# show srlg	(Optional) Displays the SRLG values for all the configured interfaces. Note You can configure up to 250 interfaces.

Related Topics

[MPLS Traffic Engineering Shared Risk Link Groups, on page 35](#)
[Explicit Path, on page 36](#)
[Fast ReRoute with SRLG Constraints, on page 36](#)
[Importance of Protection, on page 38](#)
[Delivery of Packets During a Failure, on page 39](#)
[Multiple Backup Tunnels Protecting the Same Interface , on page 39](#)
[SRLG Limitations, on page 39](#)
[MPLS TE SRLG Scale Enhancements, on page 40](#)
[Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Creating an Explicit Path With Exclude SRLG

Perform this task to create an explicit path with the exclude SRLG option.

SUMMARY STEPS

1. **configure**
2. **explicit-path {identifier number [disable | index]} { name *explicit-path-name*}**
3. **index 1 exclude-address 192.168.92.1**
4. **index 2 exclude-srlg 192.168.92.2**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	explicit-path {identifier number [disable index]} { name <i>explicit-path-name</i>} Example: RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg	Enters the explicit path configuration mode. Identifier range is 1 to 65535.
Step 3	index 1 exclude-address 192.168.92.1 Example: RP/0/RP0/CPU0:router router(config-expl-path)# index 1 exclude-address 192.168.92.1	Specifies the IP address to be excluded from the explicit path.
Step 4	index 2 exclude-srlg 192.168.92.2 Example: RP/0/RP0/CPU0:router(config-expl-path)# index 2 exclude-srlg 192.168.192.2	Specifies the IP address to extract SRLGs to be excluded from the explicit path.
Step 5	commit	

Related Topics

- [MPLS Traffic Engineering Shared Risk Link Groups, on page 35](#)
- [Explicit Path, on page 36](#)
- [Fast ReRoute with SRLG Constraints, on page 36](#)
- [Importance of Protection, on page 38](#)
- [Delivery of Packets During a Failure, on page 39](#)
- [Multiple Backup Tunnels Protecting the Same Interface , on page 39](#)
- [SRLG Limitations, on page 39](#)
- [MPLS TE SRLG Scale Enhancements, on page 40](#)
- [Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Using Explicit Path With Exclude SRLG

Perform this task to use an explicit path with the exclude SRLG option on the static backup tunnel.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **interface** *type interface-path-id*
4. **backup-path tunnel-te** *tunnel-number*
5. **exit**
6. **exit**
7. **interface tunnel-te***tunnel-id*
8. **ipv4 unnumbered** *type interface-path-id*
9. **path-option** *preference-priority*{ **dynamic** | **explicit** {*identifier* | **name** *explicit-path-name*}}
10. **destination** *ip-address*
11. **exit**
12. **commit**
13. **show run explicit-path** *name name*
14. **show mpls traffic-eng topology path destination** *name explicit-path name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
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-path-id Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0	Enables traffic engineering on a specific interface on the originating node.
Step 4	backup-path tunnel-te tunnel-number Example: RP/0/RP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2	Configures an MPLS TE backup path for a specific interface.
Step 5	exit Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# exit	Exits the current configuration mode.
Step 6	exit Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit	Exits the current configuration mode.
Step 7	interface tunnel-te tunnel-id Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 2	Configures an MPLS-TE tunnel interface.
Step 8	ipv4 unnumbered type interface-path-id Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address to set up forwarding on the new tunnel.
Step 9	path-option preference-priority{ dynamic explicit {identifier name explicit-path-name}} Example: RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg	Sets the path option to explicit with a given name (previously configured) and assigns the path ID. Note You can use the dynamic option to dynamically assign a path.
Step 10	destination ip-address Example: RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125	Assigns a destination address on the new tunnel. <ul style="list-style-type: none"> • Destination address is the remote node's MPLS-TE router ID. • Destination address is the merge point between backup and protected tunnels.

	Command or Action	Purpose
		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 11	exit Example: RP/0/RP0/CPU0:router(config-if)# exit	Exits the current configuration mode.
Step 12	commit	
Step 13	show run explicit-path name <i>name</i> Example: RP/0/RP0/CPU0:router# show run explicit-path name backup-srlg	Displays the SRLG values that are configured for the link.
Step 14	show mpls traffic-eng topology path destination <i>name</i> explicit-path <i>name</i> Example: RP/0/RP0/CPU0:router#show mpls traffic-eng topology path destination 192.168.92.125 explicit-path backup-srlg	Displays the SRLG values that are configured for the link.

Related Topics

[MPLS Traffic Engineering Shared Risk Link Groups, on page 35](#)
[Explicit Path, on page 36](#)
[Fast ReRoute with SRLG Constraints, on page 36](#)
[Importance of Protection, on page 38](#)
[Delivery of Packets During a Failure, on page 39](#)
[Multiple Backup Tunnels Protecting the Same Interface, on page 39](#)
[SRLG Limitations, on page 39](#)
[MPLS TE SRLG Scale Enhancements, on page 40](#)
[Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Creating a Link Protection on Backup Tunnel with SRLG Constraint

Perform this task to create an explicit path with the exclude SRLG option on the static backup tunnel.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **interface** *type interface-path-id*
4. **backup-path tunnel-te** *tunnel-number*
5. **exit**
6. **exit**
7. **interface tunnel-te***tunnel-id*
8. **ipv4 unnumbered** *type interface-path-id*
9. **path-option** *preference-priority* { **dynamic** | **explicit** { *identifier* | **name** *explicit-path-name* } }
10. **destination** *ip-address*
11. **exit**
12. **explicit-path** { *identifier number* [**disable** | **index**] } { **name** *explicit-path-name* }
13. **index 1** **exclude-srlg** 192.168.92.2
14. **commit**
15. **show mpls traffic-eng tunnel***tunnel-number* **detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0	Enables traffic engineering on a particular interface on the originating node.
Step 4	backup-path tunnel-te <i>tunnel-number</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2	Sets the backup path to the primary tunnel outgoing interface.
Step 5	exit Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# exit	Exits the current configuration mode.
Step 6	exit Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit	Exits the current configuration mode.

	Command or Action	Purpose
Step 7	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 2	Configures an MPLS-TE tunnel interface.
Step 8	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address to set up forwarding on the new tunnel.
Step 9	path-option <i>preference-priority</i> { dynamic explicit { identifier name <i>explicit-path-name</i> }} Example: RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg	Sets the path option to explicit with a given name (previously configured) and assigns the path ID. Identifier range is from 1 to 4294967295. Note You can use the dynamic option to dynamically assign a path.
Step 10	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125	Assigns a destination address on the new tunnel. <ul style="list-style-type: none"> • Destination address is the remote node's MPLS-TE router ID. • Destination address is the merge point between backup and protected tunnels. Note When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.
Step 11	exit Example: RP/0/RP0/CPU0:router(config-if)# exit	Exits the current configuration mode.
Step 12	explicit-path { identifier number [disable index]}{ name <i>explicit-path-name</i> } Example: RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg-nodep	Enters the explicit path configuration mode. Identifier range is 1 to 65535.
Step 13	index 1 exclude-srlg 192.168.92.2 Example: RP/0/RP0/CPU0:router:router(config-if)# index 1 exclude-srlg 192.168.192.2	Specifies the protected link IP address to get SRLGs to be excluded from the explicit path.
Step 14	commit	

	Command or Action	Purpose
Step 15	show mpls traffic-eng tunnel <i>tunnel-number</i> detail Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 2 detail	Display the tunnel details with SRLG values that are configured for the link.

Related Topics

[MPLS Traffic Engineering Shared Risk Link Groups, on page 35](#)
[Explicit Path, on page 36](#)
[Fast ReRoute with SRLG Constraints, on page 36](#)
[Importance of Protection, on page 38](#)
[Delivery of Packets During a Failure, on page 39](#)
[Multiple Backup Tunnels Protecting the Same Interface , on page 39](#)
[SRLG Limitations, on page 39](#)
[MPLS TE SRLG Scale Enhancements, on page 40](#)
[Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Creating a Node Protection on Backup Tunnel with SRLG Constraint

Perform this task to configure node protection on backup tunnel with SRLG constraint.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **interface** *type interface-path-id*
4. **backup-path tunnel-te** *tunnel-number*
5. **exit**
6. **exit**
7. **interface tunnel-te***tunnel-id*
8. **ipv4 unnumbered** *type interface-path-id*
9. **path-option** *preference-priority*{ **dynamic** | **explicit** {*identifier* | **name** *explicit-path-name*}}
10. **destination** *ip-address*
11. **exit**
12. **explicit-path** {*identifier number* [**disable** | **index**]}{ **name** *explicit-path-name*}
13. **index 1 exclude-address** 192.168.92.1
14. **index 2 exclude-srlg** 192.168.92.2
15. **commit**
16. **show mpls traffic-eng tunnels topology path destination** *ip-address explicit-path-name name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
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-path-id Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0	Enables traffic engineering on a particular interface on the originating node.
Step 4	backup-path tunnel-te tunnel-number Example: RP/0/RP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2	Sets the backup path for the primary tunnel outgoing interface.
Step 5	exit Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# exit	Exits the current configuration mode.
Step 6	exit Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit	Exits the current configuration mode.
Step 7	interface tunnel-te tunnel-id Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 2	Configures an MPLS-TE tunnel interface.
Step 8	ipv4 unnumbered type interface-path-id Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address to set up forwarding on the new tunnel.
Step 9	path-option preference-priority { dynamic explicit { identifier name explicit-path-name } } Example: RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg	Sets the path option to explicit with a given name (previously configured) and assigns the path ID. Identifier range is 1 to 4294967295. Note You can use the dynamic option to dynamically assign path.
Step 10	destination ip-address Example: RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125	Assigns a destination address on the new tunnel. • Destination address is the remote node's MPLS-TE router ID.

	Command or Action	Purpose
		<ul style="list-style-type: none"> Destination address is the merge point between backup and protected tunnels. <p>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.</p>
Step 11	exit Example: RP/0/RP0/CPU0:router(config-if)# exit	Exits the current configuration mode.
Step 12	explicit-path {identifier number [disable index]} {name explicit-path-name} Example: RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg-nodep	Enters the explicit path configuration mode. Identifier range is 1 to 65535.
Step 13	index 1 exclude-address 192.168.92.1 Example: RP/0/RP0/CPU0:router:router(config-if)# index 1 exclude-address 192.168.92.1	Specifies the protected node IP address to be excluded from the explicit path.
Step 14	index 2 exclude-srlg 192.168.92.2 Example: RP/0/RP0/CPU0:router(config-if)# index 2 exclude-srlg 192.168.192.2	Specifies the protected link IP address to get SRLGs to be excluded from the explicit path.
Step 15	commit	
Step 16	show mpls traffic-eng tunnels topology path destination ip-address explicit-path-name name Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels topology path destination 192.168.92.125 explicit-path-name backup-srlg-nodep	Displays the path to the destination with the constraint specified in the explicit path.

Related Topics

[MPLS Traffic Engineering Shared Risk Link Groups, on page 35](#)

[Explicit Path, on page 36](#)

[Fast ReRoute with SRLG Constraints, on page 36](#)

[Importance of Protection, on page 38](#)

[Delivery of Packets During a Failure, on page 39](#)

[Multiple Backup Tunnels Protecting the Same Interface, on page 39](#)

[SRLG Limitations, on page 39](#)

[MPLS TE SRLG Scale Enhancements, on page 40](#)
[Configure the MPLS-TE Shared Risk Link Groups: Example, on page 177](#)

Configuring Point-to-Multipoint TE

You must enable multicast routing on the edge router before performing Point-to-Multipoint (P2MP) TE configurations. To configure Point-to-Multipoint TE, perform these procedures:

Enabling Multicast Routing on the Router

Perform this task to enable multicast routing on the router to configure P2MP tunnels.

Before You Begin

- To configure Point-to-Multipoint (P2MP) tunnels, you must enable multicast routing on the router.
- The customer-facing interface must enable multicast.

SUMMARY STEPS

1. **configure**
2. **multicast-routing**
3. **address-family {ipv4 | ipv6 }**
4. **interface tunnel-mte *tunnel-id***
5. **enable**
6. **exit**
7. **interface *type interface-path-id***
8. **enable**
9. **commit**
10. **show pim ipv6 interface *type interface-path-id***

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	multicast-routing Example: RP/0/RP0/CPU0:router(config)# multicast-routing RP/0/RP0/CPU0:router(config-mcast)#	Enters multicast routing configuration mode.

	Command or Action	Purpose
Step 3	address-family {ipv4 ipv6 } Example: RP/0/RP0/CPU0:router(config-mcast)# address-family ipv6 RP/0/RP0/CPU0:router(config-mcast-default-ipv6)#	Configures the available IPv4 or IPv6 address prefixes to enable multicast routing and forwarding on all router interfaces.
Step 4	interface tunnel-mte <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6)# interface tunnel-mte 1 RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)#	Configures an MPLS-TE P2MP tunnel interface.
Step 5	enable Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)# enable	Enables multicast routing on the tunnel-mte interface.
Step 6	exit Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)# exit RP/0/RP0/CPU0:router(config-mcast-default-ipv6)#	Exits the current configuration mode.
Step 7	interface type <i>interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6)# interface GigabitEthernet0/2/0/3 RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)#	Configures multicast routing on the GigabitEthernet interface.
Step 8	enable Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)# enable	Enables multicast routing on the GigabitEthernet interface.
Step 9	commit	
Step 10	show pim ipv6 interface type <i>interface-path-id</i> Example: RP/0/RP0/CPU0:router# show pim ipv6 interface	Displays the output for the P2MP-TE tunnel interface that has IPv6 multicast enabled.

	Command or Action	Purpose
	<code>tunnel-mte 1</code>	

Related Topics

[Point-to-Multipoint Traffic-Engineering Overview](#), on page 30

[Configure Point-to-Multipoint for the Source: Example](#), on page 186

[Configure the Point-to-Multipoint Solution: Example](#), on page 188

[Configuring the Static Group for the Point-to-Multipoint Interface](#), on page 130

Configuring the Static Group for the Point-to-Multipoint Interface

Perform this task to configure the static group on the Point-to-Multipoint (P2MP) interface to forward specified multicast traffic over P2MP LSP.

SUMMARY STEPS

1. `configure`
2. `router mld`
3. `vrf vrf-name`
4. `interface tunnel-mte tunnel-id`
5. `static-group group-address`
6. `commit`
7. `show mrrib ipv6 route source-address`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>configure</code>	
Step 2	router mld Example: RP/0/RP0/CPU0:router(config)# router mld RP/0/RP0/CPU0:router(config-mld)#	Enters router MLD configuration mode.
Step 3	vrf vrf-name Example: RP/0/RP0/CPU0:router(config-mld)# vrf default RP/0/RP0/CPU0:router(config-mld-default)#	Configures a virtual private network (VRF) instance.

	Command or Action	Purpose
Step 4	interface tunnel-mte <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config-mld-default)# interface tunnel-mte 1 RP/0/RP0/CPU0:router(config-mld-default-if)#	Configures an MPLS-TE P2MP tunnel interface.
Step 5	static-group <i>group-address</i> Example: RP/0/RP0/CPU0:router(config-mld-default-if)# static-group ff35::1 2000::1	Configures the multicast group address in the Source-Specific Multicast (SSM) address range (ff35::/16) for the IPv6 address prefix.
Step 6	commit	
Step 7	show mrib ipv6 route <i>source-address</i> Example: RP/0/RP0/CPU0:router# show mrib ipv6 route ff35::1	Verifies the multicast static mapping.

Related Topics

[Point-to-Multipoint RSVP-TE , on page 32](#)

[Configure Point-to-Multipoint for the Source: Example, on page 186](#)

[Configure the Point-to-Multipoint Solution: Example, on page 188](#)

[Enabling Multicast Routing on the Router, on page 128](#)

Configuring Destinations for the Tunnel Interface

Perform this task to configure three destinations for the tunnel interface for Point-to-Multipoint (P2MP).

These variations are listed to ensure that the destination and path option configurations are separate from the tunnel interface.

- Different path option is used for different destinations. This task shows three destinations.
- Explicit path option is based on an ID or a name.
- Default path option is similar to the Point-to-Point (P2P) LSP.

Before You Begin

These prerequisites are required to configure destinations for the tunnel interface.

- Multicast routing must be enabled on both the tunnel-mte interface and customer-facing interface from the source.

- Static-group must be configured on the tunnel-mte interface to forward specified multicast traffic over P2MP LSP.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-mte** *tunnel-id*
3. **destination** *ip-address*
4. **path-option** *preference-priority* **explicit identifier** *path-number*
5. **path-option** *preference-priority* **dynamic**
6. **exit**
7. **destination** *ip-address*
8. **path-option** *preference-priority* **explicit name** *pathname*
9. **path-option** *preference-priority* **dynamic**
10. **exit**
11. **destination** *ip-address*
12. **path-option** *preference-priority* **explicit name** *pathname* [**verbatim**]
13. **commit**
14. **show mpls traffic-eng tunnels** [**brief**] [**p2mp** *tunnel-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-mte <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-mte 10 RP/0/RP0/CPU0:router(config-if)#	Configures an MPLS-TE P2MP tunnel interface.
Step 3	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)# destination 172.16.255.1 RP/0/RP0/CPU0:router(config-if-p2mp-dest)#	Sets the destination address for tunnel-mte 10 to 172.16.255.1. This destination uses the explicit path identified by explicit path ID 10. If destination 172.16.255.1 cannot come with explicit path ID 10, the fall back path option is dynamic.
Step 4	path-option <i>preference-priority</i> explicit identifier <i>path-number</i> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)#	Configures the path number of the IP explicit path.

	Command or Action	Purpose
	<code>path-option 1 explicit identifier 10</code>	
Step 5	<code>path-option preference-priority dynamic</code> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# <code>path-option 2 dynamic</code>	Specifies that label switched paths (LSP) are dynamically calculated.
Step 6	<code>exit</code> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# <code>exit</code> RP/0/RP0/CPU0:router(config-if)#	Exits the current configuration mode.
Step 7	<code>destination ip-address</code> Example: RP/0/RP0/CPU0:router(config-if)# <code>destination 172.16.255.2</code> RP/0/RP0/CPU0:router(config-if-p2mp-dest)#	Sets the destination address for tunnel-mte 10 to 172.16.255.2.
Step 8	<code>path-option preference-priority explicit name pathname</code> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# <code>path-option 1 explicit name how-to-get-to-172.16.255.2</code>	Specifies the path name of the IP explicit path. Destination 172.16.255.2 uses the explicit path that is identified by the explicit path name "how-to-get-to-172.16.255.2."
Step 9	<code>path-option preference-priority dynamic</code> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# <code>path-option 2 dynamic</code>	Sets the fall back path option as dynamic when the destination cannot come to the explicit path.
Step 10	<code>exit</code> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# <code>exit</code> RP/0/RP0/CPU0:router(config-if)#	Exits the current configuration mode.
Step 11	<code>destination ip-address</code> Example: RP/0/RP0/CPU0:router(config-if)# <code>destination 172.16.255.3</code>	Specifies that destination 172.16.255.3 uses only the dynamically computed path.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-if-p2mp-dest)#	
Step 12	path-option <i>preference-priority</i> explicit name <i>pathname</i> [verbatim] Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 explicit name how-to-get-to-172.16.255.3 verbatim	Specifies that destination 172.16.255.3 uses the explicit path identified by the explicit path name "how-to-get-to-172.16.255.3" in verbatim mode.
Step 13	commit	
Step 14	show mpls traffic-eng tunnels [brief] [p2mp tunnel-number] Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels brief p2mp 10	Displays the brief summary of the P2MP tunnel status and configuration.

Related Topics

[Path Option for Point-to-Multipoint RSVP-TE, on page 34](#)

[Configure the Point-to-Multipoint Tunnel: Example, on page 187](#)

[Configure the Point-to-Multipoint Solution: Example, on page 188](#)

[Enabling Multicast Routing on the Router, on page 128](#)

[Configuring the Static Group for the Point-to-Multipoint Interface, on page 130](#)

Disabling Destinations

Perform this task to disable the given destination for the Point-to-Multipoint (P2MP) tunnel interface.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-mte** *tunnel-id*
3. **ipv4 unnumbered** *type interface-path-id*
4. **destination** *ip-address*
5. **disable**
6. **path-option** *preference-priority* **dynamic**
7. **path-option** *preference-priority* **explicit name** *pathname*
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-mte <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-mte 101 RP/0/RP0/CPU0:router(config-if)#	Configures an MPLS-TE P2MP tunnel interface.
Step 3	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.
Step 4	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)# destination 140.140.140.140 RP/0/RP0/CPU0:router(config-if-p2mp-dest)#	Sets the destination address for tunnel-mte 10 to 140.140.140.140.
Step 5	disable Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# disable	Disables destination 140.140.140.140 for tunnel-mte 10.
Step 6	path-option <i>preference-priority</i> dynamic Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 dynamic	Specifies that label switched paths (LSP) are dynamically calculated.
Step 7	path-option <i>preference-priority</i> explicit name <i>pathname</i> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 2 explicit name to4	Specifies that destination 140.140.140.140 uses the explicit path identified by the explicit path name "to4."
Step 8	commit	

Related Topics

- [Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)
- [Disable a Destination: Example, on page 187](#)

Logging Per Destinations for Point-to-Multipoint

Perform this task to log destinations for Point-to-Multipoint (P2MP).

SUMMARY STEPS

1. **configure**
2. **interface tunnel-mte** *tunnel-id*
3. **ipv4 unnumbered** *type interface-path-id*
4. **destination** *ip-address*
5. **logging events lsp-status** *state*
6. **logging events lsp-status** *reroute*
7. **path-option** *preference-priority explicit name pathname*
8. **exit**
9. **fast-reroute**
10. **commit**
11. **show mpls traffic-eng tunnels** [p2mp]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-mte <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)# interface tunnel-mte 1000 RP/0/RP0/CPU0:router(config-if)#	Configures an MPLS-TE P2MP tunnel interface.
Step 3	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0	Configures the MPLS-TE tunnel to use the IPv4 address on loopback interface 0.
Step 4	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)# destination	Sets the destination address for tunnel-mte from 1000 to 100.0.0.3.

	Command or Action	Purpose
	100.0.0.3 RP/0/RP0/CPU0:router(config-if-p2mp-dest)#	
Step 5	logging events lsp-status state Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# logging events lsp-status state	Sends out the log message when the tunnel LSP goes up or down when the software is enabled.
Step 6	logging events lsp-status reroute Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# logging events lsp-status reroute	Sends out the log message when the tunnel LSP is rerouted due to an FRR event when the software is enabled.
Step 7	path-option <i>preference-priority</i> explicit name <i>pathname</i> Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 explicit name path123	Specifies the path name of the IP explicit path. Destination 100.0.0.3 uses the explicit path that is identified by the explicit path name "path123."
Step 8	exit Example: RP/0/RP0/CPU0:router(config-if-p2mp-dest)# exit RP/0/RP0/CPU0:router(config-if)#	Exits the current configuration mode.
Step 9	fast-reroute Example: RP/0/RP0/CPU0:router(config-if)# fast-reroute	Enables fast-reroute (FRR) protection for a P2MP TE tunnel.
Step 10	commit	
Step 11	show mpls traffic-eng tunnels [p2mp] Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels p2mp	Displays the information for all P2MP tunnels.

Related Topics

[Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)

[Configure the Point-to-Multipoint Tunnel: Example, on page 187](#)

[Configure the Point-to-Multipoint Solution: Example, on page 188](#)

Enabling Soft-Preemption on a Node

Perform this task to enable the soft-preemption feature in the MPLS TE configuration mode. By default, this feature is disabled. You can configure the soft-preemption feature for each node. It has to be explicitly enabled for each node.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **soft-preemption**
4. **timeout** *seconds*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	soft-preemption Example: RP/0/RP0/CPU0:router(config-mpls-te)# soft-preemption	Enables soft-preemption on a node. Note If soft-preemption is enabled, the head-end node tracks whether an LSP desires the soft-preemption treatment. However, when a soft-preemption feature is disabled on a node, this node continues to track all LSPs desiring soft-preemption. This is needed in a case when soft-preemption is re-enabled, TE will have the property of the existing LSPs without any re-signaling.
Step 4	timeout <i>seconds</i> Example: RP/0/RP0/CPU0:router(config-soft-preemption)# timeout 20	Specifies the timeout for the soft-preempted LSP, in seconds. The range is from 1 to 300.
Step 5	commit	

Related Topics[Soft-Preemption, on page 40](#)

Enabling Soft-Preemption on a Tunnel

Perform this task to enable the soft-preemption feature on a MPLS TE tunnel. By default, this feature is disabled. It has to be explicitly enabled.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te *tunnel-id***
3. **soft-preemption**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router# interface tunnel-te 10	Configures an MPLS-TE tunnel interface.
Step 3	soft-preemption Example: RP/0/RP0/CPU0:router(config-if)# soft-preemption	<p>Enables soft-preemption on a tunnel.</p> <p>When soft preemption is enabled on a tunnel, these actions occur:</p> <ul style="list-style-type: none"> • A path-modify message is sent for the current LSP with the soft preemption desired property. • A path-modify message is sent for the reopt LSP with the soft preemption desired property. • A path-modify message is sent for the path protection LSP with the soft preemption desired property. • A path-modify message is sent for the current LSP in FRR active state with the soft preemption desired property. <p>Note The soft-preemption is not available in the interface tunnel-mte and interface tunnel-gte configuration modes.</p>
Step 4	commit	

Related Topics[Soft-Preemption, on page 40](#)

Configuring Attributes within a Path-Option Attribute

Perform this task to configure attributes within a path option attribute-set template.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **attribute-set path-option** *attribute-set-name*
4. **affinity** *affinity-value* **mask** *mask-value*
5. **signalled-bandwidth** *kbits* **class-type** *class-type number*
6. **commit**
7. **show mpls traffic-eng attribute-set**
8. **show mpls traffic-eng tunnels***detail*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	attribute-set path-option <i>attribute-set-name</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# attribute-set path-option myset	Enters attribute-set path option configuration mode. Note The configuration at the path-option level takes precedence over the values configured at the level of the tunnel, and therefore is applied.
Step 4	affinity <i>affinity-value</i> mask <i>mask-value</i> Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# affinity 0xBEEF mask 0xBEEF	Configures affinity attribute under a path option attribute-set. The attribute values that are required for links to carry this tunnel.
Step 5	signalled-bandwidth <i>kbits</i> class-type <i>class-type number</i> Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# signalled-bandwidth 1000 class-type 0	Configures the bandwidth attribute required for an MPLS-TE tunnel under a path option attribute-set. Note You can configure the class type of the tunnel bandwidth request. The class-type 0 is strictly equivalent to global-pool and class-type 1 is strictly equivalent to subpool .
Step 6	commit	

	Command or Action	Purpose
Step 7	show mpls traffic-eng attribute-set Example: RP/0/RP0/CPU0:router# show mpls traffic-eng attribute-set	Displays the attributes that are defined in the attribute-set for the link.
Step 8	show mpls traffic-eng tunnels <i>detail</i> Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels detail	Displays the attribute-set path option information on a specific tunnel.

Related Topics

[Path Option Attributes, on page 41](#)

[Configuration Hierarchy of Path Option Attributes, on page 41](#)

[Traffic Engineering Bandwidth and Bandwidth Pools, on page 42](#)

[Path Option Switchover, on page 42](#)

[Path Option and Path Protection, on page 43](#)

Configuring Auto-Tunnel Mesh Tunnel ID

Perform this activity to configure the tunnel ID range that can be allocated to Auto-tunnel mesh tunnels.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **auto-tunnel mesh**
4. **tunnel-id min *value* max *value***
5. **commit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	auto-tunnel mesh Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel mesh	Enters auto-tunnel mesh configuration mode. You can configure auto-tunnel mesh related options from this mode.
Step 4	tunnel-id min <i>value</i> max <i>value</i> Example: RP/0/RP0/CPU0:router(config-te-auto-mesh)# tunnel-id min 10 max 50	Specifies the minimum and maximum number of auto-tunnel mesh tunnels that can be created on this router. The range of tunnel ID is from 0 to 65535.
Step 5	commit	

Related Topics

[Auto-Tunnel Mesh](#), on page 43

[Destination List \(Prefix-List\)](#), on page 44

Configuring Auto-tunnel Mesh Unused Timeout

Perform this task to configure a global timer to remove unused auto-mesh tunnels.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **auto-tunnel mesh**
4. **timer removal unused *timeout***
5. **commit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	auto-tunnel mesh Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel mesh	Enables auto-tunnel mesh groups globally.
Step 4	timer removal unused <i>timeout</i> Example: RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# timers removal unused 10	<p>Specifies a timer, in minutes, after which a down auto-tunnel mesh gets deleted whose destination was not in TE topology. The default value for this timer is 60.</p> <p>The timer gets started when these conditions are met:</p> <ul style="list-style-type: none"> • Tunnel destination node is removed from the topology • Tunnel is in down state <p>Note The unused timer runs per tunnel because the same destination in different mesh-groups may have different tunnels created.</p>
Step 5	commit	

Related Topics

[Auto-Tunnel Mesh, on page 43](#)

[Destination List \(Prefix-List\), on page 44](#)

Configuring Auto-Tunnel Mesh Group

Perform this task to configure an auto-tunnel mesh group globally on the router.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **auto-tunnel mesh**
4. **group *value***
5. **disable**
6. **attribute-set *name***
7. **destination-list**
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	auto-tunnel mesh Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel mesh	Enables auto-tunnel mesh groups globally.
Step 4	group value Example: RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# group 65	Specifies the membership of auto-tunnel mesh. The range is from 0 to 4294967295. Note When the destination-list is not supplied, head-end will automatically build destination list belonging for the given mesh-group membership using TE topology.
Step 5	disable Example: RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# disable	Disables the meshgroup and deletes all tunnels created for this meshgroup.
Step 6	attribute-setname Example: RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# attribute-set am-65	Specifies the attributes used for all tunnels created for the meshgroup. If it is not defined, this meshgroup does not create any tunnel.
Step 7	destination-list Example: RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# destination-list dl-65	This is a mandatory configuration under a meshgroup. If a given destination-list is not defined as a prefix-list, this meshgroup create tunnels to all nodes available in TE topology.
Step 8	commit	

Related Topics

[Auto-Tunnel Mesh, on page 43](#)

[Destination List \(Prefix-List\), on page 44](#)

Configuring Tunnel Attribute-Set Templates

Perform this task to define attribute-set templates for auto-mesh tunnels.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **attribute-set auto-mesh** *attribute-set-name*
4. **affinity** *value mask mask-value*
5. **signalled-bandwidth** *kbps class-type class-type number*
6. **autoroute announce**
7. **fast-reroute protect bandwidth node**
8. **auto-bw collect-bw-only**
9. **logging events lsp-status** {*state* | *insufficient-bandwidth* | *reoptimize* | *reroute* }
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	attribute-set auto-mesh <i>attribute-set-name</i> Example: RP/0/RP0/CPU0:router(config-te)# attribute-set auto-mesh attribute-set-mesh	Specifies name of the attribute-set of auto-mesh type.
Step 4	affinity <i>value mask mask-value</i> Example: RP/0/RP0/CPU0:router(config-te)# affinity 0101 mask 320	Configures the affinity properties the tunnel requires in its links for an MPLS-TE tunnel under an auto-mesh attribute-set.
Step 5	signalled-bandwidth <i>kbps class-type class-type number</i> Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# signalled-bandwidth 1000 class-type 0	Configures the bandwidth attribute required for an MPLS-TE tunnel under an auto-mesh attribute-set. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 0, priority 7).

	Command or Action	Purpose
		Note You can configure the class type of the tunnel bandwidth request. The class-type 0 is strictly equivalent to global-pool and class-type 1 is strictly equivalent to subpool .
Step 6	autoroute announce Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# autoroute announce	Enables parameters for IGP routing over tunnel.
Step 7	fast-reroute protect bandwidth node Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# fast-reroute	Enables fast-reroute bandwidth protection and node protection for auto-mesh tunnels.
Step 8	auto-bw collect-bw-only Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# auto-bw collect-bw-only	Enables automatic bandwidth collection frequency, and controls the manner in which the bandwidth for a tunnel collects output rate information, but does not adjust the tunnel bandwidth.
Step 9	logging events lsp-status {state insufficient-bandwidth reoptimize reroute } Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# logging events lsp-status state	Sends out the log message when the tunnel LSP goes up or down when the software is enabled. Sends out the log message when the tunnel LSP undergoes setup or reoptimize failure due to bandwidth issues. Sends out the log message for the LSP reoptimize change alarms. Sends out the log message for the LSP reroute change alarms.
Step 10	commit	

Related Topics

[Auto-Tunnel Mesh](#), on page 43

[Destination List \(Prefix-List\)](#), on page 44

Enabling LDP on Auto-Tunnel Mesh

Perform this task to enable LDP on auto-tunnel mesh group.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **traffic-eng auto-tunnel mesh**
4. **groupidall**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls ldp Example: RP/0/RP0/CPU0:router(config-ldp)# mpls ldp	Enters MPLS LDP configuration mode.
Step 3	traffic-eng auto-tunnel mesh Example: RP/0/RP0/CPU0:router(config-ldp-te-auto-mesh)# traffic-eng auto-tunnel mesh	Enters auto-tunnel mesh configuration mode. You can configure TE auto-tunnel mesh groups from this mode.
Step 4	groupidall Example: RP/0/RP0/CPU0:router(config-ldp-te-auto-mesh)# group all	Configures an auto-tunnel mesh group of interfaces in LDP. You can enable LDP on all TE meshgroup interfaces or you can specify the TE mesh group ID on which the LDP is enabled. The range of group ID is from 0 to 4294967295.
Step 5	commit	

Related Topics

[Auto-Tunnel Mesh, on page 43](#)

[Destination List \(Prefix-List\), on page 44](#)

Enabling Stateful PCE Client

Perform these steps to enable stateful PCE client.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **pce**
4. **stateful-client**
5. **capabilities { instantiation | update }**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)#mpls traffic-eng	Enters MPLS TE configuration mode.
Step 3	pce Example: RP/0/RP0/CPU0:router(config-mpls-te)#pce	Enters PCE configuration mode.
Step 4	stateful-client Example: RP/0/RP0/CPU0:router(config-mpls-te-pce)#stateful-client	Enters stateful PCE client configuration mode. When the stateful-client configuration is added to the node, it will close all existing PCEP peer connections, and add the stateful capabilities TLV to the OPEN object it exchanges during the PCEP session establishment. When the stateful-client configuration is removed from the node, it will delete all PCE instantiated tunnels, close all existing PCEP connections, and no longer add the stateful capabilities TLV to the OPEN object it exchanges during the PCEP session establishment.
Step 5	capabilities { instantiation update } Example: RP/0/RP0/CPU0:router(config-mpls-te-pce-stateful)#capabilities instantiation	Enables stateful client capabilities. <ul style="list-style-type: none"> • instantiation—enables stateful instantiate capability • update—enables stateful update capability
Step 6	commit	

Configuring VRF Redirection

Perform these steps to configure VRF redirection by installing multiple routes in the routing information base (RIB) per MPLS TE tunnel:

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te *tunnel-id***
3. **autoroute destination *ip-address***
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)#interface tunnel-te 10	Configures an MPLS-TE tunnel interface.
Step 3	autoroute destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)#autoroute destination 192.168.1.2 RP/0/RP0/CPU0:router(config-if)#autoroute destination 192.168.2.2 RP/0/RP0/CPU0:router(config-if)#autoroute destination 192.168.3.2 RP/0/RP0/CPU0:router(config-if)#autoroute destination 192.168.4.2	Adds a route (ip-address) in RIB with TE tunnel as outgoing interface. to the tunnel destination.
Step 4	commit	

This example shows how to configure installing four autoroute destination routes into the RIB along with the default route:

```
interface tunnel-te10
  autoroute destination 192.168.1.2
  autoroute destination 192.168.2.2
  autoroute destination 192.168.3.2
  autoroute destination 192.168.4.2
```

Configuring IPv6 Routing Over IPv4 MPLS-TE Tunnels

Perform these steps to configure IPv6 routing over IPv4 MPLS-TE tunnels:

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *tunnel-id*
3. **ipv4 unnumbered** *type interface-path-id*
4. **ipv6 enable**
5. **signalled-bandwidth** *bandwidth*
6. **destination** *ip-address*
7. Use one of these options:
 - **autoroute announce include-ipv6**
 - **forwarding-adjacency include-ipv6**
8. **path-option** *preference-priority* **dynamic**
9. **commit**
10. (Optional) **show mpls traffic-eng autoroute**
11. (Optional) **show mpls traffic-eng forwarding-adjacency**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router# interface tunnel-te 1	Configures an MPLS-TE tunnel interface.
Step 3	ipv4 unnumbered <i>type interface-path-id</i> 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. Loopback is the commonly-used interface type.
Step 4	ipv6 enable Example: RP/0/RP0/CPU0:router(config-if)#ipv6 enable	Enables IPv6 on interface.
Step 5	signalled-bandwidth <i>bandwidth</i> Example: RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10	Sets the tunnel bandwidth requirement to be signalled in Kbps.
Step 6	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)#destination 3.3.3.3	Specifies tunnel destination.

	Command or Action	Purpose
Step 7	<p>Use one of these options:</p> <ul style="list-style-type: none"> • autoroute announce include-ipv6 • forwarding-adjacency include-ipv6 <p>Example: RP/0/RP0/CPU0:router(config-if)#autoroute announce include-ipv6 Or RP/0/RP0/CPU0:router(config-if)#forwarding-adjacency include-ipv6</p>	Announces the tunnel as an IPv6 autoroute or an IPv6 forwarding adjacency.
Step 8	<p>path-option preference-priority dynamic</p> <p>Example: RP/0/RP0/CPU0:router(config-if)#path-option 1 dynamic</p>	Sets the path option to dynamic and assigns the path ID.
Step 9	commit	
Step 10	<p>show mpls traffic-eng autoroute</p> <p>Example: RP/0/RP0/CPU0:router#show mpls traffic-eng autoroute</p> <pre> Destination 192.168.0.2 has 1 tunnels in IS-IS ring level 1 tunnel-te1 (traffic share 0, nexthop 192.168.0.2) (IPv4 unicast) (IPv6 unicast) </pre>	(Optional) Verifies that the tunnel announces IPv6 autoroute information.
Step 11	<p>show mpls traffic-eng forwarding-adjacency</p> <p>Example: RP/0/RP0/CPU0:router#show mpls traffic-eng forwarding-adjacency</p> <pre> destination 3.3.3.3 has 1 tunnels 3.3.3.3 tunnel-te10 (traffic share 0, next-hop (Adjacency Announced: yes, holdtime 0) (IS-IS 100, IPv4 unicast) (IS-IS 100, IPv6 unicast) </pre>	(Optional) Verifies that the tunnel announces IPv6 forwarding adjacency information.

Configuring Path-selection Cost Limit

Apply the path-selection cost-limit configuration to set the upper limit on the path aggregate admin-weight when computing paths for MPLS-TE LSPs. Once the path-selection cost is configured, the periodic path verification will check if the cost-limit is crossed. Path-selection cost limit can be configured at global MPLS

TE, per interface tunnel, and per path-option attribute set. The path-selection cost limit per path-option attribute set takes the highest priority, followed by per interface and MPLS TE global path-selection cost limit values.

Configuring Global Path-selection Cost Limit on MPLS TE Tunnels

Perform these steps to configure path-selection cost limit globally for MPLS TE tunnels:

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **path-selection cost-limit *cost-limit***
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	path-selection cost-limit <i>cost-limit</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# path-selection cost-limit 3	Sets the upper limit on the path aggregate admin-weight when computing paths for MPLS TE LSPs.
Step 4	commit	

Configuring Path-selection Cost Limit per TE Tunnel

Perform these steps to configure path-selection cost limit per MPLS TE tunnel:

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te *tunnel-id***
3. **path-selection cost-limit *cost-limit***
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RP0/CPU0:router(config)#interface tunnel-te 1	Configures an MPLS-TE tunnel interface.
Step 3	path-selection cost-limit <i>cost-limit</i> Example: RP/0/RP0/CPU0:router(config-if)# path-selection cost-limit 2	Sets the upper limit on the path aggregate admin-weight when computing paths for MPLS TE LSPs for the specified MPLS TE tunnel.
Step 4	commit	

Configuring Path-selection Cost Limit per Path-option Attribute-set

Perform these steps to configure path-selection cost limit per path-option attribute-set:

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **attribute-set path-option** *attribute-set-name*
4. **path-selection cost-limit** 3

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	attribute-set path-option <i>attribute-set-name</i> Example: RP/0/RP0/CPU0:router(config-mpls-te)# attribute-set path-option PO3AttrSet	Enters attribute-set path option configuration mode. Note The configuration at the attribute-set path-option level takes precedence over the values configured at global and interface tunnel level.

	Command or Action	Purpose
Step 4	path-selection cost-limit 3 Example: RP/0/RP0/CPU0:router(config-te-attribute-set) # path-selection cost-limit 3	Sets the upper limit on the path aggregate admin-weight when computing paths for MPLS TE LSPs per path-option attribute set.

Enabling Soft-preemption over FRR Backup Tunnels

Perform these tasks to enable LSP traffic to be moved over the backup tunnel when the LSP is soft-preempted. With this configuration, when there is a soft-preemption, the MPLS TE process triggers a rewrite to move the traffic on the backup tunnel, if the backup tunnel is ready. The rest of the soft-preemption process remains unchanged.

Before You Begin

Ensure that the following configurations are enabled before enabling soft-preemption over FRR backup:

- Soft-preemption enabled.
- Fast-reroute (FRR) backup tunnel is activated.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng**
3. **soft-preemption frr-rewrite**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)#mpls traffic-eng	Enters MPLS-TE configuration mode.
Step 3	soft-preemption frr-rewrite Example: RP/0/RP0/CPU0:router(config-mpls-te)#soft-preemption frr-rewrite	Moves FRR LSP traffic over the backup tunnel, when LSP is soft-preempted.
Step 4	commit	

Enabling Auto-onehop Tunnels to Next-hop Neighbors

Perform these tasks to enable automatic creation of one-hop tunnels over MPLS traffic-engineering enabled interfaces to nexthop neighbors. A router that becomes a next hop neighbor will have a set of one-hop tunnels created automatically.

Before You Begin

The **ipv4 unnumbered mpls traffic-eng Loopback *Number*** configuration must be applied at the global configuration level.

SUMMARY STEPS

1. **configure**
2. **ipv4 unnumbered mpls traffic-eng Loopback *N***
3. **mpls traffic-eng**
4. **auto-tunnel mesh**
5. **tunne-id min *value* max *value***
6. **group *group-id***
7. **onehop**
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	ipv4 unnumbered mpls traffic-eng Loopback <i>N</i> Example: RP/0/RP0/CPU0:router(config)#ipv4 unnumbered mpls traffic-eng loopback 0	Configures the globally configured IPv4 address that can be used by the Auto-tunnel backup tunnels.
Step 3	mpls traffic-eng Example: RP/0/RP0/CPU0:router(config)#mpls traffic-eng	Enters the MPLS-TE submode.
Step 4	auto-tunnel mesh Example: RP/0/RP0/CPU0:router(config-mpls-te)#auto-tunnel mesh	Enters the auto-tunnel mesh configuration submode.

	Command or Action	Purpose
Step 5	tunnel-id min value max value Example: RP/0/0/CPU0:ios(config-te-auto-mesh)# tunnel-id min 4000 max 6000	Specifies the minimum and maximum number of auto-tunnel mesh tunnels that can be created on this router. The range of tunnel ID is from 0 to 65535.
Step 6	group group-id Example: RP/0/RP0/CPU0:router(config-te-auto-mesh)#group 50	Enters the auto-tunnel mesh group configuration submode and creates a group ID.
Step 7	onehop Example: RP/0/RP0/CPU0:router(config-te-mesh-group)#onehop	Enables automatic creation of one-hop tunnels to all next hop neighbors. The onehop keyword can be applied to as many mesh groups as desired.
Step 8	commit	

Implementing Associated Bidirectional Label Switched Paths

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

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

[Signaling Methods and Object Association for Bidirectional LSPs, on page 156](#), [Associated Bidirectional Non Co-routed and Co-routed LSPs, on page 158](#) provides details.

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

[Path Protection, on page 161](#) provides details.

Signaling Methods and Object Association for Bidirectional LSPs

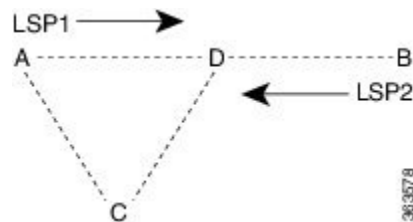
This section provides an overview of the association signaling methods for the bidirectional LSPs. Two unidirectional LSPs can be bound to form an associated bidirectional LSP in the following scenarios:

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

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

- **Single-sided Provisioning:** For the single-sided provisioning, the TE tunnel is configured only on one side. An LSP for this tunnel is initiated by the initiating endpoint with the Association Object inserted in the Path message. The other endpoint then creates the corresponding reverse TE tunnel and signals the reverse LSP in response to this. Currently, there is no support available for configuring single-sided provisioning.
- **Double-sided Provisioning:** For the double-sided provisioning, two unidirectional TE tunnels are configured independently on both sides. The LSPs for the tunnels are signaled with Association Objects inserted in the Path message by both sides to indicate that the two LSPs are to be associated to form a bidirectional LSP.

Consider this topology (an example of associated bidirectional LSP):



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

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

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



Note

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

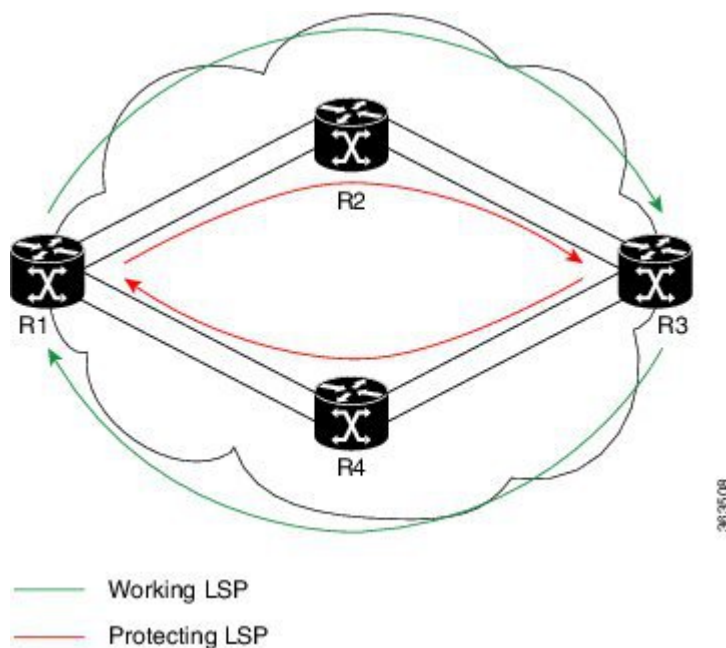
[Configure Associated Bidirectional Co-routed LSPs](#), on page 159 describes the procedure to create associated bidirectional co-routed LSPs.

Associated Bidirectional Non Co-routed and Co-routed LSPs

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

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

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



In the above topology:

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

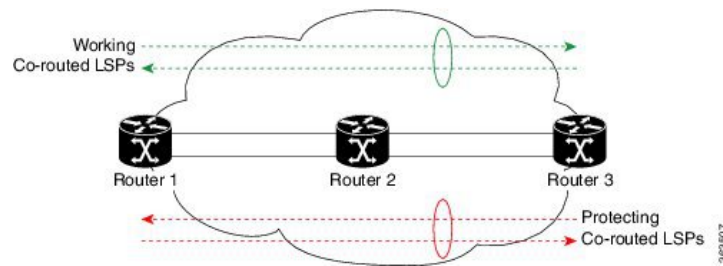
Non co-routed bidirectional TE LSP is available by default, and no configuration is required.



Note

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

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



In the above topology:

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

[Configure Associated Bidirectional Co-routed LSPs](#), on page 159 describes the procedure to configure an associated bidirectional co-routed LSP.

Configure Associated Bidirectional Co-routed LSPs

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

Before You Begin

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RSP0/CPU0:router# interface tunnel-te 1	Configures an MPLS-TE tunnel interface.
Step 3	bidirectional Example: RP/0/0/CPU0:router(config-if)# bidirectional	Configure the ingress router for the LSP and include the bidirectional statement to specify that the LSP be established as a bidirectional LSP.
Step 4	association {id <0-65535> source-address <IP address>} [global-id <0-4294967295>] Example: RP/0/0/CPU0:router(config-if-bidir)# association id 1 source-address 11.0.0.1	Set the association ID that uniquely identifies the association of LSPs, which is the tunnel ID of the bound LSP (master/slave mode) or the tunnel ID of the binding LSP (peer mode). Also, set the source address to the tunnel sender address of the bound LSP (master/slave mode) or the tunnel sender address of the binding LSP (peer mode). Optionally, specify the global ID for association global source. Note Association ID, association source and global ID <u>must be configured identically on both the endpoints.</u>
Step 5	association type co-routed Example: RP/0/0/CPU0:router(config-if-bidir)#association type co-routed	Specify that the LSP be established as a associated co-routed bidirectional LSP.
Step 6	commit	
Step 7	show mpls traffic-eng tunnels bidirectional-associated co-routed Example: RP/0/0/CPU0:router#show mpls traffic-eng tunnels bidirectional-associated co-routed	Shows details of an associated co-routed bidirectional LSP.

Show output for an associated co-routed bidirectional LSP configuration

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

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels role head

Name: tunnel-te1 Destination: 49.49.49.2
  Signalled-Name: IMC0_t1
  Status:
    Admin:      up Oper:      up Path:  valid Signalling: connected

    path option 1, type dynamic (Basis for Setup, path weight 20 (reverse 20))
    path option 1, type dynamic (Basis for Standby, path weight 20 (reverse 20))
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps CT0
    Creation Time: Sun May 4 12:09:56 2014 (03:24:11 ago)
  Config Parameters:
    Bandwidth:      0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
    Metric Type: TE (default)
    Hop-limit: disabled
    Cost-limit: disabled
    AutoRoute: disabled LockDown: disabled Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Loadshare:      0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
    Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
    Association ID: 100, Source: 49.49.49.2
    Reverse Bandwidth: 0 kbps (CT0), Standby: 0 kbps (CT0)
    BFD Fast Detection: Enabled
    BFD Parameters: Min-interval 100 ms (default), Multiplier 3 (default)
    BFD Bringup Timeout: Interval 60 seconds (default)
    BFD Initial Dampening: 16000 ms (default)
    BFD Maximum Dampening: 600000 ms (default)
    BFD Secondary Dampening: 20000 ms (default)
    Periodic LSP Ping: Interval 120 seconds (default)
    Session Down Action: ACTION_REOPTIMIZE, Reopt Timeout: 300
    BFD Encap Mode: GAL
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
```

Path Protection

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

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

[Configure Path Protection for Associated Bidirectional LSPs](#), on page 162 provides procedural details.

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

Configure Path Protection for Associated Bidirectional LSPs

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	interface tunnel-te <i>tunnel-id</i> Example: RP/0/RSP0/CPU0:router# interface tunnel-te 1	Configures an MPLS-TE tunnel interface.
Step 3	ipv4 unnumbered <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0	Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.
Step 4	bfd {fast-detect encap-mode} Example: RP/0/RSP0/CPU0:IMC0(config-if)#bfd RP/0/RSP0/CPU0:IMC0(config-if-tunte-bfd)#fast-detect RP/0/RSP0/CPU0:IMC0(config-if-tunte-bfd)#encap-mode gal	Specify if you want BFD enabled for the LSP over a Generic Associated Channel (G-ACh) or over a IP channel. IP channel is the default.
Step 5	destination <i>ip-address</i> Example: RP/0/RP0/CPU0:router(config-if)# destination 49.49.49.2	Assigns a destination address on the new tunnel. The destination address is the remote node's MPLS-TE router ID.
Step 6	bidirectional Example: Router(config-if)# bidirectional	Configure the ingress router for the LSP and include the bidirectional statement to specify that the LSP be established as a bidirectional LSP.

	Command or Action	Purpose
Step 7	bidirectional association {id <0-65535> source-address <IP address>} [global-id <0-4294967295>} Example: Router(config-if-bidir)# association id 1 source-address 11.0.0.1	Set the association ID that uniquely identifies the association of LSPs, which is the tunnel ID of the bound LSP (master/slave mode) or the tunnel ID of the binding LSP (peer mode). Also, set the source address to the tunnel sender address of the bound LSP (master/slave mode) or the tunnel sender address of the binding LSP (peer mode). Also, set the ID for associating the global source. Note Association ID, association source and optional global-id must be configured identically on both the endpoints.
Step 8	association type co-routed Example: Router(config-if-bidir)#association type co-routed	Specify that the LSP be established as a associated co-routed bidirectional LSP.
Step 9	path-protection Example: RP/0/RSP0/CPU0:IMC0 (config-if-bidir-co-routed)#path-protection	Enable path protection.
Step 10	path-option preference - priority {dynamic explicit} Example: RP/0/RP0/CPU0:router (config-if) # path-option 1 dynamic	Sets the path option and assigns the path-option ID. Note Both sides of the co-routed bidirectional LSPs must use dynamic or matching co-routed strict-hop explicit path-option.
Step 11	commit	

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

```
RP/0/RSP0/CPU0:IMC0#config
RP/0/RSP0/CPU0:IMC0(config)#interface tunnel-te 1
RP/0/RSP0/CPU0:IMC0(config-if)#ipv4 unnumbered loopback0
RP/0/RSP0/CPU0:IMC0(config-if)#destination 49.49.49.2
RP/0/RSP0/CPU0:IMC0(config-if)#bidirectional
RP/0/RSP0/CPU0:IMC0(config-if-bidir)#association id 100 source-address 49.49.49.2
RP/0/RSP0/CPU0:IMC0(config-if-bidir)#association type co-routed
RP/0/RSP0/CPU0:IMC0(config-if-bidir-co-routed)#path-protection
RP/0/RSP0/CPU0:IMC0(config-if)#path-option 1 dynamic
RP/0/RSP0/CPU0:IMC0(config-if)#commit
```

OAM Support for Associated Bidirectional LSPs

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

- **Continuity check:** You can configure bidirectional forwarding detection (BFD) over a Generic Associated Channel (G-ACh) with hardware assist. This allows for BFD Hello packets to be generated and processed in hardware making smaller Hello intervals such as 3.3 ms feasible. For more information on BFD and BFD hardware offload see *Implementing BFD* module in the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.
- **Fault notification:** You can run Fault OAM over associated bidirectional co-routed LSPs to convey fault notification from mid-point to end-point of the LSP. The following fault OAM messages are supported:
 - Link Down Indication (LDI): generated when an interface goes down (for example, to fiber-cut) at mid-point.
 - Lock Report (LKR): generated when an interface is shutdown at mid-point.
You can configure fault OAM to generate OAM message at mid-point or enable protection switching due to fault OAM at end-point. [Generate Fault OAM Messages at Mid-point, on page 164](#) and [Generate Fault OAM Messages at End-point, on page 165](#) provides procedural details.
- **Fault diagnostics:** You can use the ping and traceroute features as a means to check connectivity and isolate failure points for both co-routed and non-co-routed bidirectional TE tunnels. *MPLS Network Management with MPLS LSP Ping and MPLS SP Traceroute* provides details.

Generate Fault OAM Messages at Mid-point

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

SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng`
3. `fault-oam`
4. `commit`

DETAILED STEPS

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

Generate Fault OAM Messages at End-point

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

SUMMARY STEPS

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

DETAILED STEPS

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

Pseudowire Call Admission Control

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

Configuration Examples for Cisco MPLS-TE

These configuration examples are used for MPLS-TE:

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

Build MPLS-TE Topology and Tunnels: Example

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

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

The following example shows how to configure tunnel interfaces:

```
interface tunnel-te1
  destination 192.168.92.125
  ipv4 unnumbered loopback 0
  path-option 1 dynamic
  bandwidth 100
  commit
show mpls traffic-eng tunnels
show ipv4 interface brief
show mpls traffic-eng link-management admission-control
!
interface tunnel-te1
  autoroute announce
  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

```

Related Topics

[Building MPLS-TE Topology, on page 50](#)

[Creating an MPLS-TE Tunnel, on page 53](#)

[How MPLS-TE Works, on page 4](#)

Configure IETF DS-TE Tunnels: Example

The following example shows how to configure DS-TE:

```

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

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

```

Related Topics

[Configuring a Prestandard DS-TE Tunnel, on page 65](#)

[Prestandard DS-TE Mode, on page 10](#)

Configure MPLS-TE and Fast-Reroute on OSPF: Example

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

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

Related Topics

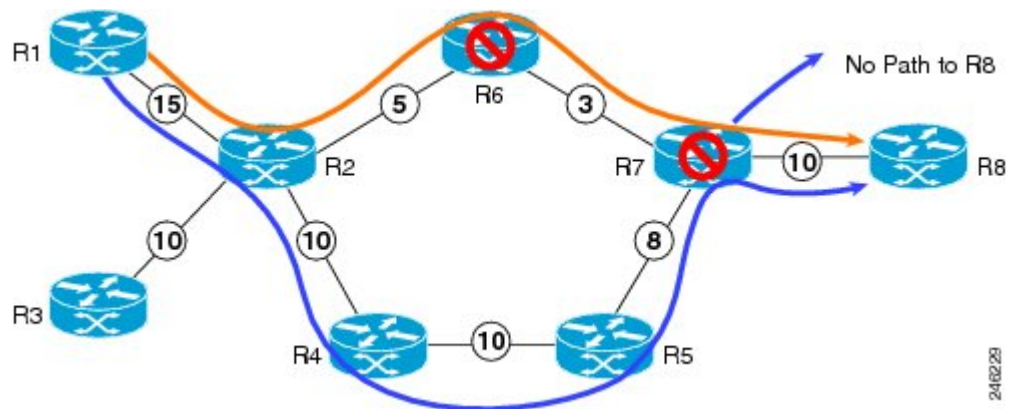
[Configuring MPLS -TE and Fast-Reroute on OSPF, on page 71](#)

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

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

This figure illustrates the IS-IS overload bit scenario:

Figure 11: IS-IS overload bit



Consider a MPLS TE topology in which usage of nodes that indicated an overload situation was restricted. In this topology, the router R7 exhibits overload situation and hence this node can not be used during TE CSPF. To overcome this limitation, the IS-IS overload bit avoidance (OLA) feature was introduced. This feature allows network administrators to prevent RSVP-TE label switched paths (LSPs) from being disabled when a router in that path has its Intermediate System-to-Intermediate System (IS-IS) overload bit set.

The IS-IS overload bit avoidance feature is activated at router R1 using this command:

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

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


Related Topics

[Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE, on page 73](#)

[Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE, on page 14](#)

Configure 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-gte 1
    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
    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-gte 1
ipv4 unnumbered Loopback 0
passive
match identifier head router_hostname_t1
destination 11.11.11.11
!
```

Configure Flexible Name-based Tunnel Constraints: Example

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

R2

```

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

```

```

!
interface GigabitEthernet0/1/0/2
 attribute-names green purple
!
interface GigabitEthernet0/1/0/3
 attribute-names green orange
!
affinity-map red 1
affinity-map blue 2
affinity-map black 80
affinity-map green 4
affinity-map white 40
affinity-map orange 20
affinity-map purple 10
affinity-map yellow 8
!

```

Related Topics

[Assigning Color Names to Numeric Values, on page 97](#)

[Associating Affinity-Names with TE Links, on page 98](#)

[Associating Affinity Constraints for TE Tunnels, on page 99](#)

[Flexible Name-based Tunnel Constraints, on page 17](#)

Configure 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 routed path is optional.

```

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

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

```

The following configuration example shows how to configure loose-path retry period (range is 30 to 600 seconds) on headend router.

```

config
 mpls traffic-eng
  timers loose-path retry-period 120

```

The following configuration example shows the global configuration for loose hop expansion affinity or metric on ABR.

```

config
 mpls traffic-eng path-selection loose-expansion affinity 0xff
 mpls traffic-eng path-selection loose-expansion metric te class-type 5

```

Configure Forwarding Adjacency: Example

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

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

Related Topics

[Configuring MPLS-TE Forwarding Adjacency, on page 103](#)

[MPLS-TE Forwarding Adjacency Benefits, on page 21](#)

Configure 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

```

Related Topics

[Setting Unequal Load Balancing Parameters, on page 104](#)

[Enabling Unequal Load Balancing, on page 105](#)

[Unequal Load Balancing, on page 22](#)

Configure PCE: Example

The following configuration example illustrates a PCE configuration:

```

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

```

The following configuration example illustrates PCC configuration:

```

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

```

Related Topics

- [Configuring a Path Computation Client, on page 106](#)
- [Configuring a Path Computation Element Address, on page 107](#)
- [Configuring PCE Parameters, on page 108](#)
- [Path Computation Element, on page 23](#)

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
```

Related Topics

- [Configuring Policy-based Tunnel Selection, on page 111](#)
- [Policy-Based Tunnel Selection Functions, on page 25](#)
- [Policy-Based Tunnel Selection, on page 24](#)

Configure Tunnels for Path Protection: Example

The path protection feature is configured on only the source router. The dynamic path option is a prerequisite to configure a path protection.

```
interface tunnel-te150
ipv4 unnumbered Loopback150
autoroute announce
destination 151.151.151.151
affinity 11 mask 11
path-protection
path-option 2 explicit name p2mp3-p2mp4-p2mp5_1
path-option 10 dynamic
```

Related Topics

- [Path Protection, on page 25](#)
- [Pre-requisites for Path Protection, on page 26](#)
- [Restrictions for Path Protection, on page 26](#)
- [Restrictions for Explicit Path Protection, on page 27](#)

Configure Tunnels for Explicit Path Protection: Example

The path protection feature is configured on only the source router. The **protected-by** keyword configures path protection for an explicit path that is protected by another explicit path.

```
interface tunnel-te150
  ipv4 unnumbered Loopback150
  autoroute announce
  destination 151.151.151.151
  affinity 11 mask 11
  path-protection
  path-option 2 explicit name p2mp3-p2mp4-p2mp5_1 protected-by 10
  path-option 10 explicit
```

Configure Tunnels for Co-existence of Path Protection with Fast Reroute: Example

The path protection feature is configured on only the source router. The dynamic path option is a prerequisite to configure a path protection.

```
interface tunnel-te1
  fast-reroute
  ipv4 unnumbered Loopback150
  autoroute announce
  destination 151.151.151.151
  affinity 11 mask 11
  path-protection
  path-option 2 explicit name p2mp3-p2mp4-p2mp5_1
  path-option 10 dynamic
```

Configure Automatic Bandwidth: Example

The following configuration example illustrates an automatic bandwidth configuration:

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

Related Topics

[Configuring the Collection Frequency, on page 112](#)

[Configuring the Automatic Bandwidth Functions, on page 114](#)

[MPLS-TE Automatic Bandwidth Overview, on page 28](#)

Configure the MPLS-TE Shared Risk Link Groups: Example

The following configuration example shows how to specify the SRLG value of each link that has a shared risk with another link:

```
config t
srlg
  interface POS0/4/0/0
    value 10
    value 11
  |
  interface POS0/4/0/1
    value 10
  |
```

The following example shows the SRLG values configured on a specific link.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng topology brief
My_System_id: 100.0.0.2 (OSPF 0 area 0)
My_System_id: 0000.0000.0002.00 (IS-IS 1 level-1)
My_System_id: 0000.0000.0002.00 (IS-IS 1 level-2)
My_BC_Model_Type: RDM

Signalling error holddown: 10 sec Global Link Generation 389225

IGP Id: 0000.0000.0002.00, MPLS TE Id: 100.0.0.2 Router Node (IS-IS 1 level-1)
IGP Id: 0000.0000.0002.00, MPLS TE Id: 100.0.0.2 Router Node (IS-IS 1 level-2)

Link[1]:Broadcast, DR:0000.0000.0002.07, Nbr Node Id:21, gen:389193
  Frag Id:0, Intf Address:51.2.3.2, Intf Id:0
  Nbr Intf Address:51.2.3.2, Nbr Intf Id:0
  TE Metric:10, IGP Metric:10, Attribute Flags:0x0
  Attribute Names:
    SRLGs: 1, 4, 5
  Switching Capability:, Encoding:
  BC Model ID:RDM
  Physical BW:1000000 (kbps), Max Reservable BW Global:10000 (kbps)
  Max Reservable BW Sub:10000 (kbps)
```

The following example shows the configured tunnels and associated SRLG values.

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

<snip>
Signalling Summary:
  LSP Tunnels Process: running
  RSVP Process: running
  Forwarding: enabled
  Periodic reoptimization: every 3600 seconds, next in 1363 seconds
  Periodic FRR Promotion: every 300 seconds, next in 181 seconds
  Auto-bw enabled tunnels: 0 (disabled)

Name: tunnel-te1 Destination: 100.0.0.3
Status:
  Admin: up Oper: up Path: valid Signalling: recovered

  path option 1, type explicit path123 (Basis for Setup, path weight 2)
    OSPF 0 area 0
  G-PID: 0x0800 (derived from egress interface properties)
  SRLGs excluded: 2,3,4,5
                  6,7,8,9
  Bandwidth Requested: 0 kbps CT0
<snip>
```

The following example shows all the interfaces associated with SRLG.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng topo srlg
My_System_id: 100.0.0.5 (OSPF 0 area 0)
My_System_id: 0000.0000.0005.00 (IS-IS 1 level-2)
My_System_id: 0000.0000.0005.00 (IS-IS ISIS-instance-123 level-2)
```

SRLG	Interface Addr	TE Router ID	IGP Area ID
10	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2
11	50.2.3.3	100.0.0.3	IS-IS 1 level-2
12	50.2.3.3	100.0.0.3	IS-IS 1 level-2
30	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2
77	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2
88	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2
1500	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2
10000000	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2
4294967290	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2
4294967295	50.4.5.5	100.0.0.5	IS-IS ISIS-instance-123 level-2

The following example shows the NHOP and NNHOP backup tunnels with excluded SRLG values.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng topology path dest 100.0.0.5 exclude-srlg ipaddr

Path Setup to 100.0.0.2:
bw 0 (CT0), min_bw 0, metric: 30
setup_pri 7, hold_pri 7
affinity_bits 0x0, affinity_mask 0xffff
Exclude SRLG Intf Addr : 50.4.5.5
SRLGs Excluded : 10, 30, 1500, 10000000, 4294967290, 4294967295
Hop0:50.5.1.5
Hop1:50.5.1.1
Hop2:50.1.3.1
Hop3:50.1.3.3
Hop4:50.2.3.3
Hop5:50.2.3.2
Hop6:100.0.0.2
```

The following example shows an extract of explicit-path set to protect a specific interface.

```
RP/0/RP0/CPU0:router#sh mpls traffic-eng topology path dest 10.0.0.5 explicit-path name
name

Path Setup to 100.0.0.5:
bw 0 (CT0), min_bw 9999, metric: 2
setup_pri 7, hold_pri 7
affinity_bits 0x0, affinity_mask 0xffff
SRLGs Excluded: 10, 30, 77, 88, 1500, 10000000
                4294967290, 4294967295

Hop0:50.3.4.3
Hop1:50.3.4.4
Hop2:50.4.5.4
Hop3:50.4.5.5
Hop4:100.0.0.5
```

Related Topics

[Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link](#), on page 117

[Creating an Explicit Path With Exclude SRLG](#), on page 119

[Using Explicit Path With Exclude SRLG](#), on page 120

[Creating a Link Protection on Backup Tunnel with SRLG Constraint](#), on page 122

[Creating a Node Protection on Backup Tunnel with SRLG Constraint](#), on page 125

[MPLS Traffic Engineering Shared Risk Link Groups, on page 35](#)
[Explicit Path, on page 36](#)
[Fast ReRoute with SRLG Constraints, on page 36](#)
[Importance of Protection, on page 38](#)
[Delivery of Packets During a Failure, on page 39](#)
[Multiple Backup Tunnels Protecting the Same Interface , on page 39](#)
[SRLG Limitations, on page 39](#)
[MPLS TE SRLG Scale Enhancements, on page 40](#)

Configure the MPLS-TE Auto-Tunnel Backup: Example

The following example shows the auto-tunnel backup configuration for core or edge routers.

```
RP/0/RP0/CPU0:router(config)#
mpls traffic-eng
  auto-tunnel backup
    tunnel-id min 60000 max 61000

  interface pos 0/1/0/0
    auto-tunnel backup
      attribute-set ab
```

The following example shows the protection (NNHOP and SRLG) that was set on the auto-tunnel backup.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 1
Signalling Summary:
    LSP Tunnels Process:  running
    RSVP Process:        running
    Forwarding:           enabled
    Periodic reoptimization: every 3600 seconds, next in 2524 seconds
    Periodic FRR Promotion: every 300 seconds, next in 49 seconds
    Auto-bw enabled tunnels: 1

Name: tunnel-te1  Destination: 200.0.0.3 (auto backup)
Status:
  Admin:    up  Oper:    up  Path:  valid  Signalling: connected

  path option 10,  type explicit (autob_nnhop_srlg_tunnel1) (Basis for Setup, path weight
11)
  path option 20,  type explicit (autob_nnhop_tunnel1)
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 0 kbps  CT0
  Creation Time: Fri Jul 10 01:53:25.581 PST  (1h 25m 17s ago)

Config Parameters:
  Bandwidth:          0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
  Metric Type: TE (default)
  AutoRoute: disabled LockDown: disabled  Policy class: not set
  Forwarding-Adjacency: disabled
  Loadshare:          0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Disabled, Protection Desired: None
  Path Protection: Not Enabled
Auto Backup:
  Protected LSPs: 4
  Protected S2L Sharing Families: 0
  Protected S2Ls: 0
  Protected i/f: Gi0/1/0/0  Protected node: 20.0.0.2
  Protection: NNHOP+SRLG
  Unused removal timeout: not running
History:
```

```

Tunnel has been up for: 00:00:08
Current LSP:
  Uptime: 00:00:08
Prior LSP:
  ID: path option 1 [545]
  Removal Trigger: configuration changed

Path info (OSPF 0 area 0):
Hop0: 10.0.0.2
Hop1: 100.0.0.2
Hop2: 100.0.0.3
Hop3: 200.0.0.3

```

The following example shows automatically created path options for this backup auto-tunnel.

```

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 1 detail
Signalling Summary:
  LSP Tunnels Process: running
  RSVP Process: running
  Forwarding: enabled
  Periodic reoptimization: every 3600 seconds, next in 2524 seconds
  Periodic FRR Promotion: every 300 seconds, next in 49 seconds
  Auto-bw enabled tunnels: 1

Name: tunnel-tel Destination: 200.0.0.3 (auto backup)
Status:
  Admin: up Oper: up Path: valid Signalling: connected

  path option 10, type explicit (autob_nnhop_srlg_tunnell) (Basis for Setup, path weight
11)
  path option 20, type explicit (autob_nnhop_tunnell)
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 0 kbps CT0
  Creation Time: Fri Jul 10 01:53:25.581 PST (1h 25m 17s ago)

Config Parameters:
  Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
  Metric Type: TE (default)
  AutoRoute: disabled LockDown: disabled Policy class: not set
  Forwarding-Adjacency: disabled
  Loadshare: 0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Disabled, Protection Desired: None
  Path Protection: Not Enabled
Auto Backup (NNHOP+SRLG):
  Protected LSPs: 4
  Protected S2L Sharing Families: 0
  Protected S2Ls: 0
  Protected i/f: Gi0/1/0/0 Protected node: 20.0.0.2
  Protection: NNHOP+SRLG
  Unused removal timeout: not running

Path Options Details:
  10: Explicit Path Name: (autob_nnhop_srlg_tel)
    1: exclude-srlg 50.0.0.1
    2: exclude-address 50.0.0.2
    3: exclude-node 20.0.0.2
  20: Explicit Path Name: (autob_nnhop_tel)
    1: exclude-address 50.0.0.1
    2: exclude-address 50.0.0.2
    3: exclude-node 20.0.0.2

History:
  Tunnel has been up for: 00:00:08
  Current LSP:
    Uptime: 00:00:08
  Prior LSP:
    ID: path option 1 [545]
    Removal Trigger: configuration changed

Path info (OSPF 0 area 0):
Hop0: 10.0.0.2

```

```
Hop1: 100.0.0.2
Hop2: 100.0.0.3
Hop3: 200.0.0.3
```

This example shows the automatically created backup tunnels.

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

TUNNEL NAME	DESTINATION	STATUS	STATE
tunnel-te0	200.0.0.3	up	up
tunnel-te1	200.0.0.3	up	up
tunnel-te2	200.0.0.3	up	up
tunnel-te50	200.0.0.3	up	up
*tunnel-te60	200.0.0.3	up	up
*tunnel-te70	200.0.0.3	up	up
*tunnel-te80	200.0.0.3	up	up

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

Tunnel Name	LSP ID	Destination Address	Source Address	State	FRR State	LSP Role	Path Prot
tunnel-te0	549	200.0.0.3	200.0.0.1	up	Inact	Head	InAct
tunnel-te1	546	200.0.0.3	200.0.0.1	up	Inact	Head	InAct
tunnel-te2	6	200.0.0.3	200.0.0.1	up	Inact	Head	InAct
tunnel-te50	6	200.0.0.3	200.0.0.1	up	Active	Head	InAct
tunnel-te60	4	200.0.0.3	200.0.0.1	up	Active	Head	InAct
tunnel-te70	4	200.0.0.3	200.0.0.1	up	Active	Head	InAct
tunnel-te80	3	200.0.0.3	200.0.0.1	up	Active	Head	InAct

This example shows the auto-tunnel backup details.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-tunnel backup detail
```

```
Name: tunnel-te400 Destination: 1.1.1.1 (auto-tunnel backup)
```

```
Status:
```

```
Admin: up Oper: up Path: valid Signalling: connected
```

```
path option 20, type explicit (autob_nnhop_te400) (Basis for Setup, path weight 2)
```

```
path option 10, type explicit (autob_nnhop_srlg_te400) [disabled]
```

```
G-PID: 0x0800 (derived from egress interface properties)
```

```
Bandwidth Requested: 0 kbps CT0
```

```
Creation Time: Thu Aug 16 18:30:41 2012 (00:01:28 ago)
```

```
Config Parameters:
```

```
Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
```

```
Metric Type: TE (default)
```

```
Metric Type: TE (default)
```

```
Hop-limit: disabled
```

```
AutoRoute: disabled LockDown: disabled Policy class: not set
```

```
Forwarding-Adjacency: disabled
```

```
Loadshare: 0 equal loadshares
```

```
Auto-bw: disabled
```

```
Fast Reroute: Disabled, Protection Desired: None
```

```
Path Protection: Not Enabled
```

```
Soft Preemption: Disabled
```

```
Auto Backup:
```

```
Protected LSPs: 1
```

```
Protected S2L Sharing Families: 0
```

```
Protected S2L: 0
```

```
Protected i/f: Gi0/1/0/3 Protected node: 3.3.3.3
```

```
Attribute-set: abl
```

```
Protection: NNHOP
```

```
Unused removal timeout: not running
```

```
Path Option Details:
```

```
10: Explicit Path Name: (autob_nnhop_srlg_te400)
```

```
1: exclude-srlg 34.9.0.4
```

```
2: exclude-address 34.9.0.3
```

```
3: exclude-node 3.3.3.3
```

```
20: Explicit Path Name: (autob_nnhop_te400)
```

```
1: exclude-address 34.9.0.4
```

```
2: exclude-address 34.9.0.3
```

```
3: exclude-node 3.3.3.3
```

```

SNMP Index: 221
History:
  Tunnel has been up for: 00:00:34 (since Thu Aug 16 18:31:35 EST 2012)
  Current LSP:
    Uptime: 00:00:34 (since Thu Aug 16 18:31:35 EST 2012)
  Current LSP Info:
    Instance: 2, Signaling Area: OSPF 100 area 1.2.3.4
    Uptime: 00:00:34 (since Thu Aug 16 18:31:35 EST 2012)
    Outgoing Interface: GigabitEthernet0/1/0/2, Outgoing Label: 16000
    Router-IDs: local      4.4.4.4
                downstream 2.2.2.2
    Soft Preemption: None
  Path Info:
    Outgoing:
      Explicit Route:
        Strict, 24.9.0.2
        Strict, 12.9.1.1
        Strict, 1.1.1.1

    Record Route: Empty
    Tspec: avg rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
    Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                        Soft Preemption Desired: Not Set

  Resv Info:
    Record Route:
      IPv4 24.9.0.2, flags 0x0
      IPv4 12.9.1.1, flags 0x0
      Fspec: avg rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
    Displayed 1 (of 104) heads, 0 (of 0) midpoints, 0 (of 201) tails
    Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

```

This example shows the automatically created backup tunnels.

```

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-tunnel backup tabular

```

Tunnel Name	LSP ID	Destination Address	Source Address	Tun State	FRR State	LSP Role	Path Prot
*tunnel-te400	2	1.1.1.1	4.4.4.4	up	Inact	Head	Inact
*tunnel-te401	2	3.3.3.3	4.4.4.4	up	Inact	Head	Inact

```

* = automatically created backup tunnel

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-tunnel backup brief

```

TUNNEL NAME	DESTINATION	STATUS	STATE
*tunnel-te400	1.1.1.1	up	up
*tunnel-te401	3.3.3.3	up	up

```

* = automatically created backup tunnel
Displayed 2 (of 104) heads, 0 (of 0) midpoints, 0 (of 201) tails
Displayed 2 up, 0 down, 0 recovering, 0 recovered heads

```

This example shows the attribute-set for auto-backup tunnels.

```

RP/0/RP0/CPU0:router# show mpls traffic-eng attribute-set auto-backup

```

```

Attribute Set Name: ab (Type: auto-backup)
Number of affinity constraints: 2
  Include bit map      : 0x4
  Include name         : blue
  Exclude bit map      : 0x2
  Exclude name         : red
Priority: 7 7 (Default)
Record-route: Enabled
Policy-class: 1
Logging: reoptimize, state
List of protected interfaces (count 1)
  POS0_3_0_1
List of tunnel IDs (count 1)
  3000

```

This example shows the attribute-set for auto-mesh tunnels.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng attribute-set auto-mesh
```

```
Attribute Set Name: am (Type: auto-mesh)
  Bandwidth: 100 kbps (CT0)
  Number of affinity constraints: 2
    Include bit map      : 0x8
    Include name         : yellow
    Exclude bit map      : 0x2
    Exclude name         : red
  Priority: 2 2
  Interface Bandwidth: 0 kbps (Default)
  AutoRoute Announce: Disabled
  Auto-bw: Disabled
  Soft Preemption: Disabled
  Fast Reroute: Enabled, Protection Desired: Node, Bandwidth
  Record-route: Enabled
  Policy-class: 0 (Not configured)
  Logging: None
  List of Mesh Groups (count 1)
  1
```

This example shows the details about the tunnel that is using auto-backup type of attribute-set.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels attribute-set auto-backup ab
```

```
Name: tunnel-te3000 Destination: 1.1.1.1 (auto-tunnel backup)
Status:
  Admin:    up Oper:    up Path:  valid Signalling: connected

  path option 20, type explicit (autob_nhop_te3000) (Basis for Setup, path weight 2)
  path option 10, type explicit (autob_nhop_srlg_te3000) [disabled]
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 0 kbps CT0
  Creation Time: Tue Aug 14 23:24:27 2012 (00:05:28 ago)
Config Parameters:
  Bandwidth:      0 kbps (CT0) Priority:  7  7
  Number of affinity constraints: 2
    Include bit map      : 0x4
    Include name         : blue
    Exclude bit map      : 0x2
    Exclude name         : red

  Metric Type: TE (default)
  Hop-limit: disabled
  AutoRoute: disabled LockDown: disabled Policy class: 1
  Forwarding-Adjacency: disabled
  Loadshare:      0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Disabled, Protection Desired: None
  Path Protection: Not Enabled
  Soft Preemption: Disabled
Auto Backup:
  Protected LSPs: 2
  Protected S2L Sharing Families: 0
  Protected S2L: 0
  Protected i/f: P00/3/0/1
  Attribute-set: ab
  Protection: NHOP
  Unused removal timeout: not running
History:
  Tunnel has been up for: 00:04:57 (since Tue Aug 14 23:24:58 EST 2012)
  Current LSP:
    Uptime: 00:04:57 (since Tue Aug 14 23:24:58 EST 2012)

  Path info (OSPF 100 area 16909060):
  Node hop count: 2
  Hop0: 23.9.0.2
  Hop1: 12.9.0.2
  Hop2: 12.9.0.1
  Hop3: 1.1.1.1
  Displayed 1 (of 7) heads, 0 (of 3) midpoints, 0 (of 0) tails Displayed 1 up, 0 down, 0
```

recovering, 0 recovered heads

This example shows the protected interface for auto-backup auto-tunnels.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels backup protected-interface
```

```
Interface: Gi0/2/0/1 (auto-tunnel backup)
  SRLG: N/A, NHOP-only: No
  Attribute-set: Not configured
  Auto-tunnel backup recreate time remaining: timer not running
  No backup tunnel found

Interface: Gi0/2/0/3
  tunnel-te340 PROTECTED : out i/f: PO0/3/0/2 Admin: up Oper: up

Interface: PO0/3/0/1 (auto-tunnel backup)
  SRLG: N/A, NHOP-only: No
  Attribute-set: ab
  Auto-tunnel backup recreate time remaining: timer not running
  *tunnel-te3000 NHOP : out i/f: Gi0/2/0/2 Admin: up Oper: up
```

* = automatically created backup tunnel

This example shows the details about all the tunnels that are using auto-mesh type of attribute-set.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels attribute-set auto-mesh all
```

```
Name: tunnel-te3501 Destination: 1.1.1.1 (auto-tunnel mesh)
Status:
  Admin: up Oper: up Path: valid Signalling: connected

  path option 10, type dynamic (Basis for Setup, path weight 2)
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 100 kbps CT0
  Creation Time: Tue Aug 14 23:25:41 2012 (00:06:13 ago)
Config Parameters:
  Bandwidth: 100 kbps (CT0) Priority: 2 2
  Number of affinity constraints: 2
    Include bit map : 0x8
    Include name : yellow
    Exclude bit map : 0x2
    Exclude name : red

  Metric Type: TE (default)
  Hop-limit: disabled
  AutoRoute: disabled LockDown: disabled Policy class: not set
  Forwarding-Adjacency: disabled
  Loadshare: 0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Enabled, Protection Desired: Node, Bandwidth
  Path Protection: Not Enabled
  Attribute-set: am (type auto-mesh)
  Soft Preemption: Disabled
Auto-tunnel Mesh:
  Group ID: 1
  Destination list: blah
  Unused removal timeout: not running
History:
  Tunnel has been up for: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)
  Current LSP:
    Uptime: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)

  Path info (OSPF 100 area 16909060):
  Node hop count: 2
  Hop0: 23.9.0.2
  Hop1: 12.9.0.2
  Hop2: 12.9.0.1
  Hop3: 1.1.1.1

Name: tunnel-te3502 Destination: 2.2.2.2 (auto-tunnel mesh)
Status:
  Admin: up Oper: up Path: valid Signalling: connected

  path option 10, type dynamic (Basis for Setup, path weight 1)
```



```

G-PID: 0x0800 (derived from egress interface properties)
Bandwidth Requested: 100 kbps CT0
Creation Time: Tue Aug 14 23:25:41 2012 (00:06:13 ago)
Config Parameters:
  Bandwidth: 100 kbps (CT0) Priority: 2 2
  Number of affinity constraints: 2
    Include bit map : 0x8
    Include name : yellow
    Exclude bit map : 0x2
    Exclude name : red

  Metric Type: TE (default)
  Hop-limit: disabled
  AutoRoute: disabled LockDown: disabled Policy class: not set
  Forwarding-Adjacency: disabled
  Loadshare: 0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Enabled, Protection Desired: Node, Bandwidth
  Path Protection: Not Enabled
  Attribute-set: am (type auto-mesh)
  Soft Preemption: Disabled
Auto-tunnel Mesh:
  Group ID: 1
  Destination list: blah
  Unused removal timeout: not running
History:
  Tunnel has been up for: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)
  Current LSP:
    Uptime: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)

Path info (OSPF 100 area 16909060):
Node hop count: 1
Hop0: 23.9.0.2
Hop1: 2.2.2.2

Name: tunnel-te3503 Destination: 4.4.4.4 (auto-tunnel mesh)
Status:
  Admin: up Oper: down Path: not valid Signalling: Down

  path option 10, type dynamic
  Last PCALC Error: Tue Aug 14 23:31:26 2012
  Info: No path to destination, 4.4.4.4 (affinity)
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 100 kbps CT0
  Creation Time: Tue Aug 14 23:25:41 2012 (00:06:13 ago)
Config Parameters:
  Bandwidth: 100 kbps (CT0) Priority: 2 2
  Number of affinity constraints: 2
    Include bit map : 0x8
    Include name : yellow
    Exclude bit map : 0x2
    Exclude name : red

  Metric Type: TE (default)
  Hop-limit: disabled
  AutoRoute: disabled LockDown: disabled Policy class: not set
  Forwarding-Adjacency: disabled
  Loadshare: 0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Enabled, Protection Desired: Node, Bandwidth
  Path Protection: Not Enabled
  Attribute-set: am (type auto-mesh)
  Soft Preemption: Disabled
Auto-tunnel Mesh:
  Group ID: 1
  Destination list: blah
  Unused removal timeout: not running
Displayed 3 (of 7) heads, 0 (of 3) midpoints, 0 (of 0) tails Displayed 2 up, 1 down, 0
recovering, 0 recovered heads

```

Related Topics

[Enabling an AutoTunnel Backup, on page 60](#)

[Removing an AutoTunnel Backup, on page 61](#)

[Establishing MPLS Backup AutoTunnels to Protect Fast Reroutable TE LSPs, on page 62](#)

[Establishing Next-Hop Tunnels with Link Protection, on page 63](#)

[Backup AutoTunnels, on page 5](#)

Configure Point-to-Multipoint TE: Examples

These configuration examples show how to configure Point-to-Multipoint TE:

Configure Point-to-Multipoint for the Source: Example

At the source, multicast routing must be enabled on both the tunnel-mte interface and customer-facing interface. Then, the static-group must be configured on the tunnel-mte interface to forward specified multicast traffic over P2MP LSP.

**Note**

The multicast group address, which is in Source-Specific Multicast (SSM) address range (ff35::/16), must be used on the static-group configuration because Cisco IOS XR software supports only SSM for Label Switch Multicast (LSM). Additionally, the customer-facing interface must have an IPv6 address.

```
multicast-routing
 address-family ipv6
   interface tunnel-mte 1
     enable
   !
   interface GigabitEthernet0/2/0/3
     enable
   !
   !
   !
router mld
 vrf default
   interface tunnel-mte 1
     static-group ff35::1 2000::1 3eFF::A
   !
   !
   !
 interface tunnel-mte 1
   ipv4 unnumbered Loopback0
   destination 3.3.3.3
     path-option 1 dynamic
   destination 4.4.4.4
     path-option 1 dynamic
   !
   !
```

Related Topics

[Enabling Multicast Routing on the Router, on page 128](#)

[Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)

[Configuring the Static Group for the Point-to-Multipoint Interface, on page 130](#)

[Point-to-Multipoint RSVP-TE , on page 32](#)

Configure the Point-to-Multipoint Tunnel: Example

There is no difference between logging events at the tunnel level for both P2P and P2MP. The P2MP tunnel reoptimizes only at the per tunnel level.

```
interface tunnel-mte1
  ipv4 unnumbered Loopback0
  destination 60.60.60.60
  logging events lsp-status state
  logging events lsp-status reroute
  path-option 10 explicit name toR6_via_R2andR3
  !
  logging events lsp-status reoptimize
  logging events lsp-status state
  logging events lsp-status reroute
  fast-reroute
  record-route
  !
explicit-path name PATH7
  index 1 next-address strict ipv4 unicast 192.168.7.2
  index 2 next-address strict ipv4 unicast 192.168.7.1
  index 3 next-address strict ipv4 unicast 192.168.16.1
  index 4 next-address strict ipv4 unicast 192.168.16.2
  !
```

Related Topics

[Configuring Destinations for the Tunnel Interface, on page 131](#)

[Path Option for Point-to-Multipoint RSVP-TE, on page 34](#)

[Logging Per Destinations for Point-to-Multipoint , on page 136](#)

[Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)

Disable a Destination: Example

From the tunnel-mte interface, you can disable the destination.

```
interface tunnel-mte101
  ipv4 unnumbered Loopback0
  destination 150.150.150.150
  disable
  path-option 10 dynamic
  !
  destination 150.150.150.150
  path-option 2 dynamic
  !
  !
```

Related Topics

[Disabling Destinations, on page 134](#)

[Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)

Configure the Point-to-Multipoint Solution: Example

Requirements for MPLS-TE Configuration

Before the Point-to-Multipoint (P2MP) tunnel is defined, these MPLS-TE requirements must be configured:

- Multiprotocol Label Switching traffic engineering (MPLS-TE)
- Resource ReSerVation Protocol (RSVP)
- Open Shortest Path First (OSPF)

This example shows the entire P2MP solution:

- Source is the location where the P2MP-TE tunnel interface is created.
- Tunnel contains multiple destinations. For example, the P2MP-TE tunnel is configured with two leaf node destinations by using the dynamic and explicit path options.
- Fast-Reroute (FRR) is specified on the P2MP tunnel.
- All regular TE tunnel options such as affinity or bandwidth are configured.
- Static mapping of the group address to the P2MP tunnel is done in IGMP.
Internet Group Management Protocol (IGMP).
- The P2MP-TE midpoint configuration requires only TE and Interior Gateway Protocol (IGP) information.
- The P2MP-TE receiver configuration requires a static group and RPF map.

```
!
explicit-path name g2-r2-r1
 index 1 next-address strict ipv4 unicast 10.2.15.1
!
explicit-path name g2-r2-r3
 index 1 next-address strict ipv4 unicast 10.2.25.1
 index 2 next-address strict ipv4 unicast 10.2.23.2
!
explicit-path name g2-r2-r4
 index 1 next-address strict ipv4 unicast 10.2.25.1
 index 2 next-address strict ipv4 unicast 10.2.24.2
!
ipv4 access-list ssm
 10 permit ipv4 232.1.0.0/16 any
 20 permit ipv4 232.3.0.0/16 any
 30 permit ipv4 232.4.0.0/16 any
!
ipv4 access-list ssm-test
 10 permit ipv4 235.0.0.0/8 any
!
interface Loopback0
 ipv4 address 192.168.1.2 255.255.255.255
!
interface tunnel-mte221
 ipv4 unnumbered Loopback0
 destination 192.168.1.1
  path-option 1 dynamic
!
 destination 192.168.1.3
  path-option 1 dynamic
!
 destination 192.168.1.4
  path-option 1 dynamic
!
```

```

!
interface tunnel-mte222
  ipv4 unnumbered Loopback0
  destination 192.168.1.1
  path-option 1 explicit name g2-r2-r1
!
  destination 192.168.1.3
  path-option 1 explicit name g2-r2-r3
!
  destination 192.168.1.4
  path-option 1 explicit name g2-r2-r4
!
  signalled-bandwidth 1000
!
interface MgmtEth0/RP0/CPU0/0
  ipv4 address 172.20.163.12 255.255.255.128
!
interface MgmtEth0/RP1/CPU0/0
  shutdown
!
interface GigabitEthernet0/0/0/0
  ipv4 address 172.2.1.2 255.255.255.0
  load-interval 30
!
interface GigabitEthernet0/0/0/1
  ipv4 address 10.1.15.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1.2
  ipv4 address 10.2.15.2 255.255.255.0
  encapsulation dot1q 2
!
interface GigabitEthernet0/0/0/2
  ipv4 address 10.1.25.2 255.255.255.0
!
interface GigabitEthernet0/0/0/2.2
  ipv4 address 10.2.25.2 255.255.255.0
  encapsulation dot1q 2
!
interface GigabitEthernet0/0/0/3
  shutdown
!
interface GigabitEthernet0/0/0/4
  shutdown
!
interface GigabitEthernet0/0/0/5
  shutdown
!
interface GigabitEthernet0/0/0/6
  shutdown
!
interface GigabitEthernet0/0/0/7
  shutdown
!
router static
  address-family ipv4 unicast
    0.0.0.0/0 1.56.0.1
    0.0.0.0/0 172.20.163.1
  !
!
router ospf 100
  nsr
  router-id 192.168.70.1
  area 0
    mpls traffic-eng
    interface Loopback0
    !
    interface GigabitEthernet0/0/0/0
    !
    interface GigabitEthernet0/0/0/1
    !
    interface GigabitEthernet0/0/0/1.2
    !
    interface GigabitEthernet0/0/0/2

```

```

!
interface GigabitEthernet0/0/0/2.2
!
!
mpls traffic-eng router-id 192.168.70.1
!
mpls oam
!
rsvp
interface GigabitEthernet0/0/0/0
bandwidth 20000
!
interface GigabitEthernet0/0/0/1
bandwidth 20000
!
interface GigabitEthernet0/0/0/2
bandwidth 20000
!
interface GigabitEthernet0/0/0/1.2
bandwidth 20000
!
interface GigabitEthernet0/0/0/2.2
bandwidth 20000
!
!
mpls traffic-eng
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/1.2
!
interface GigabitEthernet0/0/0/2.2
!
!
mpls ldp
router-id 192.168.1.2
nsr
graceful-restart
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/1.2
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/2.2
!
!
multicast-routing
address-family ipv4
core-tree-protocol rsvp-te
ssm range ssm
static-rpf 172.1.1.1 32 mpls 192.168.1.1
static-rpf 172.3.1.1 32 mpls 192.168.1.3
static-rpf 172.4.1.1 32 mpls 192.168.1.4
interface all enable
!
!
router igmp
!
interface tunnel-mte221
static-group 232.2.2.1 172.2.1.1
!
interface tunnel-mte222
static-group 232.2.2.2 172.2.1.1
!
interface GigabitEthernet0/0/0/0
static-group 232.1.2.1 172.1.1.1
static-group 232.1.2.2 172.1.1.1

```

```

static-group 232.3.2.1 172.3.1.1
static-group 232.3.2.2 172.3.1.1
static-group 232.4.2.1 172.4.1.1
static-group 232.4.2.2 172.4.1.1
!
!
end

```

Related Topics

[Enabling Multicast Routing on the Router, on page 128](#)

[Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)

[Configuring the Static Group for the Point-to-Multipoint Interface, on page 130](#)

[Point-to-Multipoint RSVP-TE , on page 32](#)

[Configuring Destinations for the Tunnel Interface, on page 131](#)

[Path Option for Point-to-Multipoint RSVP-TE, on page 34](#)

[Logging Per Destinations for Point-to-Multipoint , on page 136](#)

[Point-to-Multipoint Traffic-Engineering Overview, on page 30](#)

Configure MPLS TE Path-selection Cost Limit: Example

This example shows how to set the path-selection cost limit for MPLS TE tunnels at global, TE tunnel interface, and path-option attribute-set levels. By default, the cost-limit set at path-option attribute set takes the priority, if all options are configured and per tunnel interface level takes priority over global cost-limit. At per tunnel interface level, the global cost-limit takes the priority.

```

interface tunnel-tel
 path-selection cost-limit 2
!
mpls traffic-eng
 attribute-set path-option PO3AttrSet
   path-selection cost-limit 3
!
 path-selection cost-limit 1
!
!

```

Additional References

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

Related Documents

Related Topic	Document Title
MPLS-TE commands	<i>MPLS Traffic Engineering Commands</i> module in <i>Cisco IOS XR MPLS Command Reference for the Cisco CRS Router</i> .

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

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

