Implementing MPLS Traffic Engineering

Traditional IP routing emphasizes on forwarding traffic to the destination as fast as possible. As a result, the routing protocols find out the least-cost route according to its metric to each destination in the network and every router forwards the packet based on the destination IP address and packets are forwarded hop-by-hop. Thus, traditional IP routing does not consider the available bandwidth of the link. This can cause some links to be over-utilized compared to others and bandwidth is not efficiently utilized. Traffic Engineering (TE) is used when the problems result from inefficient mapping of traffic streams onto the network resources. Traffic engineering allows you to control the path that data packets follow and moves traffic flows from congested links to non-congested links that would not be possible by the automatically computed destination-based shortest path.

Multiprotocol Label Switching (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) relies on the MPLS backbone to replicate and expand upon the TE capabilities of Layer 2 ATM and Frame Relay networks.

MPLS-TE learns the topology and resources available in a network and then maps traffic flows to particular paths based on resource requirements and network resources such as bandwidth. MPLS-TE builds a unidirectional tunnel from a source to a destination in the form of a label switched path (LSP), which is then used to forward traffic. The point where the tunnel begins is called the tunnel headend or tunnel source, and the node where the tunnel ends is called the tunnel tailend or tunnel destination. A router through which the tunnel passes is called the mid-point of the tunnel.

MPLS uses extensions to a link-state based Interior Gateway Protocol (IGP), such as Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF). MPLS calculates TE tunnels at the LSP head based on required and available resources (constraint-based routing). If configured, the IGP automatically routes the traffic onto these LSPs. Typically, a packet that crosses the MPLS-TE backbone travels on a single LSP that connects the ingress point to the egress point. MPLS TE automatically establishes and maintains the LSPs across the MPLS network by using the Resource Reservation Protocol (RSVP).

Note Combination of unlabelled paths protected by labelled paths is not supported.

Prerequisites for Implementing Cisco MPLS Traffic Engineering

These prerequisites are required to implement MPLS TE:

- Router that runs Cisco IOS XR software .
- Installed composite mini-image and the MPLS package, or a full composite image.
- IGP activated.

Overview of MPLS-TE Features

In MPLS traffic engineering, IGP extensions flood the TE information across the network. Once the IGP distributes the link attributes and bandwidth information, the headend router calculates the best path from head to tail for the MPLS-TE tunnel. This path can also be configured explicitly. Once the path is calculated, RSVP-TE is used to set up the TE LSP (Labeled Switch Path).

To forward the traffic, you can configure autoroute, forward adjacency, or static routing. The autoroute feature announces the routes assigned by the tailend router and its downstream routes to the routing table of the headend router and the tunnel is considered as a directly connected link to the tunnel.

If forward adjacency is enabled, MPLS-TE tunnel is advertised as a link in an IGP network with the link's cost associated with it. Routers outside of the TE domain can see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network.

MPLS-TE provides protection mechanism known as fast reroute to minimize packet loss during a failure. For fast reroute, you need to create back up tunnels. The autotunnel backup feature enables a router to dynamically build backup tunnels when they are needed instead of pre-configuring each backup tunnel and then assign the backup tunnel to the protected interfaces.

DiffServ Aware Traffic Engineering (DS-TE) enables you to configure multiple bandwidth constraints on an MPLS-enabled interface to support various classes of service (CoS). These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

The MPLS traffic engineering auto-tunnel mesh feature allows you to set up full mesh of TE tunnels automatically with a minimal set of MPLS traffic engineering configurations. The MPLS-TE auto bandwidth feature allows you to automatically adjusts bandwidth based on traffic patterns without traffic disruption.

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

For detailed information about MPLS-TE features, see the *MPLS-TE Features - Details* topic.

Note MPLS-TE Nonstop Routing (NSR) is enabled by default without any user configuration and cannot be disabled.

MPLS-TE NSR means the application is in hot-standby mode and standby MPLS-TE instance is ready to take over from the active instance quickly on RP failover.

Note that the MPLS-TE does not do routing. If there is standby card available then the MPLS-TE instance is in a hot-standby position.

The following output shows the status of MPLS-TE NSR:

```
Router#show mpls traffic-eng nsr status

TE Process Role        : V1 Active
Current Status      : Ready
    Ready since        : Tue Nov 01 10:42:34 UTC 2022 (1w3d ago)
   IDT started         : Tue Nov 01 03:28:48 UTC 2022 (1w3d ago)
    IDT ended          : Tue Nov 01 03:28:48 UTC 2022 (1w3d ago)
Previous Status        : Not ready
    Not ready reason   : Collaborator disconnected
    Not ready since    : Tue Nov 01 10:42:34 UTC 2022 (1w3d ago)
```

During any issues with the MPLS-TE, the NSR on the router gets affected which is displayed in the show redundancy output as follows:

```
Router#show mpls traffic-eng nsr status details
.
.
.

Current active rmf state: 4 (I_READY)
All standby not-ready bits clear - standby should be ready

Current active rmf state for NSR: Not ready
<jid> <node> <name> Reason for standby not NSR-ready
1082 0/RP0/CPU0 te_control TE NSR session not synchronized
Not ready set Wed Nov 19 17:28:14 2022: 5 hours, 23 minutes ago
1082 0/RP1/CPU0 te_control Standby not connected
Not ready set Wed Nov 19 17:29:11 2022: 5 hours, 22 minutes ago
```

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 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.

The following sections describe the components of MPLS-TE:

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 keep track on topology and resource information to be flooded.

Link-state IGP

Either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF) can be used as IGPs. These IGPs are used to globally flood topology and resource information from the link management module.

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.

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.

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.

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 signaled, they are verified dynamically in the next path verification cycle.

RSVP-TE Dark Bandwidth Accounting

This section describes the RSVP-TE Dark Bandwidth Accounting feature that allows for the co-existence of non-zero bandwidth RSVP-TE tunnels and Segment Routing (SR) in the same network domain. This feature measures dark bandwidth traffic and accounts for it in the RSVP-TE bandwidth reservations to avoid overbooking the links in the network.

Dark bandwidth is the actual utilization of the link by the subset of the traffic that is not explicitly admission controlled by RSVP-TE. Dark bandwidth is not considered during path computation and admission control for distributed RSVP-TE LSPs.

In this solution, SR is assumed to be the main source of dark bandwidth on the links in the network. In addition, SR traffic is considered to have a higher priority than any other traffic transported by RSVP-TE LSPs. Therefore, the bandwidth consumed by SR effectively reduces the link bandwidth available to RSVP-TE LSPs.

The RSVP-TE Dark Bandwidth Accounting feature consists of the following:

- The measurement of SR traffic on interfaces via new per-interface aggregate SR counters
- The calculation of dark bandwidth rate based on the measured SR traffic statistics
- The calculation of the RSVP-TE effective maximum reservable bandwidth (BMRe).

  The BMRe is used for the purpose of pre-emption as well as advertisement (flooding) via IGP. A threshold is evaluated before triggering flooding.

Computing the Dark Bandwidth and RSVP-TE Effective Maximum Reservable Bandwidth

The statistics collector process (statsD) is responsible for returning statistics counters for each feature. For each traffic engineering (TE)-enabled interface, the TE process collects new SR bandwidth rate statistics (samples) from the statsD process, within a specified sampling interval. These samples are collected over a period of time called an application interval.

After each application interval, the average value of the collected rate samples is used to compute the dark bandwidth rate and the BMRe rate.

The following example shows how the BMRe is computed (assuming a link capacity of 10Gbps and a configured BMR [BMRc] of 90%):

- Link capacity = 10Gbps
- BMRc = RSVP percentage of link capacity = 9Gbps
- Calculated dark bandwidth rate = 2Gpbs
- BMRe = 7Gbps

In this example, the bandwidth available for RSVP-TE LSP admission is 7Gbps. This value is flooded in the network if the flooding threshold is crossed.

Note When you change the RSVP bandwidth percentage configuration or when the bundle capacity changes due to bundle-member state change, TE accounts for the dark bandwidth when new bandwidth values are advertised.

Note The measured dark bandwidth can be increased or decreased based on a configurable adjustment factor.

When the dark bandwidth rate increases for a link, it will lower the BMR of that link, which might trigger preemption of the RSVP-TE LSPs. 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.

Point-to-Multipoint Traffic-Engineering

This section contains the following topics:

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 network.

Note

- For P2MP tunnels, a Cisco 8000 Series router supports the mid-point router function, and does not support source or receiver functions. To know how to configure a source or receiver (destination) router in a P2MP tunnel, refer the MPLS configuration guide for the corresponding platform.
- The FRR function is not supported for P2MP tunnels.

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. Cisco 8000 Series routers only support the role of a mid-point.

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.
- The **interface tunnel-mte** command identifies the P2MP interface type.
- P2MP tunnel setup is supported with label replication.

- 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.

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.

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.

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.

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.

Point-to-Multipoint Implicit Null

The Point-to-Multipoint (P2MP) implicit null feature enables the forwarding of unicast traffic over P2MP tunnels. This feature is enabled by default and requires no configuration.

In a P2MP tunnel, the tailend router signals the implicit null label to the midpoint router. If the given MPI leg of the P2MP tunnel is implicit null capable (where the penultimate router is capable to do penultimate hop popping), the FIB (Forwarding Information Base) creates two NRLDI (Non Recursive Load Distribution Index) entries, one for forwarding the IPv6 labeled packets, and the other for non-labeled IPv4 unicast traffic.

The headend and the tailend routers handle the unicast traffic arriving on the P2MP tunnel. The midpoint router forwards the unicast traffic to its bud and tailend routers.

The use of implicit null at the end of a tunnel is called penultimate hop popping (PHP). The FIB entry for the tunnel on the PHP router shows a "pop label" as the outgoing label.

In some cases, it could be that the packets have two or three or more labels in the label stack. Then the implicit null label used at the tailend router would signal the penultimate hop router to pop one label and send the labeled packet with one label less to the tailend router. Then the tailend router does not have to perform two label lookups. The use of the implicit null label does not mean that all labels of the label stack must be removed; only one label is "popped" off (remove the top label on the stack). In any case, the use of the implicit null label prevents the tailend router from performing two lookups.

Restriction - The P2MP implicit null feature may cause multicast traffic drop with implicit null label on the tailend routers. This is because the P2MP implicit null feature does not support forwarding of multicast traffic when no label is received on the tailend router.

Configuring MPLS-TE

MPLS-TE requires co-ordination among several global neighbor routers. RSVP, MPLS-TE and IGP are configured on all routers and interfaces in the MPLS traffic engineering network. Explicit path and TE tunnel interfaces are configured only on the head-end routers. MPLS-TE requires some basic configuration tasks explained in this section.

Building MPLS-TE Topology

Building MPLS-TE topology, sets up the environment for creating MPLS-TE tunnels. This procedure includes the basic node and interface configuration for enabling MPLS-TE. To perform constraint-based routing, you need to enable OSPF or IS-IS as IGP extension.

Before You Begin

Before you start to build the MPLS-TE topology, the following pre-requisites are required:

- 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.
- Enable RSVP on the port interface.

Example

This example enables MPLS-TE on a node and then specifies the interface that is part of the MPLS-TE. Here, OSPF is used as the IGP extension protocol for information distribution.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface hundredGigE 0/0/0/3
Router(config)# router ospf area 1
Router(config-ospf)# area 0
Router(config-ospf-ar)# mpls traffic-eng
Router(config-ospf-ar)# interface hundredGigE 0/0/0/3
Router(config-ospf-ar-if)# exit
Router(config-ospf)# mpls traffic-eng router-id 192.168.70.1
Router(config)# commit
```

Example

This example enables MPLS-TE on a node and then specifies the interface that is part of the MPLS-TE. Here, IS-IS is used as the IGP extension protocol for information distribution.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface hundredGigE 0/0/0/3
Router(config)# router isis 1
Router(config-isis)# net 47.0001.0000.0000.0002.00
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
```

```
Router(config-isis-af)# mpls traffic-eng level 1
Router(config-isis-af)# exit
Router(config-isis)# interface hundredGigE 0/0/0/3
Router(config-isis-if)# exit
Router(config)# commit
```

Teardown and Reestablishment of RSVP-TE Tunnels

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Teardown and Reestablishment of RSVP-TE Tunnels	Release 7.11.1	You can now teardown and reestablish the existing tunnels of headend, midend, or tailend router tunnels of an MPLS network for optimized distribution of the traffic across MPLS and RSVP-TE to improve network performance and enhance resource utilization. Previously, you could reestablish tunnels only at the headend router using the mpls traffic-eng-resetup command. The feature introduces these changes: **CLI**: mpls traffic-eng teardown **YANG Data Model**: Cisco-IOS-XR-mpls-te-act.yang (see GitHub, YANG Data Models Navigator)

In an MPLS-TE network which is configured with RSVP-TE, the headend, midend, and tailend router work together to establish and maintain tunnels or adjacencies for traffic engineering purposes. When the headend router boots up, it plays a critical role in MPLS-TE tunnel establishment using RSVP-TE signaling along the computed path with the tailend router.

During a system reboot, if a tailend router comes up first, the Interior Gateway Protocol (IGP) adjacency establishes a path for the tailend with the adjacent network devices. However, if the headend router comes up and starts creating tunnels during this tailend path creation, it may result in a poor distribution of tunnels at the tailend. In such cases, it's necessary to tear down all the tunnels and recreate them to ensure optimal distribution and functioning of the RSVP-TE as per the network conditions.

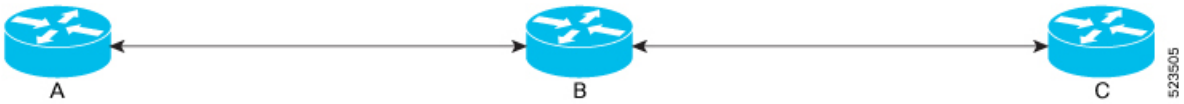
From Release 7.11.1, you can teardown and reestablish tunnels across all headend, midend, and tailend routers using the **mpls traffic-eng teardown** command with **all, head, mid, tail** parameters.

Table 2: Network Routers and Description

Router Name	Router Function	Tunnel Tear-Down and Reestablish Command
Headend Router	The headend router is responsible for determining the tunnel's path. It initiates the signaling process, specifying the tunnel's parameters and requirements. It sets up and manages the MPLS-TE tunnels.	**mpls traffic-eng teardown head**

Router Name	Router Function	Tunnel Tear-Down and Reestablish Command
Midend Router	The midend router, or an intermediate router or network device, is located between the headend and the tailend along the tunnel's path. It's involved in the signaling process and ensures the proper forwarding of traffic along the established tunnel.	**mpls traffic-eng teardown mid**
Tailend Router	The tailend router is the terminating router or network device located at the other end of the tunnel. It receives and processes the traffic that traverses through the tunnel and ensures proper traffic delivery to the destination.	**mpls traffic-eng teardown tail**

Figure 1: RSVP-TE tunnel Teardown and Reestablishment Topology



When RSVP-TE tunnel teardown message is triggered and:

- Router B is configured as tailend and Router A is configured as headend: A **ResvTear upstream** message is sent to the headend router. This message informs the headend router to tear down the RSVP-TE tunnel and release the resources associated with the tunnel.
- Router B is configured as headend and Router C as tailend: A **PathTear downstream** message is sent to the tailend router. A headend router triggers the process of recomputing the tunnels. The headend router (Router B) initiates the process of recomputing the tunnels, which involves recalculating the path and parameters for establishing new tunnels.
- Router B is configured as midend where Router A and Router B as headend and tailend respectively: A **ResvTear upstream** message is sent to the headend router and a **PathTear downstream** message is sent to the tailend router. These messages inform the respective routers to tear down the RSVP-TE tunnel and release the associated resources.

You can tear down and set up all types of tunnels including, P2P, P2MP, numbered, named, flex LSPs, and auto tunnels.

Limitations

- A maximum 90 seconds are required for the tunnels to get reestablished once they are torn.
- Use the **mpls traffic-eng resetup** command to reestablish the tunnels only at the headend router.

Configure Tear down and Reestablishment of RSVP-TE Tunnels

Configuration Example

Use the **mpls traffic-eng teardown all** command to tear down and reestablish all the RSVP-TE tunnels in a network node. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown all
```

Use the **mpls traffic-eng teardown head** command to tear down and reestablish the RSVP-TE tunnels at the headend router. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown head
```

Note You can also use the **mpls traffic-eng resetup** command to reestablish tunnels only at the headend router in XR EXEC mode.

Use the **mpls traffic-eng teardown mid** command to tear down and reestablish the RSVP-TE tunnels at the midend router. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown mid
```

Use the **mpls traffic-eng teardown tail** command to tear down and reestablish the RSVP-TE tunnels at the tailend router. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown tail
```

Use the **show mpls traffic-eng tunnels summary** command to check RSVP-TE tunnel status after you run the teardown command.

```
Router# show mpls traffic-eng tunnels summary
Output received:
Thu Sep 14 10:48:45.007 UTC

        Path Selection Tiebreaker:  Min-fill (default)
              LSP Tunnels Process:  running
                     RSVP Process:  running
                       Forwarding:  enabled
          Periodic reoptimization:  every 3600 seconds, next in 2806 seconds
           Periodic FRR Promotion:  every 300 seconds, next in 9 seconds
      Periodic auto-bw collection:  5 minute(s) (disabled)

Signalling Summary:
    Head: 14006 interfaces, 14006 active signalling attempts, 14006 established
          14006 explicit, 0 dynamic
          14006 activations, 0 deactivations
          0 recovering, 0 recovered
    Mids: 2000
    Tails: 4003

Fast ReRoute Summary:
    Head:     14000 FRR tunnels, 14000 protected, 0 rerouted
    Mid:      2000 FRR tunnels, 2000 protected, 0 rerouted
    Summary:  16000 protected, 13500 link protected, 2500 node protected, 0 bw protected
    Backup:   6 tunnels, 4 assigned
    Interface: 10 protected, 0 rerouted

Bidirectional Tunnel Summary:
    Tunnel Head: 0 total, 0 connected, 0 associated, 0 co-routed
    Tunnel Tail: 0 total, 0 connected, 0 associated, 0 co-routed
    LSPs Head:   0 established, 0 proceeding, 0 associated, 0 standby
    LSPs Mid:    0 established, 0 proceeding, 0 associated, 0 standby
    LSPs Tail:   0 established, 0 proceeding, 0 associated, 0 standby
```

Configuring Automatic Bandwidth

Automatic bandwidth allows you to dynamically adjust bandwidth reservation based on measured traffic. MPLS-TE automatic bandwidth monitors the traffic rate on a tunnel interface and resizes the bandwidth on the tunnel interface to align it closely with the traffic in the tunnel. MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every headend router.

Adjustment Threshold - It 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 signaled 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.

The following table specifies the parameters that can be configured as part of automatic bandwidth configuration.

Table 3: Automatic Bandwidth Parameters

Bandwidth Parameters	Description
Application frequency	Configures how often the tunnel bandwidths changed for each tunnel. The default value is 24 hours.
Bandwidth limit	Configures the minimum and maximum automatic bandwidth to set on a tunnel.
Bandwidth collection frequency	Enables bandwidth collection without adjusting the automatic bandwidth. The default value is 5 minutes.
Overflow threshold	Configures tunnel overflow detection.
Adjustment threshold	Configures the tunnel-bandwidth change threshold to trigger an adjustment.

Adjustment Threshold

Configuration Example

This example enables automatic bandwidth on MPLS-TE tunnel interface and configure the following automatic bandwidth variables.

- Application frequency
- Bandwidth limit
- Adjustment threshold
- Overflow detection

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# auto-bw
Router(config-if-tunte-autobw)# application 1000
Router(config-if-tunte-autobw)# bw-limit min 30 max 1000
Router(config-if-tunte-autobw)# adjustment-threshold 50 min 800
Router(config-if-tunte-autobw)# overflow threshold 100 limit 1
Router(config)# commit
```

Verification

Verify the automatic bandwidth configuration using the **show mpls traffic-eng tunnels auto-bw brief** command.

```
Router# show mpls traffic-eng tunnels auto-bw brief

  Tunnel    LSP  Last appl  Requested  Signalled    Highest    Application
  Name       ID   BW(kbps)   BW(kbps)   BW(kbps)   BW(kbps)      Time Left
  ------------- ------ ---------- ---------- ---------- ---------- --------------
  tunnel-te1      5          500         300        420          1h 10m
```

Configuring Automatic Capacity With Load-Interval Configuration

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Configuring Automatic Capacity With Load-Interval Configuration	Release 7.3.3	With this feature, you can enable the load-interval configuration for a main tunnel's clones, along with the automatic capacity feature.

The auto-bandwidth feature resizes MPLS-TE tunnels based on traffic loads. Multiple auto-bandwidth tunnels can be created for balancing traffic loads and redundancy.

The auto-capacity feature is an extension of the auto-bandwidth feature. With auto-capacity, for an auto-bandwidth enabled MPLS-TE tunnel, you can enable automatic creation and deletion of tunnels based on real-time capacity demands. These tunnels are called *clones*. For a main TE tunnel, you can specify the minimum and maximum number of clones, and allocate a nominal tunnel bandwidth value. Clones are automatically added to, or removed from, the main TE tunnel, based on the nominal bandwidth.

Consider the auto-capacity configuration example:

```
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels tunnel-te YOW2YZZ


Router(config-te-tun-name)# load-interval 90
Router(config-te-tun-name)# auto-bw
Router(config-mpls-te-tun-autobw)# auto-capacity
Router(config-mpls-te-tun-autobw)# commit
```

retries to establish the LSPs. The timeout range is 1 to 600 seconds.

The auto-capacity function is disabled by default. Since auto-bandwidth and auto-capacity functions are inter-related, these are the corresponding changes in behavior:

- When you enable auto-capacity, it is associated with a specific TE-tunnel, under the auto-bandwidth function. When you disable auto-bandwidth, auto-capacity is also disabled.
- If the load interval is enabled for the main tunnel, it is automatically applied to its clones too. For a main tunnel, if the auto-capacity feature is enabled but a load interval is not enabled, the clones' load interval value is set to a default of 300 seconds.

Splitting and Merging Tunnels

When there is a change in demand for bandwidth, MPLS-TE adds or reduces the number of tunnels and resizes the bandwidth of all the tunnels. It verifies these rules during this activity.

1. The number of tunnels between the headend-tailend router pair is within the specified range, and the bandwidth per tunnel is within the auto-bandwidth range.
2. The (*Bandwidth-per-tunnel*)*(*Number-of-Tunnels*) >= *Total-tunnel-bandwidth* requirement.

   While Rule 1 is enforced, MPLS-TE attempts to enforce Rule 2.
3. When the split requirement is met, and the maximum number of clones is not reached, at least one extra clone is added.

   When the merge requirement is met, and the minimum number of clones is not reached, at least one clone is removed
4. The nominal bandwidth value is used to balance the requirements of: (a) Number of tunnels and (b) Bandwidth for each tunnel. This helps in avoiding a merging or splitting instance at the next application event.

Configurations

```
/* Automatic Capacity Function */

Router# configure
Router(config)# mpls traffic-eng
```

The auto-capacity feature is only valid for the main tunnel **YOW2YZZ**, with reference to which, clones are created or removed.

```
Router(config-mpls-te)# named-tunnels tunnel-te YOW2YZZ
Router(config-te-tun-name)# auto-bw auto-capacity
```

MPLS-TE maintains the number of clones between 1 and 7. Including the main tunnel **YOW2YZZ**, the tunnel count range is between 2 and 8.

```
Router(config-te-tun-autocapacity)# max-clones 7
Router(config-te-tun-autocapacity)# min-clones 1
```

The nominal-bandwidth option is used for specifying the target bandwidth based on which MPLS-TE calculates the number of required tunnels.

```
Router(config-te-tun-autocapacity)# nominal-bandwidth 2000000
```

MPLS-TE also uses the merge-bandwidth and split-bandwidth values when implementing the auto-capacity feature.

```
Router(config-te-tun-autocapacity)# merge-bandwidth 1000000
Router(config-te-tun-autocapacity)# split-bandwidth 3000000
Router(config-te-tun-autocapacity)# commit
```

Verification

```
/* View the Auto-Capacity Feature Configuration */

Router# show mpls traffic-eng tunnels name YOW2YZZ

Name: YOW2YZZ     Ifhandle:0xf000014
  ..
  Config Parameters:
  ..
    Load-interval: 300 seconds  ..
Auto-Capacity: Enabled
  Minimum Clones: 1; Maximium Clones: 7
  Nominal BW: 2000000 kbps; Merge BW: 1000000 kbps; Split BW: 3000000 kbps
  Statistics:
    Splits: 0; Merges: 0
    Clones Created: 1; Clones Deleted: 0
    Clones High Watermark: 1
  Number of clones: 1
    Clone: YOW2YZZ-1
      Created: Thu Jan 27 13:55:06 2022; State: down
  ..
```

Configuring Auto-Bandwidth Bundle TE++

Table 5: Feature History Table

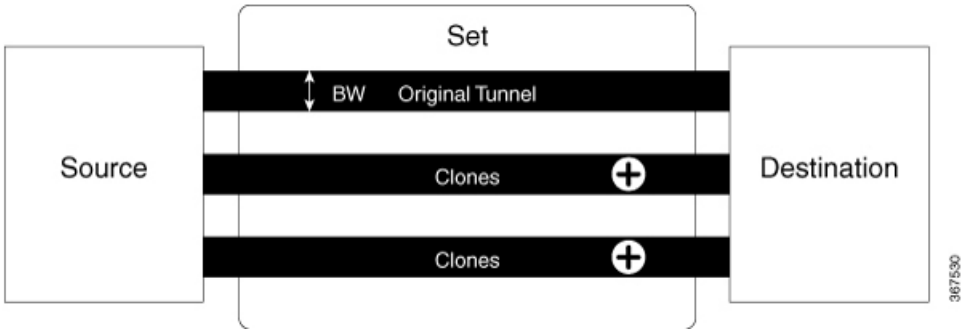
Feature Name	Release Information	Feature Description
MPLS-TE++ Support Over Numbered Tunnel-TE Interfaces	Release 7.10.1	We have optimized network performance and enabled efficient utilization of resources for numbered tunnels based on real-time traffic by automatically adding or removing tunnels between two endpoints. This is made possible because this release introduces support for auto-bandwidth TE++ for numbered tunnels, expanding upon the previous support for only named tunnels, letting you define explicit paths and allocate the bandwidth to each tunnel. The feature introduces these changes: • **CLI:** The **auto-capacity** keyword is added to the **interface tunnel-te** command. • **YANG Data Model:** New XPaths for `Cisco-IOS-XR-mpls-te-cfg.yang` (see GitHub, YANG Data Models Navigator)

An MPLS-TE tunnel sets up labeled connectivity and provides dynamic bandwidth capacity between its endpoints. The auto-bandwidth function addresses the dynamic bandwidth capacity demands by resizing the MPLS-TE tunnels based on the measured traffic loads. However, many customers require multiple auto-bandwidth tunnels between two endpoints for load balancing and redundancy. The auto-bandwidth bundle TE++ function is an extension of the auto-bandwidth feature, and provides this support. When the aggregate bandwidth between the endpoints changes, MPLS-TE creates new tunnels or removes existing tunnels to load balance the traffic.

When MPLS-TE automatically creates new tunnels to meet increasing bandwidth demands, they are called clones. The original tunnel and its clones collectively form a *set*. The clones inherit the properties of the main tunnel.You can specify an upper limit and lower limit on the number of clones.

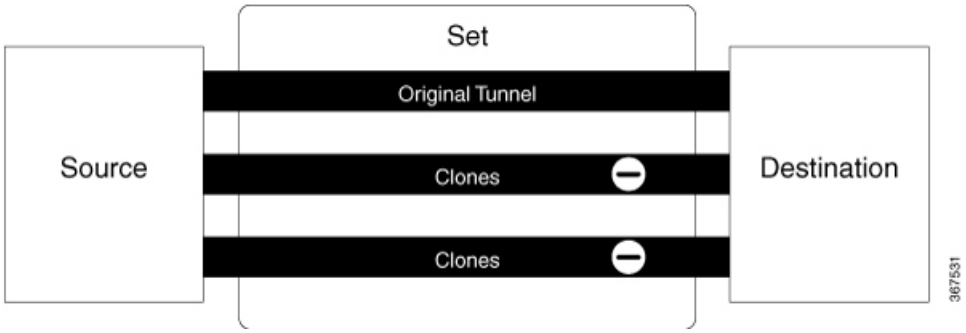
Splitting is the process of creating a new clone. When the bandwidth of a tunnel crosses the split bandwidth value, MPLS-TE creates a clone.

The following figure explains how MPLS-TE creates clones when the split bandwidth exceeds the configured value.



Merging is the process of removing a clone. If the bandwidth goes below the merge bandwidth value in a set of tunnels, MPLS-TE removes a clone.

The following figure explains how MPLS-TE removes clones when the bandwidth falls below the merge bandwidth value.



There are multiple ways to *load-share* the aggregate bandwidth demand among the tunnels in a set. An algorithm chooses the pair that satisfies the aggregate bandwidth requirements. You can configure a nominal bandwidth to guide the algorithm that determines the average bandwidth of the tunnels. If you don't configure, MPLS-TE uses the average of the split bandwidth and merge bandwidth values as the nominal bandwidth.

Restrictions and Guidelines

The following guidelines and restrictions apply for the auto-bandwidth bundle TE++ feature.

- The range for the lower limit on the number of clones is 0–63. The default value is 0. The upper limit range is 1–63. The default value is 63.

Configure Auto-Bandwidth Bundle TE++

Configure the following parameters:

- **min-clones**: Specifies the minimum number of clones that the original tunnel can create.
- **max-clones**: Specifies the maximum number of clones that the original tunnel can create.
- **nominal-bandwidth**: Specifies the average bandwidth for computing the number of tunnels to satisfy the overall demand.

- **split-bandwidth**: Specifies the bandwidth for splitting the original tunnel. If the tunnel bandwidth exceeds the configured split bandwidth, MPLS-TE creates clones.
- **merge-bandwidth**: Specifies the bandwidth for merging clones with the original tunnel. If the bandwidth goes below the merge bandwidth value, MPLS-TE removes the clones.

Configuration Example: Named MPLS-TE Tunnel

This example shows how to configure the auto-bandwidth bundle TE++ feature for a named MPLS-TE tunnel.

Here, the lower and upper limits on the number of clones are two and four, respectively. The bandwidth size for splitting and merging are 200 kbps and 100 kbps, respectively.

```
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels
Router(config-te-named-tunnels)# tunnel-te xyz
Router(config-te-tun-name)# auto-bw
Router(config-mpls-te-tun-autobw)# auto-capacity
Router(config-te-tun-autocapacity)# min-clones 2
Router(config-te-tun-autocapacity)# max-clones 4
Router(config-te-tun-autocapacity)# nominal-bandwidth 150
Router(config-te-tun-autocapacity)# split-bandwidth 200
Router(config-te-tun-autocapacity)# merge-bandwidth 100
```

Configuration Example: Numbered TE-Tunnel

This example shows how to configure the auto-bandwidth bundle TE++ feature for a numbered te-tunnel. The lower limit and the upper limit of clones are 3 and 10, respectively. The bandwidth size for splitting and merging are 3,000,000 kbps and 1,000,000 kbps, respectively.

```
Router(config)#interface tunnel-te 20
Router(config-if)#load-interval 90
Router(config-if)#auto-bw auto-capacity
Router(config-if-tunte-autocapacity)#max-clones 10
Router(config-if-tunte-autocapacity)#min-clones 3
Router(config-if-tunte-autocapacity)#nominal-bandwidth 2000000
Router(config-if-tunte-autocapacity)#merge-bandwidth 1000000
Router(config-if-tunte-autocapacity)#split-bandwidth 3000000
Router(config-if-tunte-autocapacity)#commit
```

Running Configuration

```
Router# show run interface tunnel-te20

interface tunnel-te20
 auto-bw
  auto-capacity
   merge-bandwidth 1000000
   split-bandwidth 3000000
   max-clones 10
   min-clones 3
   nominal-bandwidth 2000000
```

Verification

Here, you can verify that MPLS-TE has created the te-tunnel **tunnel-te20** and three clones.

```
Router# show mpls traffic-eng tunnels name tunnel-te20 detail

Name: tunnel-te20     Ifhandle:0xf00002c
  Signalled-Name: ios_t20
  Status:
    Admin:    up Oper: down   Path: not valid   Signalling: Down
    G-PID: 0x0800 (derived from egress interface properties)
```

```
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Wed May 31 11:45:58 2023 (00:04:35 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Delay-limit: disabled
    Delay-measurement: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (not enabled)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Load-interval: 300 seconds
    Auto-bw: enabled
      No BW Applied
      Currently no BW application allowed because: Tunnel is down
      Bandwidth Min/Max: 0-4294967295 kbps
      Application Frequency: 1440 min   Jitter: 0s   Time Left: 23h 55m 30s
      Collection Frequency: 5 min
      Samples Collected: 0   Next: 2m 5s
      Highest BW: 0 kbps   Underflow BW: 0 kbps
      Adjustment Threshold: 10%   10 kbps
      Overflow Detection disabled
      Underflow Detection disabled
      Resignal Last-bandwidth Disabled
    Auto-Capacity: Enabled
      Minimum Clones: 3; Maximium Clones: 10
      Nominal BW: 200000 kbps; Merge BW: 1000000 kbps; Split BW: 3000000 kbps
      Statistics:
        Splits: 0; Merges: 0
        Clones Created: 3; Clones Deleted: 0
        Clones High Watermark: 3
      Number of clones: 3
        Clone: tunnel-te20-1
          Created: Wed May 31 11:45:58 2023; State: down
          Signaled BW: 0 kbps; Current BW Demand: 0 kbps
        Clone: tunnel-te20-2
          Created: Wed May 31 11:45:58 2023; State: down
          Signaled BW: 0 kbps; Current BW Demand: 0 kbps
        Clone: tunnel-te20-3
          Created: Wed May 31 11:45:58 2023; State: down
          Signaled BW: 0 kbps; Current BW Demand: 0 kbps
    Self-ping: Disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  Reason for the tunnel being down: No destination is configured
  SNMP Index: 105
  Binding SID: None
  Persistent Forwarding Statistics:
    Out Bytes: 0
    Out Packets: 0

Displayed 1 (of 7) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 0 up, 1 down, 0 recovering, 0 recovered heads
```

Configuring Auto-Tunnel Backup

The MPLS Traffic Engineering Auto-Tunnel Backup feature enables a router to dynamically build backup tunnels on the interfaces that are configured with MPLS TE tunnels instead of building MPLS-TE tunnels statically.

The MPLS-TE Auto-Tunnel Backup feature has these benefits:

- Backup tunnels are built automatically, eliminating the need for users to pre-configure 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.

The TE attribute-set template that specifies a set of TE tunnel attributes, is locally configured at the headend of auto-tunnels. The control plane triggers the automatic provisioning of a corresponding TE tunnel, whose characteristics are specified in the respective attribute-set.

Configuration Example

This example configures Auto-Tunnel backup on an interface and specifies the attribute-set template for the auto tunnels. In this example, unused backup tunnels are removed every 20 minutes using a timer and also the range of tunnel interface numbers are specified.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# auto-tunnel backup
Router(config-mpls-te-if-auto-backup)# attribute-set ab
Router(config-mpls-te)# auto-tunnel backup timers removal unused 20
Router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
Router(config-mpls-te)# commit
```

Verification

This example shows a sample output for automatic backup tunnel configuration.

```
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
```

Removing an AutoTunnel Backup

To remove all the backup autotunnels, perform this task.

Configuration Example

```
Router# clear mpls traffic-eng auto-tunnel backup unused all
```

Verification

Use the **show mpls traffic-eng auto-tunnel summary** command to verify MPLS-TE autotunnel information, including the ones removed.

Configuring Auto-Tunnel Mesh

The MPLS-TE auto-tunnel mesh (auto-mesh) feature allows you to set up full mesh of TE Point-to-Point (P2P) tunnels automatically with a minimal set of MPLS traffic engineering configurations. You can configure one or more mesh-groups and 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 can configure MPLS-TE auto-mesh type attribute-sets (templates) and associate them to mesh-groups. Label Switching Routers (LSRs) can create tunnels using the tunnel properties defined in this attribute-set.

Auto-Tunnel mesh configuration minimizes the initial configuration of the network. You can configure tunnel properties template and mesh-groups or destination-lists on TE LSRs that further creates full mesh of TE tunnels between those LSRs. 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.

Configuration Example

This example configures an auto-tunnel mesh group and specifies the attributes for the tunnels in the mesh-group.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# auto-tunnel mesh
Router(config-mpls-te-auto-mesh)# tunnel-id min 1000 max 2000
Router(config-mpls-te-auto-mesh)# group 10
Router(config-mpls-te-auto-mesh-group)# attribute-set 10
Router(config-mpls-te-auto-mesh-group)# destination-list dl-65
Router(config-mpls-te)# attribute-set auto-mesh 10
Router(config-mpls-te-attribute-set)# autoroute announce
Router(config-mpls-te-attribute-set)# auto-bw collect-bw-only
Router(config)# commit
```

Verification

Verify the auto-tunnel mesh configuration using the **show mpls traffic-eng auto-tunnel mesh** command.

```
Router# show mpls traffic-eng auto-tunnel mesh

Auto-tunnel Mesh Global Configuration:
  Unused removal timeout: 1h 0m 0s
  Configured tunnel number range: 1000-2000

Auto-tunnel Mesh Groups Summary:
  Mesh Groups count: 1
  Mesh Groups Destinations count: 3
  Mesh Groups Tunnels count:
    3 created, 3 up, 0 down, 0 FRR enabled

Mesh Group: 10 (3 Destinations)
  Status: Enabled
  Attribute-set: 10
  Destination-list: dl-65 (Not a prefix-list)
  Recreate timer: Not running
       Destination    Tunnel ID    State  Unused timer
  ---------------  -----------  -------  ------------
```

```
      192.168.0.2         1000     up   Not running
      192.168.0.3         1001     up   Not running
      192.168.0.4         1002     up   Not running
  Displayed 3 tunnels, 3 up, 0 down, 0 FRR enabled

Auto-mesh Cumulative Counters:
  Last cleared: Wed Oct  3 12:56:37 2015 (02:39:07 ago)
                     Total
  Created:               3
  Connected:             0
  Removed (unused):      0
  Removed (in use):      0
  Range exceeded:        0
```

Enable LDP over TE Automatically on Cloned Tunnels

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Enable LDP over TE Automatically on Cloned Tunnels	Release 7.5.3	You can now enable LDP automatically on all the cloned tunnels and associate the parent and cloned tunnels. This feature helps you to overcome the 1k tunnel limitation under LDP because the cloned tunnels are not accounted for in the global count. Earlier, LDP had to be enabled on all the cloned tunnels manually, which was time-consuming. This feature introduces the **clone-tunnels** command.

When you run tunnels end-to-end from a PE device to a remote PE, there's no need to enable LDP on tunnels. However, this approach requires a full mesh of tunnels which may lead to a high tunnel scale in large networks.

So in a large network, you may prefer to start a tunnel on a midpoint router. In this case, enable LDP over the tunnel so that a label path can be established end-to-end.

When LDP is enabled on named TE tunnels, and when the tunnels are cloned, currently there's no way to configure LDP automatically on all the cloned tunnels. You have to enable LDP on all the cloned tunnels manually, which might be time consuming.

Now you can enable LDP automatically for the named tunnel interfaces with minimal configuration steps. You don't need to configure hundreds of tunnels under LDP explicitly. Use the **clone-tunnel** command to enable LDP automatically on all the cloned tunnels.

Though LDP has 1K tunnel-interface limitation, cloned tunnels that are present is not considered in the global count.

For information about cloned tunnels, see Configuring Autobandwidth Bundle TE++.

Configure LDP over TE on Cloned Tunnels

You can enable LDP over TE automatically on cloned tunnels using the **clone-tunnel** configuration under **mpls-ldp address family ipv4**.

```
Router(config)#configure
Router(config)#mpls ldp
Router(config-ldp)#nsr
Router(config-ldp)#router-id 10.10.1.1
Router(config-ldp)#address-family ipv4
Router(config-ldp-af)#discovery targeted-hello accept
Router(config-ldp-af)#exit
Router(config-ldp)#interface tunnel-te n1
Router(config-ldp-if)#address-family ipv4
Router(config-ldp-if-af)#clone-tunnel
Router(config-ldp-if-af)#exit
Router(config-ldp-if)#interface tunnel-te n2
Router(config-ldp-if)#address-family ipv4
Router(config-ldp-if-af)#clone-tunnel
Router(config-ldp-if-af)#exit
Router(config-ldp-if)#interface tunnel-te n3
Router(config-ldp-if)#address-family ipv4
Router(config-ldp-if-af)#clone-tunnel
Router(config-ldp-if-af)#commit
```

Running Configuration

This section shows LDP over TE on cloned tunnels running configuration.

```
Router#show running-config mpls ldp
mpls ldp
log
  hello-adjacency
  neighbor
  nsr
  session-protection
!
nsr
router-id 10.10.1.1
address-family ipv4
  discovery targeted-hello accept
!
interface tunnel-te n1
  address-family ipv4
   clone-tunnel
  !
!
interface tunnel-te n2
  address-family ipv4
   clone-tunnel
  !
!
interface tunnel-te n3
  address-family ipv4
   clone-tunnel
  !
!
!
```

Verification

Verify that LDP is enabled on cloned tunnels.

```
Router#show mpls ldp interface
Interface n1 (0xf000014)
   VRF: 'default' (0x60000000)
   Enabled via config: LDP interface
Interface n1-1 (0xf001f84)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n1-2 (0xf001f8c)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n1-3 (0xf001fb4)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n1-4 (0xf001fbc)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n2 (0xf00001c)
   VRF: 'default' (0x60000000)
   Enabled via config: LDP interface
Interface n2-1 (0xf001f64)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n3 (0xf000024)
   VRF: 'default' (0x60000000)
   Enabled via config: LDP interface
Interface n3-1 (0xf001f74)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n3-2 (0xf001fc4)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
```

Attribute Set for Named Tunnels

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Attribute Set for Named Tunnels	Release 7.5.3	You can now add an attribute set to the named tunnels under the path options eliminating the need to configure the TE-tunnel attributes individually. When you change the attribute set values for the path option, the same values are inherited to all the attached path options under named tunnels. The attribute set is a template that specifies a set of tunnel attributes, such as priority, affinity, signaled bandwidth, logging, policy class, record route and so on. Earlier, the attribute set was available only for numbered tunnels. The **attribute-set** keyword is added in the **path option (Named Tunnels)**.

In the traditional TE tunnel naming scheme, the tunnels are configured with IDs, where an ID is a 16-bit number. With the increased TE tunnel scale in the network, and with the 64K limit, there is a scarcity of unique tunnel IDs. So there came the need for named tunnels.

The named tunnel allows you to provision TE tunnels using a string name, which lets you differentiate various TE tunnels.

For a tunnel, you can configure one or more path options and each is identified by a unique name. You can configure the path option attributes through a template. This template is named as attribute set and configured under MPLS traffic-engineering tunnel. 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 TE attribute set template specifies a set of TE tunnel attributes and is configured at the headend of autotunnels. The control plane triggers the automatic provisioning of a corresponding TE tunnel, whose characteristics are specified in the respective attribute set.

Earlier, the attribute set was configurable only for numbered tunnels. With this feature, you can now apply an attribute set to a path option on a per-LSP basis for a named tunnel. The path option configuration is extended to take a path option attribute name. LSPs compute a path option using the set attributes as specified by the attribute set under the path option.

Configure Attribute Set for Named Tunnels

Perform the following task to configure an attribute set for the named tunnel:

- Define attribute set for path option.

- Associate the attribute set to the named tunnel.

```
/* Define attribute set for path option */
Router#configure
Router(config)#mpls traffic-eng
Router(config-mpls-te)#attribute-set path-option P1
Router(config-te-attribute-set)#signalled-bandwidth 10000 class-type 0
Router(config-te-attribute-set)# path-selection
Router(config-te-attrset-psel)#cost-limit 1000
Router(config-te-attrset-psel)#delay-limit 5000
Router(config-te-attrset-psel)#exclude AAA
Router(config-te-attrset-psel)#exit
Router(config-te-attribute-set)#affinity include YELLOW
Router(config-te-attribute-set)#affinity include-strict BLUE GREEN
Router(config-te-attribute-set)#affinity exclude RED
Router(config-te-attribute-set)#affinity exclude-all
Router(config-te-attribute-set)#commit
Router(config-te-attribute-set)#root

/* Associate the attribute set to the named tunnel */
Router(config)#mpls traffic-eng
Router(config-mpls-te)#named-tunnels
Router(config-te-named-tunnels)#tunnel-te cisco
Router(config-te-tun-name)#path-option 1
Router(config-po-name)#attribute-set P1
Router(config-po-name)#commit
Router(config-po-name)#root
Router(config)#exit
```

Running Configuration

This section shows the running configurations.

```
Router#show run
mpls traffic-eng
 attribute-set path-option P1
  signalled-bandwidth 10000 class-type 0
  path-selection
   cost-limit 1000
   delay-limit 5000
   exclude AAA
  !
  affinity include YELLOW
  affinity include-strict BLUE GREEN
  affinity exclude RED
  affinity exclude-all
 !
 named-tunnels
  tunnel-te cisco
   path-option 1
    attribute-set P1
   !
  !
 !
!
```

Verification

Verify that the attribute set is associated with the named tunnel.

```
Router#show mpls traffic-eng tunnel name cisco detail

Name: cisco  Destination: 0.0.0.0  Ifhandle:0xf000014
  Tunnel-ID: 32768
  Status:
    Admin:    up Oper: down   Path: not valid   Signalling: Down
    path option 1, preference unspecified,  type unspecified
      Path-option attribute: P1
        Number of affinity constraints: 4
           Include bit map           : 0x0
           Include ext bit map       :
               Length: 256 bits
               Value : 0x::
           Include affinity name     :
           Undefined affinity name   : YELLOW
           Include-strict bit map    : 0x0
           Include-strict ext bit map:
               Length: 256 bits
               Value : 0x::
           Include-strict name       :
           Undefined affinity name   : BLUE GREEN
           Exclude bit map           : 0x0
           Exclude ext bit map       :
               Length: 256 bits
               Value : 0x::
           Exclude affinity name     :
           Undefined affinity name   : RED
           Exclude all
```

Configuring 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. The path of the backup tunnel can be an IP explicit path, a dynamically calculated path, or a semi-dynamic path. For detailed conceptual information on fast reroute, see the MPLS-TE Features - Details topic.

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.

Configuration Example

This example configures fast reroute on an MPLS-TE tunnel. Here, tunnel-te 2 is configured as the back-up tunnel. You can use the **protected-by** command to configure path protection for an explicit path that is protected by another path.

```
Router# configure
Router(config)# interface tunnel-te 1
```

```
Router(config-if)# fast-reroute
Router(config-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# backup-path tunnel-te 2
Router(config)# interface tunnel-te 2
Router(config-if)# backup-bw global-pool 5000
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 192.168.92.125
Router(config-if)# path-option 1 explicit name backup-path protected by 10
Router(config-if)# path-option 10 dynamic
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng fast-reroute database** command to verify the fast reroute configuration.

```
Router# show mpls traffic-eng fast-reroute database

Tunnel head FRR information:
Tunnel      Out intf/label                    FRR intf/label      Status
----------  ----------------                  ----------------    -------
tt4000       HundredGigabitEthernet 0/0/0/3:34     tt1000:34        Ready
tt4001       HundredGigabitEthernet 0/0/0/3:35     tt1001:35        Ready
tt4002       HundredGigabitEthernet 0/0/0/3:36     tt1001:36        Ready
```

Configuring 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 the MPLS-TE tunnels.

In traditional TE, 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.

Cisco IOS XR Release 7.5.4 introduces support to configure the affinities to include any of the affinity names for the links. The link with any additional colors or a link without a color is also accepted.

Configuration Example

This example shows assigning a how to associate a tunnel with affinity constraints.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# affinity-map red 1
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# attribute-names red
Router(config)# interface tunnel-te 2
Router(config-if)# affinity include red
Router(config)# commit
```

Configuring Forwarding Path

Perform this task to configure forwarding path in the MPLS-TE interface.

Configuration Example

```
Router # configure
Router(config)# interface tunnel-te 1
Router(config-if)# forward-class 1
Router(config-if)# exit
Router(config)# commit
```

Configuring an IETF DS-TE Tunnel Using MAM

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 Russian Doll Model (RDM) and Maximum Allocation Model (MAM), both with two bandwidth pools.

Configuration Example

This example configures an IETF DS-TE tunnel using MAM.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth mam max-reservable-bw 1000 bc0 600 bc1 400
Router(config-rsvp-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# ds-te mode ietf
Router(config-mpls-te)# ds-te bc-model mam
Router(config-mpls-te)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the IETF DS-TE tunnel using MAM configuration.

Configuring an IETF DS-TE Tunnel Using RDM

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 Russian Doll Model (RDM) and Maximum Allocation Model (MAM), both with two bandwidth pools. In an IETF DS-TE network, identical bandwidth constraint models must be configured on all nodes.

Before you Begin

The following prerequisites are required to create a 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.

Configuration Example

This example configures an IETF DS-TE tunnel using RDM.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth rdm 100 150
Router(config-rsvp-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# ds-te mode ietf
Router(config-mpls-te)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10 class-type 1
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the IETF DS-TE tunnel using RDM configuration.

Configuring an MPLS Traffic Engineering Interarea Tunneling

The MPLS TE Interarea Tunneling feature allows you to establish MPLS TE tunnels that span multiple Interior Gateway Protocol (IGP) areas and levels. This feature removes the restriction that required the tunnel headend and tailend routers both to be in the same area. The IGP can be either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).To configure an inter-area tunnel, you specify on the headend router a loosely routed explicit path for the tunnel label switched path (LSP) that identifies each area border router (ABR) the LSP should traverse using the next-address loose command. The headend router and the ABRs along the specified explicit path expand the loose hops, each computing the path segment to the next ABR or tunnel destination.

Configuration Example

This example configures an IPv4 explicit path with ABR configured as loose address on the headend router.

```
Router# configure
Router(config)# explicit-path name interarea1
Router(config-expl-path)# index1 next-address loose ipv4 unicast 172.16.255.129
Router(config-expl-path)# index 2 next-address loose ipv4 unicast 172.16.255.131
Router(config)# interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 172.16.255.2
Router(config-if)# path-option 10 explicit name interarea1
Router(config)# commit
```

Configuring MPLS-TE Path Protection

Path protection provides an end-to-end failure recovery mechanism 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. Failover is triggered by a RSVP error message sent to the LSP head end. Once the head end received this error message, it switches over to the secondary tunnel. 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. Both the explicit and dynamic path-options are supported for the MPLS-TE path protection feature. You should make sure that the same attributes or bandwidth requirements are configured on the protected option.

Before You Begin

The following prerequisites are required for enabling path protection.

- You should ensure that your network supports MPLS-TE, Cisco Express Forwarding, and Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).
- You should configure MPLS-TE on the routers.

Configuration Example

This example configures how to configure path protection for a mpls-te tunnel. The primary path-option should be present to configure path protection. In this configuration, R1 is the headend router and R3 is the tailend router for the tunnel while R2 and R4 are mid-point routers. In this example, 6 explicit paths and 1 dynamic path is created for path protection. You can have upto 8 path protection options for a primary path.

Note **Path-protection** through user-specified path-options is not supported and the **protected-by** is used specifically only for numbered tunnels and unavailable for named-tunnels.

```
Router # configure
Router(config)# interface tunnel-te 0
Router(config-if)# destination 192.168.3.3
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# autoroute announce
Router(config-if)# path-protection
Router(config-if)# path-option 1 explicit name r1-r2-r3-00 protected-by 2
Router(config-if)# path-option 2 explicit name r1-r2-r3-01 protected-by 3
Router(config-if)# path-option 3 explicit name r1-r4-r3-01 protected-by 4
Router(config-if)# path-option 4 explicit name r1-r3-00 protected-by 5
Router(config-if)# path-option 5 explicit name r1-r2-r4-r3-00 protected-by 6
Router(config-if)# path-option 6 explicit name r1-r4-r2-r3-00 protected-by 7
Router(config-if)# path-option 7 dynamic
Router(config-if)# exit
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng tunnels** command to verify the MPLS-TE path protection configuration.

```
Router# show mpls traffic-eng tunnels 0

Name: tunnel-te0  Destination: 192.168.92.125  Ifhandle:0x8007d34
  Signalled-Name: router
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1,  type explicit r1-r2-r3-00 (Basis for Setup, path weight 2)
      Protected-by PO index: 2
    path option 2,  type explicit r1-r2-r3-01 (Basis for Standby, path weight 2)
      Protected-by PO index: 3
    path option 3,  type explicit r1-r4-r3-01
      Protected-by PO index: 4
    path option 4,  type explicit r1-r3-00
      Protected-by PO index: 5
    path option 5,  type explicit r1-r2-r4-r3-00
      Protected-by PO index: 6
    path option 6,  type explicit r1-r4-r2-r3-00
      Protected-by PO index: 7
    path option 7,  type dynamic
```

```
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Fri Oct 13 15:05:28 2017 (01:19:11 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Delay-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (not enabled)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Current LSP:
      Uptime: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
    Standby Reopt LSP:
      Last Failure:
        LSP not signalled, identical to the [STANDBY] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
        First Destination Failed: 192.3.3.3
    Prior LSP:
      ID: 8 Path Option: 1
      Removal Trigger: path protection switchover
    Standby LSP:
      Uptime: 01:13:56 (since Fri Oct 13 15:10:43 UTC 2017)
  Path info (OSPF 1 area 0):
  Node hop count: 2
  Hop0: 192.168.1.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
  Standby LSP Path info (OSPF 1 area 0), Oper State: Up :
  Node hop count: 2
  Hop0: 192.168.2.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
Displayed 1 (of 4001) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Configuring Next Hop Backup Tunnel

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

Configuration Example

This example configures next hop backup tunnel on an interface and specifies the attribute-set template for the auto tunnels. In this example, unused backup tunnels are removed every 20 minutes using a timer and also the range of tunnel interface numbers are specified.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# auto-tunnel backup nhop-only
Router(config-mpls-te-if-auto-backup)# attribute-set ab
Router(config-mpls-te)# auto-tunnel backup timers removal unused 20
Router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
Router(config)# commit
```

Configuring Point-to-Multipoint TE Tunnels

For P2MP tunnels, a Cisco 8000 Series router supports the mid-point router function, and does not support source or receiver functions. To know how to configure a source or receiver (destination) router in a P2MP tunnel, refer the MPLS configuration guide for the corresponding platform.

Configuring Point-to-Multipoint TE Auto-Tunnels

The P2MP-TE Auto-tunnels feature enables dynamic creation and management of P2MP auto-tunnels for the transport of VPLS traffic on Cisco IOS XR Software. The P2MP-TE auto-tunnel configuration is disabled by default. Use the **auto-tunnel p2mp tunnel-id** command to enable a P2MP-TE Auto-tunnel. This configures the tunnel ID range that can be allocated to P2MP auto-tunnels. This also determines the maximum number of P2MP auto-tunnels that can be created.

Configuration Example

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# auto-tunnel p2mp
Router(config-te-auto-p2mp)# tunnel-id min 10000 max 11000
Router(config-te-auto-p2mp)# commit
```

Enabling Soft-Preemption

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.

Configuration Example

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.

```
Router# configure
Router(config)# mpls traffic-eng
```

```
Router(config-mpls-te)# soft-preemption
Router(config-soft-preemption)# timeout 100
Router(config-soft-preemption)# commit
```

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.

Configuration Example

When soft preemption is enabled on a tunnel, a path-modify message is sent for the current LSP, reopt LSP, path protection LSP, and current LSP in FRR active state, with the **soft preemption desired** property.

```
Router# configure
Router(config)# interface tunnel-te 10
Router(config-if)# soft-preemption
Router(config-if)# commit
```

Enabling Soft-preemption over FRR Backup Tunnels

Before enabling soft-preemption over FRR backup, ensure that you enable soft-preemption, and activate the FRR backup tunnel.

Configuration Example

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# soft-preemption frr-rewrite
Router(config-mpls-te)# commit
```

Configuring Pre-Standard DS-TE

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. MPLS DS-TE enables you to configure multiple bandwidth constraints on an MPLS-enabled interface. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint. Cisco IOS XR software supports two DS-TE modes: Pre-standard and IETF. Pre-standard 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. Pre-standard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces.

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

Before You Begin

The following prerequisites are required to configure a Pre-standard 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.

Configuration Example

This example configures a pre-standard DS-TE tunnel.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth 100 150 sub-pool 50
Router(config-rsvp-if)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the pre-standard DS-TE tunnel configuration.

Configuring SRLG Node Protection

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

OSPF and 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.

MPLS-TE SRLG feature enhances backup tunnel path selection by avoiding using links that are in the same SRLG as the interfaces it is protecting while creating backup tunnels.

Configuration Example

This example creates a backup tunnel and excludes the protected node IP address from the explicit path.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# backup-path tunnl-te 2
Router(config-mpls-te-if)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# path-option 1 explicit name backup-srlg
Router(config-if)# destination 192.168.92.125
Router(config-if)# exit
Router(config)# explicit-path name backup-srlg-nodep
Router(config-if)# index 1 exclude-address 192.168.91.1
Router(config-if)# index 1 exclude-srlg 192.168.92.2
Router(config)# commit
```

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 signaled, they are verified dynamically in the next path verification cycle.

Creating an MPLS-TE Tunnel

Creating an MPLS-TE tunnel is a process of customizing the traffic engineering to fit your network topology. The MPLS-TE tunnel is created at the headend router. You need to specify the destination and path of the TE LSP.

To steer traffic through the tunnel, you can use the following ways:

- Static Routing
- Autoroute Announce
- Forwarding Adjacency

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.

Configuration Example

This example configures an MPLS-TE tunnel on the headend router with a destination IP address 192.168.92.125. The bandwidth for the tunnel, path-option, and forwarding parameters of the tunnel are also configured. You can use static routing, autoroute announce or forwarding adjacency to steer traffic through the tunnel.

Cisco IOS XR Release 7.5.4 introduces support to configure the timeout period before the headend router retries to establish the LSPs. The timeout range is 1 to 600 seconds.

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# destination 192.168.92.125
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# path-option 1 dynamic
Router(config-if)# retry-timer 300
Router(config-if)# autoroute announce | forwarding-adjacency
Router(config-if)# signalled-bandwidth 100
Router(config)# commit
```

Verification

Verify the configuration of MPLS-TE tunnel using the following command.

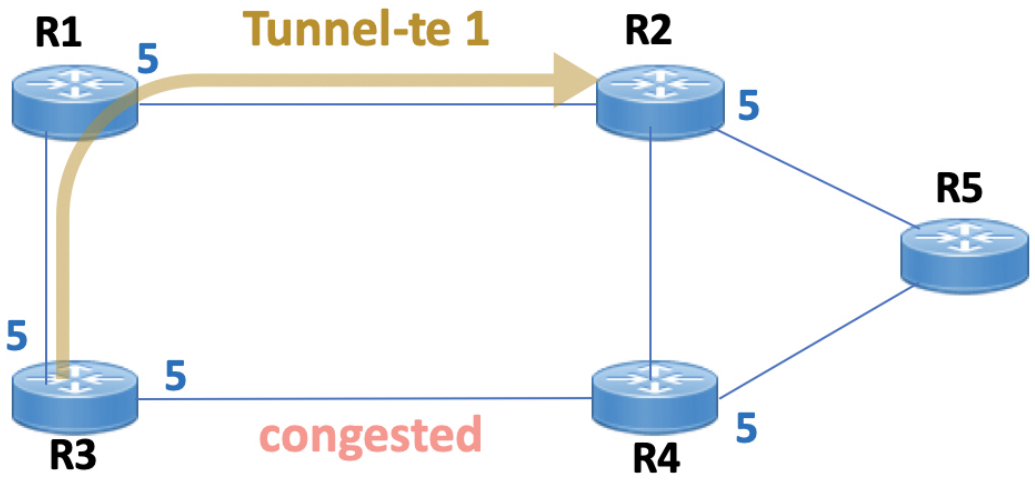
```
Router# show mpls traffic-engineering tunnels brief

            Signalling Summary:
                LSP Tunnels Process:  running
                        RSVP Process:  running
                          Forwarding:  enabled
            Periodic reoptimization:  every 3600 seconds, next in 2538 seconds
             Periodic FRR Promotion:  every 300 seconds, next in 38 seconds
            Auto-bw enabled tunnels:  0 (disabled)
                        TUNNEL NAME         DESTINATION      STATUS  STATE
```

```
                tunnel-te1       192.168.92.125      up       up
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Automatic Modification Of An MPLS-TE Tunnel's Metric

If the IGP calculation on a router results in an equal cost multipath (ECMP) scenario where next-hop interfaces are a mix of MPLS-TE tunnels and physical interfaces, you may want to ensure that a TE tunnel is preferred. Consider this topology:

Figure 2: MPLS-TE Tunnel



1. All links in the network have a metric of 5.
2. To offload a congested link between R3 and R4, an MPLS-TE tunnel is created from R3 to R2.
3. If the metric of the tunnel is also 5, traffic from R3 to R5 is load-balanced between the tunnel and the physical R3-R4 link.

To ensure that the MPLS-TE tunnel is preferred in such scenarios, configure the **autoroute metric** command on the tunnel interface. The modified metric is applied in the routing information base (RIB), and the tunnel is preferred over the physical path of the same metric. Sample configuration:

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# autoroute metric relative -1
```

The **autoroute metric** command syntax is **autoroute metric** {**absolute**|**relative**} *value*

- **absolute** enables the absolute metric mode, for a metric range between 1 and 2147483647.
- **relative** enables the relative metric mode, for a metric range between -10 and 10, including zero.

Note

- Since the **relative** metric is not saved in the IGP database, the advertised metric of the MPLS-TE tunnel remains 5, and doesn't affect SPF calculation outcomes on other nodes.
- Configuring Segment Routing and Autoroute Destination together is not supported. If autoroute functionality is required in an Segment Routing network, we recommend you to configure Autoroute Announce.

Configuring Dark Bandwidth Accounting

To enable RSVP-TE Dark Bandwidth Accounting feature, perform the following steps:

1. Enable per-interface aggregate SR counters.
2. Configure TE dark bandwidth accounting.

SUMMARY STEPS

1. **configure**
2. **hw-module profile cef dark-bw enable**
3. **mpls traffic-eng**
4. **bandwidth-accounting**
5. **application interval** *seconds*
6. **application enforced**
7. **sampling-interval** *seconds*
8. **adjustment-factor** *percentage*
9. **flooding threshold up** *percentage* **down** *percentage*
10. **flooding sr-traffic** *percentage*
11. Use the **commit** or **end** command.
12. **mpls traffic-eng link-management bandwidth-accounting enforce all**
13. **clear mpls traffic-eng link-management bandwidth-accounting**
14. **show interface** *type_path* **accounting**
15. **show mpls traffic-eng link-management summary**
16. **show mpls traffic-eng link-management advertisements**
17. **show mpls traffic-eng link-management interfaces** [*type interface-path-id*] [**detail**] [**bandwidth-accounting**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose

Step 2	**hw-module profile cef dark-bw enable**
	Example:
	`RP/0/RP0/CPU0:router(config)# hw-module profile cef dark-bandwidth enable`
	`RP/0/RP0/CPU0:router(config)# commit`
		Enables per-interface aggregate SR counters for all interfaces on the router.
		Note After you enter this command, you must reload the router.
		Caution This command should only be enabled on a router where a prefix with an SR prefix SID learned via ECMP has the same out label across all its paths. This condition is met for prefixes learned via ECMP in an SR network with homogenous SRGB and when either no protection or IP-FRR LFA protection is enabled.
		Do not use this command on a router with TI-LFA enabled while expecting backup paths that would require extra labels to be imposed.
		In Cisco IOS XR release 7.5.2 and earlier, do not use this command on a router where a prefix with an SR prefix SID is learned via ECMPs with different egress action (pop and swap). Label programming errors and traffic loss would be observed for those prefixes. In Cisco IOS XR release 7.5.3 and later, this restriction no longer applies.

Step 3	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS TE configuration mode.
Step 4	**bandwidth-accounting** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# bandwidth-accounting`	Enables RSVP-TE dark bandwidth accounting and enters bandwidth accounting configuration mode.
Step 5	**application interval** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# application interval 90`	Configures the length of the application interval in seconds. At the end of application interval, dark bandwidth rates are computed and applied to all RSVP-TE enabled interfaces. If the interval is reconfigured while the timer is running, the new value is compared to the time remaining for the running timer. The timer is adjusted so that the lower of these two values is used for this interval. The subsequent interval will use the newly configured value.

	Command or Action	Purpose
		Note TE stores sample history for the current and previous application intervals. If the application interval is lowered, TE may discard the sample history. Range is from 90 to 1800. The default value is 180.
Step 6	**application enforced** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# application enforced`	Enables enforcement of the calculated BMRe rate.
Step 7	**sampling-interval** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# sampling-interval 30`	Configures the length of the sampling interval in seconds. The dark bandwidth rate is collected from the statistics collector process (statsD) at the end of each sampling interval for each TE link. If the interval is reconfigured while the timer is running, the new value is compared to the time remaining for the running timer. The timer is adjusted so that the lower of these two values is used for this interval. The subsequent interval will use the newly configured value. Range is from 10 to 600. The default is 60.
Step 8	**adjustment-factor** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# adjustment-factor 200`	Configures TE to over-book (>100%) or under-book (<100%) the effective maximum reservable bandwidth (BMRe). The measured dark-bandwidth will be scaled based on the adjustment factor. Range is from 0 to 200. The default value is 100.
Step 9	**flooding threshold up** *percentage* **down** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# flooding threshold up 30 down 30`	Configures the reserved bandwidth thresholds. When bandwidth crosses one of these thresholds, flooding is triggered. Range is from 0 to 100. The default value is 10.
Step 10	**flooding sr-traffic** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# flooding sr-traffic 30`	Configures the flooding trigger for bandwidth accounting in segment routing traffic. When the bandwidth crosses the threshold value, flooding is triggered. Range is from 0 to 100. The default value is 10. This support is introduced in Cisco IOS XR Release 7.5.4.
Step 11	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

	Command or Action	Purpose
		• **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 12	**mpls traffic-eng link-management bandwidth-accounting enforce all** **Example:** `RP/0/RP0/CPU0:router# mpls traffic-eng link-management bandwidth-accounting enforce all`	(Optional) Applies the measured rates immediately. When you apply measured rates immediately, the RSVP-TE bandwidth-accounting might flood the updated bandwidth values immediately. Applying measured rates immediately does not affect the periodic application of the bandwidth.
Step 13	**clear mpls traffic-eng link-management bandwidth-accounting** **Example:** `RP/0/RP0/CPU0:router# clear mpls traffic-eng link-management bandwidth-accounting`	(Optional) Erases the collected sample history and resets the application and sample timers.
Step 14	**show interface** *type_path* **accounting** **Example:** `RP/0/RP0/CPU0:router# show interface hundredGigE 0/0/0/26 accounting`	(Optional) Displays the per-interface SR accounting.
Step 15	**show mpls traffic-eng link-management summary** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management summary`	(Optional) Displays a summary of link management information, including bandwidth accounting information.
Step 16	**show mpls traffic-eng link-management advertisements** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management advertisements`	(Optional) Displays local link information that MPLS-TE link management is currently flooding into the global TE topology.
Step 17	**show mpls traffic-eng link-management interfaces** [*type interface-path-id*] [**detail**] [**bandwidth-accounting**] **Example:**	(Optional) Displays bandwidth accounting and utilization details and link management information.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# show mpls traffic-eng link-management interfaces gig0/1/1/1 detail`	

To display the per-interface SR counters, use the **show interface** *type_path* **accounting** command:

```
RP/0/RP0/CPU0:router# show interface hundredGigE 0/0/0/26 accounting
Mon Feb  3 23:29:48.449 UTC
HundredGigE0/0/0/26
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  ARP                         3              222            3              126
  IPV6_ND                    11             1122           13             1112
  CLNS                       99           121910           94           116212
  SR_MPLS                     0                0         3126           581436
```

Note The SR_MPLS counter is an egress-only counter and includes all traffic from the following:

- IPv4 unlabelled - SR last-hop traffic after PHP
- IPv6 unlabelled - SR last-hop traffic after PHP
- SR label switched traffic

To display detailed SR bandwidth utilization, use the **show mpls traffic-eng link-management interface** *type_path* **detail** command:

```
Router# show mpls traffic-eng link-management interface hundredGigE 0/0/0/26 detail
bandwidth-accounting

  System Information::
      Links Count         : 16 (Maximum Links Supported 800)

  Link ID:: HundredGigE0/0/0/26 (26.1.1.1)
    Local Intf ID: 22
    Link Status:

      Link Label Type          : PSC
      Physical BW              : 1000000 kbits/sec
      BCID                     : RDM
      Max Reservable BW        : 529309 kbits/sec (reserved: 94% in, 94% out)
      Flooded Max Reservable BW: 529309 kbits/sec
      BC0 (Res. Global BW)     : 529309 kbits/sec (reserved: 94% in, 94% out)
      BC1 (Res. Sub BW)        : 0 kbits/sec (reserved: 100% in, 100% out)
      MPLS TE Link State       : MPLS TE on, RSVP on, admin-up
      IGP Neighbor Count       : 1
      Max Res BW (RDM)    : 900000 kbits/sec
      BC0 (RDM)           : 900000 kbits/sec
      BC1 (RDM)           : 0 kbits/sec
      Max Res BW (MAM)    : 0 kbits/sec
      BC0 (MAM)           : 0 kbits/sec
      BC1 (MAM)           : 0 kbits/sec

    Bandwidth Accounting: Segment-Routing
    Bandwidth Accounting Enforced: Yes
      Bandwidth Utilization Details:
```

```
      Sampling Interval               : 30 sec
      Application Interval            : 90 sec
      Adjustment Factor               : 200%
      Max Reservable BW Up Threshold  : 30
      Max Reservable BW Down Threshold: 30

  Last Application at:  23:46:32 Mon 03 Feb 2020 (51 seconds ago)
        Segment-Routing BW Utilization      : 185346 kbits/sec
        Adjusted BW Utilization             : 370692 kbits/sec
        Enforced BW Utilization             : 370692 kbits/sec
      Next Application at:  19:42:43 Sun 30 Apr 2017 (in 38 seconds)
      Last Collection at :  19:41:42 Sun 30 Apr 2017 (23 seconds ago)
      Next Collection at :  19:42:11 Sun 30 Apr 2017 (in 6 seconds)

  Bandwidth Samples (Kbps):
      Timestamp                     Segment-Routing
      19:40:12 Sun 30 Apr 2017               187961
      19:40:42 Sun 30 Apr 2017               180130
      19:41:12 Sun 30 Apr 2017               187949
```

To display a summary of link management information, including bandwidth accounting information, use the **show mpls traffic-eng link-management summary** command:

```
Router# show mpls traffic-eng link-management summary

  System Information::
      Links Count         : 14 (Maximum Links Supported 800)
      Flooding System     : enabled
      IGP Areas Count     : 1

  IGP Areas
  ----------

  IGP Area[1]:: IS-IS 0 level 2
      Flooding Protocol   : IS-IS
      Flooding Status     : flooded
      Periodic Flooding   : enabled (every 180 seconds)
      Flooded Links       : 7
      IGP System ID       : 0000.0000.0001
      MPLS TE Router ID   : 10.0.0.1
      IGP Neighbors       : 7

  Bandwidth accounting:
      Sampling interval: 30 seconds, Next in 29 seconds
      Application interval: 90 seconds, Next in 1 seconds
```

To display local link information that MPLS-TE link management is currently flooding into the global TE topology, use the **showmpls traffic-eng link-management advertisements** command:

```
Router# show mpls traffic-eng link-management advertisements

  Flooding Status             : Ready
  Last Flooding               : 470 seconds ago
  Last Flooding Trigger       : Link BW changed
  Next Periodic Flooding In   : 143 seconds
  Diff-Serv TE Mode           : Not enabled
  Configured Areas            : 1

  IGP Area[1]:: IS-IS 0 level 2
      Flooding Protocol   : IS-IS
      IGP System ID       : 0000.0000.0001
      MPLS TE Router ID   : 10.0.0.1
      Flooded Links       : 5
```

```
Link ID:: 0 (GigabitEthernet0/1/1/0)
    Link IP Address      : 10.12.110.1
    O/G Intf ID          : 22
    Neighbor             : ID 0000.0000.0002.00, IP 10.12.110.2
    TE Metric            : 10
    IGP Metric           : 10
    Physical BW          : 1000000 kbits/sec
    BCID                 : RDM
    Max Reservable BW    : 899999 kbits/sec
    Res Global BW        : 899999 kbits/sec
    Res Sub BW           : 0 kbits/sec
```

Configure Autoroute Tunnel as Designated Path

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Configure Autoroute Tunnel as Designated Path	Release 7.6.2	Simplify the path selection for a traffic class and split traffic among multiple TE tunnels to achieve many benefits such as security and service-level agreements. You can now exclusively specify an autoroute tunnel to forward traffic to a particular tunnel destination address without considering the IS-IS metric for traffic path selection. Earlier, MPLS-TE considered either the Forwarding Adjacency (FA) or Autoroute (AA) tunnel to forward traffic based only on IS-IS metric. The feature introduces the **mpls traffic-eng tunnel restricted** command.

MPLS-TE builds a unidirectional tunnel from a source to a destination using label switched path (LSP) to forward traffic.

To forward the traffic through MPLS tunneling, you can use autoroute, forwarding adjacency, or static routing:

- Autoroute (AA) functionality allows to insert the MPLS TE tunnel in the Shortest Path First (SPF) tree for the tunnel to transport all the traffic from the headend to all destinations behind the tail-end. AA is only known to the tunnel headend router.
- Forwarding Adjacency (FA) allows the MPLS-TE tunnel to be advertised as a link in an IGP network with the cost of the link associated with it. Routers outside of the TE domain can see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network.
- Static routing allows you to inject static IP traffic into a tunnel as the output interface for the routing decision.

Prior to this release, by default, MPLS-TE considers FA or AA tunnels to forward traffic based on the IS-IS metric. The lower metric is always used to forward traffic. There was no mechanism to forward traffic to a specific tunnel interface.

For certain prefixes to achieve many benefits such as security and service-level agreements, there might be a need to forward traffic to a specific tunnel interface that has a matching destination address.

With this feature, you can exclusively use AA tunnels to forward traffic to their tunnel destination address irrespective of IS-IS metric. Traffic steering is performed based on the prefixes and not metrics. Traffic to other prefixes defaults to the forwarding-adjacency (FA) tunnels.

To enable this feature, use the **mpls traffic-eng tunnel restricted** command.

Also, you may require more than one AA tunnel to a particular remote PE and use ECMP to forward traffic across AA tunnels. You can configure a loopback interface with one primary address and multiple secondary addresses on the remote PE, using one IP for the FA tunnel destination, and others for the AA tunnels destinations. Multiple IP addresses are advertised in the MPLS TE domain using the typed length value (TLV) 132 in IS-IS. A TLV-encoded data stream contains code related to the record type, the record length of the value, and value. TLV 132 represents the IP addresses of the transmitting interface.

Feature Behavior

When MPLS-TE tunnel restricted is configured, the following is the behavior:

- A complete set of candidate paths is available for selection on a per-prefix basis during RIB update as the first hop computation includes all the AA tunnels terminating on a node up to a limit of 64 and the lowest cost forwarding-adjacency or native paths terminating on the node or inherited from the parent nodes in the first hops set for the node.
- During per-prefix computation, AA tunnel first hops are used for traffic sent to their tunnel destination address even if FA tunnel or native first hops have a better metric. AA tunnel first-hops are not used for any other prefixes.
- ECMP is used when multiple AA tunnel first hops have the same destination address and metric.
- During per-prefix computation, AA tunnel first hops are used for traffic sent to their tunnel destination address, and for all other destinations on the tunnel tail node or behind it, even if a native path has a better metric.

  Adding **mpls traffic-eng tunnel preferred** configuration has no effect when the tunnel restricted is already configured.
- If there's no AA tunnel or if the tunnel is down, then native paths are used for all other destinations on the tunnel tail node or behind it.

  The route metric for a prefix reflects the chosen first-hop, not necessarily the lowest cost SPF distance to the node.

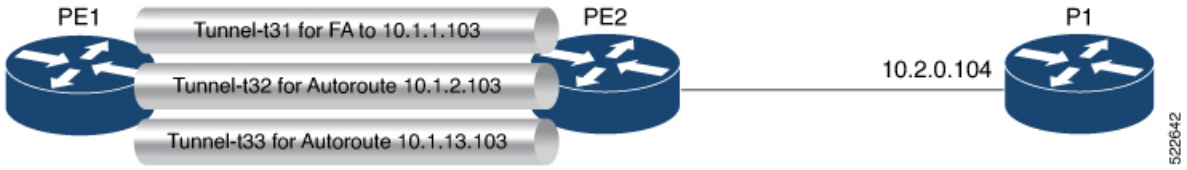
Restrictions for Configure Autoroute Tunnel as Designated Path

- The total number of interface addresses to the number that can be contained in 255 bytes is 63 for IPv4 and 15 for IPv6.
- When this feature is enabled, a maximum of 64 tunnels can terminate on the tail node.

Configure Autoroute Tunnel as Designated Path

Let's understand how to configure the feature using the following topology:

Figure 3: Topology



Consider the topology where PE1 has three MPLS tunnels connecting to PE2.

- Tunnel-t31: Forwarding adjacency (FA) is configured to the primary address of Loopback 0 on PE2 (10.1.1.103).
- Tunnel- t32: Autoroute announce (AA) is configured to a secondary address of Loopback 0 on PE2 (10.1.2.103).
- Tunnel-t33: Autoroute announce (AA) is configured to a secondary address of Loopback 0 on PE2 (10.1.3.103).

This feature is not enabled by default. When this feature is not enabled, traffic is load balanced over all AA tunnels towards the same remote PE provided the tunnel metric is the same:

```
Router# show routes
i L2 10.1.1.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.1.3.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
 10.2.0.104/32 [115/50] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/50] via 10.1.3.103, 00:00:30, tunnel-t33
```

Configuration Example

You can configure the feature using the **mpls traffic-eng tunnel restricted** command.

```
RP/0/RSP0/CPU0:ios# configure
RP/0/RSP0/CPU0:ios(config)# router isis 1
RP/0/RSP0/CPU0:ios(config-isis)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-af# mpls traffic-eng tunnel restricted
```

Running Configuration

The following example shows the AA tunnel metric running configuration:

```
router isis 1
 address-family ipv4 unicast
  mpls traffic-eng tunnel restricted
 !
!
end
```

Verification

When you enable the feature, traffic towards a particular prefix is sent only over the tunnel that has that IP address as destination.

```
Router# show route
i L2 10.1.1.103/32 [115/40] via 10.1.1.103, 00:00:04, tunnel-t31
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:04, tunnel-t32
i L2 10.1.3.103/32 [115/40] via 10.1.3.103, 00:00:04, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.1.103, 00:00:04, tunnel-t31
i L2 10.2.0.104/32 [115/50] via 10.1.1.103, 00:00:04, tunnel-t31
```

When multiple restricted AA tunnels are created towards the same destination IP address, router load balances traffic across all those tunnels:

```
Router# show route
i L2 10.1.1.103/32 [115/40] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/40] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:08, tunnel-t32
                   [115/40] via 10.1.2.103, 00:00:30, tunnel-t34
i L2 10.1.3.103/32 [115/40] via 10.1.3.103, 00:00:08, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/40] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
i L2 10.2.0.104/32 [115/50] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/50] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
```

Configure MPLS over UDP

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
Configure MPLS over UDP	Release 7.5.3	You can now enable MPLS over UDP, where a dynamic tunnel is created with the BGP next hop that helps in traffic distribution along the path. Traffic distribution considers parameters such as IP source address, IP destination address, and UDP source port along different tunnels for hashing. This feature introduces the following new commands: • **set encapsulation-type mpls-udp** • **overlay-encapsulation mpls-udp**

There are different encapsulation methods to segregate traffic along the path, and one such method is MPLS over UDP encapsulation. In other encapsulation methods, such as MPLS, the load-balancing decision on core routers is based on MPLS label values. And for VPN, the top label is typically an IGP label that identifies the target PE. Hence the MPLS label is the same for all flows towards that PE.

MPLS over UDP considers IP source address, IP destination address, and UDP source port for distributing traffic effectively along different paths. This good variety of parameters enhances load-balancing efficiently, which is not available in other encapsulation methods. When you enable this feature, MPLS packets are encapsulated in UDP, and the tunnels are created dynamically with the BGP next hop.

Here is the encapsulation format for MPLS over UDP:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Source Port = Entropy      |       Dest Port = MPLS        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           UDP Length          |        UDP Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      MPLS Label Stack                         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
~                        Message Body                           ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

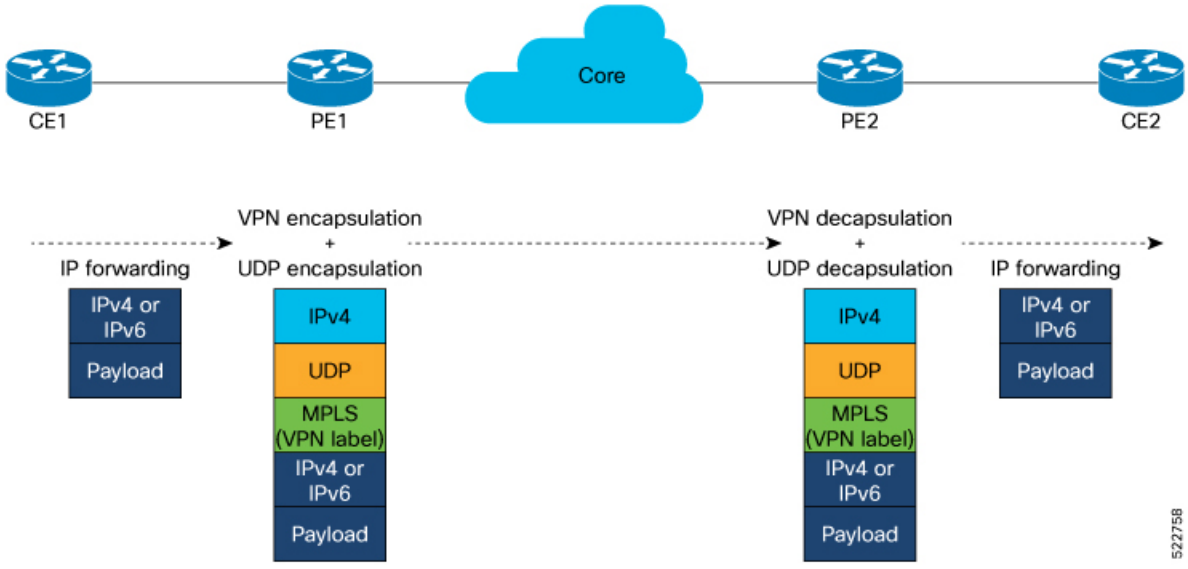
where,

- Source IP address is the tunnel source address.
- Destination IP address is the tunnel destination address, which is the BGP next-hop.
- UDP source port is the entropy value (allowed range is from 49152 through 65535).
- UDP destination port is 6635 used for MPLS payload.

Topology

Let's consider the following topology to understand MPLS over UDP.

Figure 4: MPLS over UDP



In this topology, traffic is sent from CE1 to CE2.

- When PE1 receives traffic from CE1.
- PE1 adds the MPLS label.
- PE1 adds a UDP flow label header. The UDP header contains a source and a destination port.
  - When the next-hop is defined as MPLS over UDP, a dynamic tunnel is created and the traffic is sent through this tunnel to the core router.
  - If next hop is not enabled with MPLS over UDP, regular IPv4 or IPv6 routing lookup is performed.
- The core router forwards the packet to PE2.
- PE2 decapsulates the packet and forwards it to the CE2 router.

You can define a set of decap-prefix lists and filter incoming packet source IP addresses using a set of specified decap-prefix. If the source IP address matches, then packets are terminated for the UDP tunnel. If the source IP address does not match, then packets are discarded without tunnel termination. When the decap-prefix list is not defined, all the incoming MPLS over UDP traffic is allowed regardless of their source IP addresses.

Restriction for Configuring MPLS over UDP

- Supported on routers with line cards based on Q200 Silicon.
- MPLS over UDP is not supported over IPv6 underlay.
- MPLS over UDP is not supported with permit ACEs in ACLs configured on underlay interfaces.

Configure MPLS over UDP

Perform the following tasks to configure MPLS over UDP:

- Configure interface NVE
- Enable MPLS over UDP
- Configure BGP
- Define prefix-set for encapsulation

Configuration Example

```
/* Configure interface NVE and enable MPLS over UDP. */
Router#configure
Router(config)# interface nve1
Router(config-if)# overlay-encapsulation mpls-udp
Router(config-if)# source-interface Loopback0
Router(config-if)# logging events link-status
Router(config-if)# commit
Router(config-if)# exit

/* Configure BGP */
Router(config)#router bgp 100
Router(config-bgp)#nsr
Router(config-bgp)#bgp router-id 10.1.1.1
```

```
Router(config-bgp)#bgp graceful-restart
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths ibgp 16
Router(config-bgp-af)#address-family vpnv4 unicast
Router(config-bgp-af)#nexthop route-policy MPLSoUDP-Encap-1
Router(config-bgp-af)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths ebgp 10
Router(config-bgp-af)#maximum-paths ibgp 16
Router(config-bgp-af)#address-family vpnv6 unicast
Router(config-bgp-af)#nexthop route-policy MPLSoUDP-Encap-1
Router(config-bgp-af)#neighbor 192.0.2.1
Router(config-bgp-nbr)#remote-as 100
Router(config-bgp-nbr)#update-source Loopback0
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#address-family vpnv4 unicast
Router(config-bgp-nbr-af)#address-family vpnv6 unicast
Router(config-bgp-nbr-af)#vrf vrf100
Router(config-bgp-vrf)#rd 100:1
Router(config-bgp-vrf)#address-family ipv4 unicast
Router(config-bgp-vrf-af)#label mode per-vrf
Router(config-bgp-vrf-af)#maximum-paths ebgp 16
Router(config-bgp-vrf-af)#maximum-paths ibgp 16
Router(config-bgp-vrf-af)#redistribute connected
Router(config-bgp-vrf-af)#address-family ipv6 unicast
Router(config-bgp-vrf-af)#label mode per-vrf
Router(config-bgp-vrf-af)#maximum-paths ebgp 16
Router(config-bgp-vrf-af)#maximum-paths ibgp 16
Router(config-bgp-vrf-af)#redistribute connected
Router(config-bgp-vrf-af)#commit
```

```
/* Define prefix-set for encapsulation*/
Router(config)#prefix-set BGP-NH-1
Router(config-pfx)# 192.0.2.1/32 le 32,
Router(config-pfx)# 192.0.3.1/32 le 32,
Router(config-pfx)# 192.0.4.1/32 le 32,
Router(config-pfx)# 192.0.5.1/32 le 32
Router(config-pfx)#exit
Router(config)#commit

Router(config)#route-policy MPLSoUDP-Encap-1
Router(config-rpl)#if next-hop in BGP-NH-1 then
Router(config-rpl-if)#set encapsulation-type mpls-udp
Router(config-rpl-if)#else
Router(config-rpl-else)#pass
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
Router(config)#commit
```

The following configuration example shows how to filter the source IP address for decapsulation. The incoming packet source IP that is part of the object-group is decapsulated:

```
Router#configure
Router(config)#object-group network ipv4 decap_2
Router(config-object-group-ipv4)#192.0.2.1/32
Router(config-object-group-ipv4)#192.0.3.1/32
Router(config-object-group-ipv4)#192.0.4.1/32
Router(config-object-group-ipv4)#192.0.5.1/32
Router(config-object-group-ipv4)#commit
Router(config-object-group-ipv4)#root
Router(config)#nve
```

```
Router(config-nve)#decap-prefix source ipv4 prefix-1
Router(config-nve-decap-src-prefix-ipv4)#object-group decap_2
Router(config-nve-decap-src-prefix-ipv4)#commit
RP/0/RP0/CPU0:PE1(config-nve-decap-src-prefix-ipv4)#root
RP/0/RP0/CPU0:PE1(config)#exit
```

Running Configuration

```
Router#show run

interface nve1
 overlay-encapsulation mpls-udp
 source-interface Loopback0
 logging events link-status
!

prefix-set BGP-NH-1
  192.0.2.1/32 le 32,
  192.0.3.1/32 le 32,
  192.0.4.1/32 le 32,
  192.0.5.1/32 le 32
end-set
!
route-policy MPLSoUDP-Encap-1
  if next-hop in BGP-NH-1 then
    set encapsulation-type mpls-udp
  else
    pass
  endif
end-policy
!
router bgp 100
 nsr
 bgp router-id 10.10.1.1
 bgp graceful-restart
 address-family ipv4 unicast
  maximum-paths ibgp 16
 !
 address-family vpnv4 unicast
  nexthop route-policy MPLSoUDP-Encap-1
 !
 address-family ipv6 unicast
  maximum-paths ebgp 10
  maximum-paths ibgp 16
 !
 address-family vpnv6 unicast
  nexthop route-policy MPLSoUDP-Encap-1
 !
 neighbor 192.0.2.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
 !
 vrf vrf100
  rd 100:1
  address-family ipv4 unicast
   label mode per-vrf
   maximum-paths ebgp 16
```

```
   maximum-paths ibgp 16
   redistribute connected
  !
  address-family ipv6 unicast
   label mode per-vrf
   maximum-paths ebgp 16
   maximum-paths ibgp 16
   redistribute connected
  !
 !
!
mpls traffic-eng
 attribute-set path-option P1
  signalled-bandwidth 10000 class-type 0
  path-selection
   cost-limit 1000
   delay-limit 5000
   exclude AAA
  !
  affinity include YELLOW
  affinity include-strict BLUE GREEN
  affinity exclude RED
  affinity exclude-all
 !
 named-tunnels
  tunnel-te cisco
   path-option 1
    attribute-set P1
   !
  !
 !
!
end
```

Running configuration for source adress filtering for decapsulation:

```
object-group network ipv4 decap_2
 192.0.2.1/32
 192.0.3.1/32
 192.0.4.1/32
 192.0.5.1/32
!
nve
 decap-prefix source ipv4 prefix-1
  object-group decap_2
 !
!
```

Verification

Verify that the MPLS over UDP is configured.

```
Router#show nve interface detail
Interface: nve1 State: Up Encapsulation: mpls-udp
Source Interface: Loopback0 (primary: 10.1.1.1)
Source Interface State: Up
NVE Flags: 0xa1, Admin State: Up, Interface Handle 0xf000014
UDP Port: 6635
Flags decode: ready, IM syn, SOC
LTEP Publish Info:
Last Del: NA
```

Verify that the tunnel type is MPLS UDP.

```
Router#show route vrf vrf100 10.1.3.1 detail
Routing entry for 10.1.3.1/32
Known via "bgp 100", distance 200, metric 0, type internal
Installed Aug 24 06:52:47.539 for 11:49:38
Routing Descriptor Blocks
192.0.2.1, from 192.0.2.1
Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000
Route metric is 0
Label: 0x10 (16)
Tunnel ID: None
Binding Label: None
Extended communities count: 0
Source RD attributes: 0x0000:100:1
NHID:0x0(Ref:0)
IP Tunnel Info: Auto create: 1 Tunnel Type: mpls-udp
MPLS eid:0x1
Route version is 0x1 (1)
No local label
IP Precedence: Not Set
QoS Group ID: Not Set
Flow-tag: Not Set
Fwd-class: Not Set
Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_REMOTE
Download Priority 3, Download Version 334
Route eid: 0x1
No advertising protos.
RP/0/RP0/CPU0:R1#
```

```
Router#show cef tunnel encap

Tunnel-type: mpls-udp, Dest addr: 192.0.2.1, Underlay Table: 0xe0000000
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000

Tunnel-type: mpls-udp, Dest addr: 192.0.3.1, Underlay Table: 0xe0000000
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000

Tunnel-type: mpls-udp, Dest addr: 192.0.4.1, Underlay Table: 0xe0000000
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000
```

The following show commands are used to verify the decapsulation:

```
Router#show cef global tunnel decap-nve

Overlay Table: 0xe0000000, Nve-If-ID: 1
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000

Router#show rib opaques ltep


Summary of LTEP opaque data in IPv4 RIB:

Opaque key: LTEP Overlay Table: 0xe0000000, IF ID: 1
Opaque data: LTEP Type: mpls-udp, NVE IFH: 0xf00001c, Source: 10.1.1.1, Underlay Table:
0xe00
00000, UDP Dest. Port: 6635

Router#show ofa objects iptnldecap object-count location 0/1/CPU0

Table [IPTNLDECAP] has 1 entries in DB
```

Common Label for IPv4 and IPv6 Address Families

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Common MPLS Label for IPv4 and IPv6 Address Families	Release 24.1.1	This release simplifies network operations and reduces the complexity of maintaining separate MPLS labels for IPv4 and IPv6 address families by enabling you to assign a single MPLS label to a Virtual Routing and Forwarding (VRF) instance common to both address families. You can allocate the label statically and dynamically. Previously, you could only allocate one label per address family. The feature introduces these changes: **CLI:** • The **per-vrf-46** keyword is introduced in the **label mode** command. • Modified the output of the following show commands: • **show bgp labels** • **show bgp process** • **show bgp vrf** **YANG Data Models:** `Cisco-IOS-XR-um-router-bgp-cfg.yang` (see GitHub, YANG Data Models Navigator)

You have the flexibility to enable the **per-vrf-46** mode either globally or at the VRF level. Dynamic allocation of **per-vrf-46** MPLS label is supported in global routing table (default VRF) and at the VRF level. Static allocation of **per-vrf-46** MPLS label is supported only at the VRF level. If a static configuration is enabled after a dynamic MPLS label assignment, the static label takes precedence over the dynamic label.

Furthermore, you can allocate the **per-vrf-46** label mode globally under the VPNv4 and VPNv6 unicast address families. The VRFs inherit the configured label mode unless a label mode is explicitly configured under a VRF.

To use the same label for advertising both IPv4 and IPv6 prefixes in a VRF or global routing table, you must configure the **per-vrf-46** label allocation mode explicitly under each address family. Enabling this mode in only one address family limits the allocated VPN label to that address family's advertisements.

Guidelines and Limitations for Using Common Label for IPv4 and IPv6 Address Families

The guidelines and restrictions for using the **per-vrf-46** label modes:

- The allocation of a static label is limited to a VRF. Dynamic allocation is supported for the global routing table and VRF.
- Unconfiguring the **per-vrf-46** label allocation mode from one address family only impacts that address family.
- If another address family has the **per-vrf-46** label allocation mode, the MPLS label remains reserved for use in that address family. When the last address family on a VRF has the **per-vrf-46** label allocation mode unconfigured, the allocated MPLS label gets released.
- In the case of static label allocation discrepancies, you must clear the discrepancy by running the **clear mpls static local-label discrepancy <label-value>** command.

Configure a Common Label for IPv4 and IPv6 Address Families

BGP configurations

Perform the following steps to configure **per-vrf-46** label mode:

- To enable the **per-vrf-46** label mode for global table IPv4 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv4 unicast
Router(configure-bgp-af)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for global table IPv6 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv6 unicast
Router(configure-bgp-af)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for global table VPNv4 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family vpnv4 unicast
Router(configure-bgp-af)#vrf all
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for global table VPNv6 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family vpnv6 unicast
Router(configure-bgp-af)#vrf all
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for VRF IPv4 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv4 unicast
Router(configure-bgp-af)#vrf INET
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for VRF IPv6 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv6 unicast
```

```
Router(configure-bgp-af)#vrf INET
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

MPLS Static Configurations

To allocate a static label to a VRF:

```
Router(configure)#mpls static
Router(config-mpls-static)#vrf Test
Router(config-mpls-static-vrf)#lsp Test-LSP
Router(config-mpls-static-vrf-lsp)#in-label 24100 allocate per-vrf-46
```

Verifications

Verfiy that the **per-vrf-46** label mode is enabled on your router.

- To verify the configurations:
  - RP/0/RP0/CPU0:tb13-r1#show bgp vpnv4 unicast process | i abel
    VRF all label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1#show bgp vpnv6 unicast process | i abel
    VRF all label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1#show bgp vrf INET ipv4 unicast process | i abel
    Label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1#show bgp vrf INET ipv6 unicast process | i abel
    Label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1# show bgp vrf INET ipv4 unicast labels | i 192.0.2.0/24
    *>192.0.2.0/24 198.51.100.0 24200 nolabel
  - RP/0/RP0/CPU0:tb13-r3#show bgp vrf INET ipv4 unicast labels | i 192.0.2.0/24
    *> 192.0.2.0/24 0.0.0.0 nolabel 24200
  - RP/0/RP0/CPU0:tb13-r1#show bgp vrf INET ipv6 unicast labels | i 2001:DB8::/32
    *>2001:DB8::/32 198.51.100.0 24200 nolabel
  - RP/0/RP0/CPU0:tb13-r3#show bgp vrf INET ipv6 unicast labels | i 2001:DB8::/32
    *> 2001:DB8::/32 :: nolabel 24200

Policy-Based Tunnel Selection

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Enhanced Policy-Based Tunnel Selection to Support Recursive Prefixes	Release 7.5.4	We now support forwarding VPN traffic using policy based tunnel selection (PBTS) and ensure that traffic like L3VPN, IPv6 Provider Edge or IPv6 VPN Provider Edge (6PE or 6VPE) services flow as intended. This is possible because we have enhanced PBTS to support recursive prefix traffic.
Policy-Based Tunnel Selection	Release 7.5.3	You can now set the traffic preference based on Policy-Based Tunnel Selection (PBTS) to steer traffic to specific MPLS tunnels to optimize service distribution. It also aids in bandwidth utilization by steering traffic towards the same destination through different paths, independent of IGP metric. PBTS lets you direct traffic into specific TE tunnels based on class-based forwarding of IP (DSCP marking) or MPLS (EXP marking) packets to carry voice and data traffic through the MPLS network. This feature is supported only on routers with line cards based on Q200 Silicon. This feature introduces the following commands: • **forward-class** • **set forward-class** • **hw-module profile cef cbf forward-class-list** • **cef pbts class**

In a network, there are various traffic types such as data, voice, and video. All these traffic types are sent with equal importance to reach a destination through the MPLS tunnel. But different types of traffic needs different priority.

You can classify and prioritize traffic types using Quality of Service. In a packet, there are fields that show the traffic types and classification is performed based on the different fields of a packet. For example:

- IP Precedence value
- Differentiated Services Code Point (DSCP) values
- Source and destination IP address

After the traffic types are classified, QoS marking is performed at different levels to prioritize traffic. For IP traffic, IP precedence or DSCP values are used to mark the packets. For MPLS traffic, EXP values are used to mark these headers for packet marking.

For more Information, see Chapter *Mark Packets to Change Priority Settings* in *Modular QoS Configuration Guide for Cisco 8000 Series Routers*.

You can now use DSCP or EXP to classify and steer traffic to a specific MPLS tunnels using Policy-Based Tunnel Selection (PBTS).

PBTS selects the tunnel based on the classification criteria of the incoming packets, which are based on the experimental (EXP), differentiated services code point (DSCP), or type of service (ToS) field in the packet and forwards traffic to TE tunnels associated with that class-map.

A class-map is defined for various types of packet, and associates this class-map with a forward-class. The class-map defines the matching criteria for classifying a particular type of traffic, while the forward-class defines the forwarding path these packets should take. After a class-map is associated with a forwarding-class in the policy-map, all the packets that match the class-map are forwarded as defined in the policy-map. The egress TE tunnel interfaces route traffic based on the forwarding-class. For each forwarding-class, specify the TE interface explicitly or implicitly in the case of default value with the forward group.

Default-class for paths is always zero (0). If there's no MPLS TE tunnel for a given forward-class, then the default-class (0) is applied. If there's no default-class, then the packet is dropped.

Table 12: Packet Classification

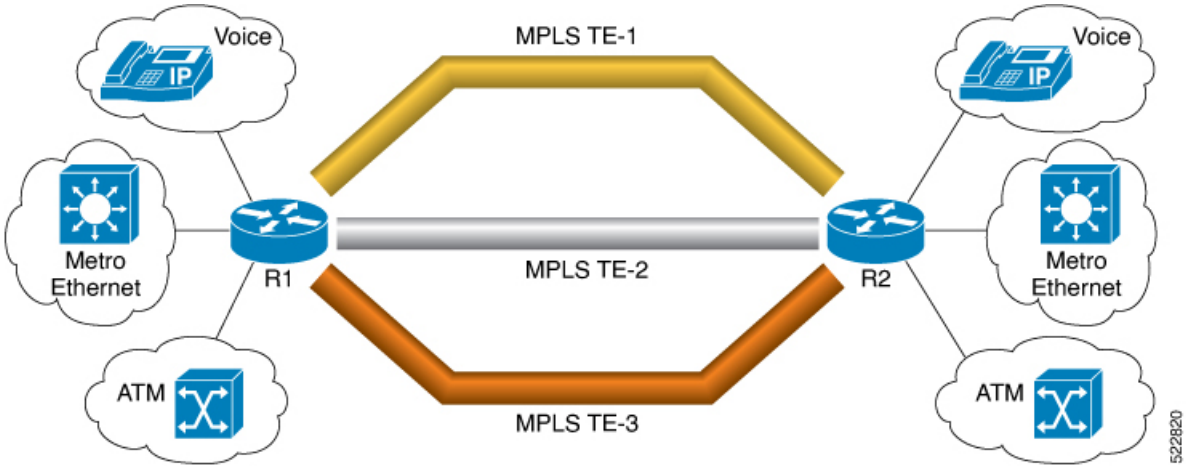
Packet Classification	
EXP	The MPLS experimental bits (EXP) field is a 3-bit field in the MPLS header that defines the QoS treatment on a node for a packet to classify and mark network traffic You can set the EXP value from 0 through 7 on a single MPLS TE tunnel. Value 0 is best effort forwarding and 7 is for highest priority. Router identifies the EXP field values in the topmost MPLS label as match criteria for a class map. By choosing different values for the MPLS EXP field, you can mark packets so that packets have the priority that they require during congestion.

Packet Classification	
DSCP	DiffServ Code Point (DSCP) is a 6-bit field that defines a class and drop precedence. Routers within your network can then use the newly marked DSCP precedence values to determine how the traffic should be treated. You can set the DSCP values for packets entering the network. You can set DSCP values from 0 through 63. Zero is used for best effort forwarding, 1 is used for medium effort forwarding, and 2 for high priority forwarding. The rest of the values are for specific applications such as VOIP calling forwarding, video conference forwarding, and so on.

Topology

Let's understand how the traffic types are classified and steered into a specific tunnel using the EXP marking for MPLS traffic in the following topology:

Figure 5: Policy-Based Tunnel Selection



In this topology, there are three MPLS-TE tunnels: TE1, TE2, and TE3.

These tunnels are associated with a specific forward-class and marked with different EXP marking to forward traffic.

The tunnels forward the traffic based on the forward-class and its associated EXP marking.

- MPLS TE1 is indicated in Gold color that carries voice traffic associated with forward-class 1 and EXP value 1.
- MPLS TE2 is indicated in Silver color that carries Metro Ethernet and ATM traffic associated with forward-class 2 and EXP value 2.
- MPLS TE3 is indicated in Bronze color that carries all other traffic associated with forward-class 0, which is a default class to take all other EXP traffic.

When PBTS is configured on R1and R2, traffic associated with EXP 1 is sent through the MPLS TE1. Similarly, traffic associated with EXP 2 is sent through the MPLS TE2.

Traffic associated other than EXP 1 and 2 is sent through the Bronze MPLS TE3.

Similarly, IP traffic is classified and prioritized using DSCP value.

Note When the TE tunnel associated with the forward-class goes down, you can use the **cef pbts class** command to redirect traffic to another forward-class.

When **cef pbts class any fallback-to any** is specified, then the lowest available forward-class in ascending order of 0-7 is chosen as fallback when paths to any class go down.

Guidelines and Limitations

- Only one forward-class can be associated with a TE tunnel at any time.
- Eight unique forwarding-class values are supported.

This feature supports the following prefixes:

- IPv4 and IPv6 prefixes for LDP over TE
- IPv4 and IPv6 prefixes for RSVP-TE
- Statically configured IPv4 or IPv6 prefix to LDP over TE tunnel destination
- Statically configured IPv4 or IPv6 prefix to TE tunnel destination

Restrictions

- This feature is supported only on routers with line cards based on Q200 Silicon.
- Statically configured MPLS prefix over LDP over TE tunnel is not supported.
- Statically configured MPLS prefix over TE tunnel is not supported.
- Recursive prefixes are not supported for LDP over TE tunnels that are enabled with PBTS.
- Recursive prefixes are not supported for TE tunnels that are enabled with PBTS.

All prefixes advertised by BGP are recursive, that is there's no directly associated next-hop interface. Router performs a lookup in the routing table to find the next-hop interface.

Starting from Cisco IOS XR Release 7.5.4, the Policy-Based Tunnel Selection feature supports the recursive prefixes for the following services:

- L3VPN, 6VPE, 6PE services and overlays over recursive prefixes through the TE tunnel.
- BGP prefixes over TE tunnel.
- Statically configured MPLS prefix over LDP through TE tunnel.
- Statically configured MPLS prefix over TE tunnel.
- Recursive prefixes for LDP over TE tunnels that are enabled with PBTS.

- Recursive prefixes for TE tunnels that are enabled with PBTS.

Configure Policy-Based Tunnel Selection

Perform the following tasks to configure PBTS on the edge routers:

- Classify traffic using IP DSCP or EXP marking
- Associate forward-class
- Apply service-policy to ingress interface
- (Optional) Configure forward-class PBTS using Hardware profile
- Create TE-tunnels to carry traffic based on priority
- Associate egress TE tunnels to forward-class

Configuration Example

The following example shows how to configure PBTS with EXP marking:

```
/* Classify traffic using EXP marking */
Router#configure
Router(config)#class-map match-any exp-1
Router(config-cmap)#match mpls experimental topmost 1
Router(config-cmap)#end-class-map

Router(config)#class-map match-any exp-2
Router(config-cmap)#match mpls experimental topmost 2
Router(config-cmap)#end-class-map

Router(config)#class-map match-any exp-3
Router(config-cmap)#match mpls experimental topmost 3
Router(config-cmap)#end-class-map
Router(config)#commit

/* Associate forward-class */
Router(config)#policy-map INGRESS-POLICY
Router(config-pmap)#class exp-1
Router(config-pmap-c)#set forward-class 1
Router(config-pmap-c)#exit

Router(config-pmap)#class exp-2
Router(config-pmap-c)#set forward-class 2
Router(config-pmap-c)#exit

Router(config-pmap)#class exp-3
Router(config-pmap-c)#set forward-class 3
Router(config-pmap-c)#exit
Router(config-pmap)#exit

/* Apply service-policy to ingress interface */
Router(config)#interface gigabitEthernet 0/0/0/1
Router(config-if)#service-policy input INGRESS-POLICY
Router(config-if)#ipv4 address 10.1.1.1 255.255.255.0
Router(config-if)#exit

/* (Optional) Configure forward-class PBTS using Hardware profile */
Router(config)#hw-module profile cef cbf forward-class-list 1 2 3
```

```
/* Create TE-tunnels to carry traffic based on priority */
Router(config)#interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)#signalled-bandwidth 1000
Router(config-if)#autoroute announce
Router(config-if)#destination 10.50.50.1
Router(config-if)#record-route

/* Associate egress TE tunnels to forward-class */
Router(config-if)#forward-class 1
Router(config-if)#path-option 1 dynamic
Router(config-if)#exit
```

The following example shows how to configure PBTS with DSCP marking:

```
Router#configure
Router(config)#class-map match-any AF41-Class
/* Classify traffic using IP DSCP or EXP marking*/
Router(config-cmap)#match dscp AF41
Router(config-cmap)#exit

/* Associate forward-class */
Router(config)#policy-map INGRESS-POLICY
Router(config-pmap)#class AF41-Class
Router(config-pmap-c)#set forward-class 1
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)#commit

/* Apply service-policy to ingress interface */
Router(config)#interface GigabitEthernet0/0/0/1
Router(config-if)#service-policy input INGRESS-POLICY
Router(config-if)#exit

/* (Optional) Configure forward-class PBTS using Hardware profile */
Router(config)#hw-module profile cef cbf forward-class-list 1
Router(config)#reload

/* Create TE-tunnels to carry traffic based on priority */
Router(config)#interface tunnel-te1
Router(config-if)#ipv4 unnumbered Loopback0
Router(config-if)#signalled-bandwidth 1000
Router(config-if)#autoroute announce
Router(config-if-tunte-aa)#destination 10.50.50.1
Router(config-if)#record-route

/* Associate egress TE tunnels to forward-class */
Router(config-if)#forward-class 1
Router(config-if)#path-option 1 dynamic
Router(config-if)#commit
Router(config)#exit
```

Running Configuration

This example shows the running show configuration for the EXP markings:

```
class-map match-any exp-1
 match mpls experimental topmost 1
 end-class-map
!
```

```
class-map match-any exp-2
 match mpls experimental topmost 2
 end-class-map
!
class-map match-any exp-3
 match mpls experimental topmost 3
 end-class-map
!
class-map match-any AF41-Class
 match dscp af41
 end-class-map
!
policy-map INGRESS-POLICY

 class exp-1
  set forward-class 1
 !
 class exp-2
  set forward-class 2
 !
 class exp-3
  set forward-class 3
 !
 class class-default
 !
 end-policy-map
!
interface tunnel-te1
 ipv4 unnumbered Loopback0
 signalled-bandwidth 1000
 autoroute announce
 !
 destination 10.50.50.1
 record-route
 forward-class 1
 path-option 1 dynamic
!
interface preconfigure GigabitEthernet0/0/0/1
 service-policy input INGRESS-POLICY
 ipv4 address 10.1.1.1 255.255.255.0
!
hw-module profile cef cbf forward-class-list 1 2 3
end
```

This example shows the running show configuration for the DSCP markings:

```
class-map match-any AF41-Class
 match dscp af41
 end-class-map
!
policy-map INGRESS-POLICY
 class AF41-Class
  set forward-class 1
 !
 class class-default
 !
 end-policy-map
!
interface tunnel-te1
 ipv4 unnumbered Loopback0
 signalled-bandwidth 1000
 autoroute announce
 !
 destination 10.50.50.1
```

```
!

interface preconfigure GigabitEthernet0/0/0/1
 service-policy input INGRESS-POLICY
!
hw-module profile cef cbf forward-class-list 1 2 3
end
```

Verification

Verify the PBTS path based on IP information:

```
Router#show cef 10.1.1.1/24

10.1.1.1/24, version 2236, internal 0x1000001 0x30 (ptr 0x940b61b0) [1], 0x600 (0x931456e8),
 0xa20 (0xa98eb318)
 Updated Oct 11 00:01:09.999
 Prefix Len 16, traffic index 0, precedence n/a, priority 3, encap-id 0x1001200000001
  gateway array (0x92fa97f0) reference count 6, flags 0x68, source lsd (5), 1 backups
                [3 type 5 flags 0x208401 (0x94c71960) ext 0x0 (0x0)]
  LW-LDI[type=5, refc=3, ptr=0x931456e8, sh-ldi=0x94c71960]
  gateway array update type-time 1 Oct 11 00:01:09.982
 LDI Update time Oct 11 00:01:09.992
 LW-LDI-TS Oct 11 00:01:09.999
   via 202.158.0.2/32, tunnel-te1, 7 dependencies, weight 0, forward class 1 [flags 0x0]
    path-idx 0 NHID 0x0 [0xa933e300 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, named_4, 5 dependencies, weight 0, forward class 4 [flags 0x0]
    path-idx 1 NHID 0x0 [0xa933e9f0 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, named_5, 7 dependencies, weight 0, forward class 5 [flags 0x0]
    path-idx 2 NHID 0x0 [0xa933ec40 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, named_6, 7 dependencies, weight 0, forward class 6 [flags 0x0]
    path-idx 3 NHID 0x0 [0xa933ee90 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, named_7, 7 dependencies, weight 0, forward class 7 [flags 0x0]
    path-idx 4 NHID 0x0 [0xa933f0e0 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, tunnel-te0, 5 dependencies, weight 0, forward class 0 [flags 0x0]
    path-idx 5 NHID 0x0 [0xa933e0b0 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, tunnel-te2, 7 dependencies, weight 0, forward class 2 [flags 0x0]
    path-idx 6 NHID 0x0 [0xa933e550 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, tunnel-te3, 7 dependencies, weight 0, forward class 3 [flags 0x0]
    path-idx 7 NHID 0x0 [0xa933e7a0 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
```

```
Weight distribution:
slot 0, weight 0, normalized_weight 1, forward class 0
slot 1, weight 0, normalized_weight 1, forward class 1
slot 2, weight 0, normalized_weight 1, forward class 2
slot 3, weight 0, normalized_weight 1, forward class 3
slot 4, weight 0, normalized_weight 1, forward class 4
slot 5, weight 0, normalized_weight 1, forward class 5
slot 6, weight 0, normalized_weight 1, forward class 6
slot 7, weight 0, normalized_weight 1, forward class 7

PBTS class information:
  forward class 0: 1 paths, offset 0
  forward class 1: 1 paths, offset 1
  forward class 2: 1 paths, offset 2
  forward class 3: 1 paths, offset 3
  forward class 4: 1 paths, offset 4
  forward class 5: 1 paths, offset 5
  forward class 6: 1 paths, offset 6
  forward class 7: 1 paths, offset 7
Load distribution: 0 1 2 3 4 5 6 7 (refcount 3)

Hash  OK  Interface                 Address
0     Y   tunnel-te0                point2point
1     Y   tunnel-te1                point2point
2     Y   tunnel-te2                point2point
3     Y   tunnel-te3                point2point
4     Y   named_4                   point2point
5     Y   named_5                   point2point
6     Y   named_6                   point2point
7     Y   named_7                   point2point
```

Verify the PBTS path based on MPLS LDP info:

```
Router#show mpls ldp forwarding 10.1.1.1/24


Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup
  E = Entropy label capability

Prefix          Label   Label(s)       Outgoing     Next Hop            Flags
                In      Out            Interface                        G S R E
--------------- ------- -------------- ------------ ------------------- -------
10.1.1.1/24     24011ImpNull        tt1          202.158.0.2
                          ImpNull        named_4      202.158.0.2
                          ImpNull        named_5      202.158.0.2
                          ImpNull        named_6      202.158.0.2
                          ImpNull        named_7      202.158.0.2
                          ImpNull        tt0          202.158.0.2
                          ImpNull        tt2          202.158.0.2
                          ImpNull        tt3          202.158.0.2
```

To view the PBTS information at CEF level for a label:

```
Router#show cef mpls local-label 24011 eoS detail

Label/EOS 24011/1, Label-type LDP, version 2236, internal 0x1000001 0x30 (ptr 0x94d4ae60)
[1], 0x600 (0x931456e8), 0xa20 (0xa98eb318)
 Updated Oct 11 00:01:09.996
 Prefix Len 21, traffic index 0, precedence n/a, priority 3, encap-id 0x1001200000001
  gateway array (0x92fa97f0) reference count 6, flags 0x68, source lsd (5), 0 backups
```

```
              [3 type 5 flags 0x208401 (0x94c71960) ext 0x0 (0x0)]
 LW-LDI[type=5, refc=3, ptr=0x931456e8, sh-ldi=0x94c71960]
 gateway array update type-time 1 Oct 11 00:01:09.979
LDI Update time Oct 11 00:01:09.989
LW-LDI-TS Oct 11 00:01:09.996
  via 202.158.0.2/32, tunnel-te1, 7 dependencies, weight 0, forward class 1 [flags 0x0]
   path-idx 0 NHID 0x0 [0xa933e300 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_4, 5 dependencies, weight 0, forward class 4 [flags 0x0]
   path-idx 1 NHID 0x0 [0xa933e9f0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_5, 7 dependencies, weight 0, forward class 5 [flags 0x0]
   path-idx 2 NHID 0x0 [0xa933ec40 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_6, 7 dependencies, weight 0, forward class 6 [flags 0x0]
   path-idx 3 NHID 0x0 [0xa933ee90 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_7, 7 dependencies, weight 0, forward class 7 [flags 0x0]
   path-idx 4 NHID 0x0 [0xa933f0e0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, tunnel-te0, 5 dependencies, weight 0, forward class 0 [flags 0x0]
   path-idx 5 NHID 0x0 [0xa933e0b0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, tunnel-te2, 7 dependencies, weight 0, forward class 2 [flags 0x0]
   path-idx 6 NHID 0x0 [0xa933e550 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, tunnel-te3, 7 dependencies, weight 0, forward class 3 [flags 0x0]
   path-idx 7 NHID 0x0 [0xa933e7a0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}

   Weight distribution:
   slot 0, weight 0, normalized_weight 1, forward class 0
   slot 1, weight 0, normalized_weight 1, forward class 1
   slot 2, weight 0, normalized_weight 1, forward class 2
   slot 3, weight 0, normalized_weight 1, forward class 3
   slot 4, weight 0, normalized_weight 1, forward class 4
   slot 5, weight 0, normalized_weight 1, forward class 5
   slot 6, weight 0, normalized_weight 1, forward class 6
   slot 7, weight 0, normalized_weight 1, forward class 7

   PBTS class information:
     forward class 0: 1 paths, offset 0
     forward class 1: 1 paths, offset 1
     forward class 2: 1 paths, offset 2
     forward class 3: 1 paths, offset 3
     forward class 4: 1 paths, offset 4
     forward class 5: 1 paths, offset 5
     forward class 6: 1 paths, offset 6
```

```
  forward class 7: 1 paths, offset 7
Load distribution: 0 1 2 3 4 5 6 7 (refcount 3)

Hash  OK  Interface                 Address
0     Y   tunnel-te0                point2point
1     Y   tunnel-te1                point2point
2     Y   tunnel-te2                point2point
3     Y   tunnel-te3                point2point
4     Y   named_4                   point2point
5     Y   named_5                   point2point
6     Y   named_6                   point2point
7     Y   named_7                   point2point
```

Verify if PBTS policy-map is in use:

```
Router#show qos interface tenGigE 0/0/0/0/0 input

NOTE:- Configured values are displayed within parentheses
Interface TenGigE0/0/0/0/0 ifh 0xf000678  -- input policy
NPU Id:                        0
Total number of classes:       2
Interface Bandwidth:           10000000 kbps
Policy Name:                INGRESS-POLICY
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   prec_7
Policer not configured for this class

Level1 Class                             =   class-default
Policer not configured for this class
```

Verify whether the traffic is taking the correct tunnel-te using the **show interfaces accounting** command. The value of the `pkts out` field increases if traffic is forwarding through the specified tunnel.

```
Router#show interfaces tunnel-te named_4 accounting
named_4
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  MPLS                        0                0        87496          9624560
```

MPLS-TE Features - Details

MPLS TE Fast Reroute Link and Node Protection

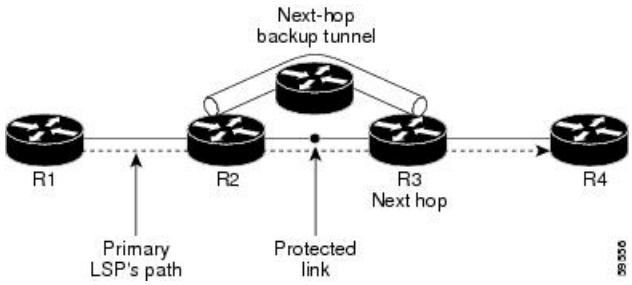
Fast Reroute (FRR) is a mechanism for protecting MPLS TE LSPs from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers try to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or node.

Note If FRR is greater than 50ms, it might lead to a loss of traffic.

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

The following figure illustrates link protection.

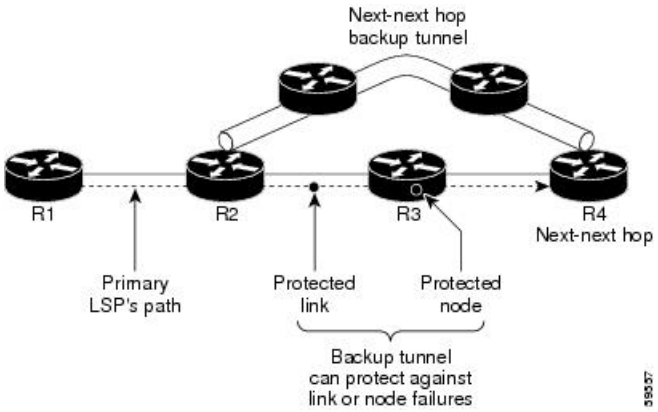
Figure 6: Link Protection



FRR provides node protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called next-next-hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSP paths, bypassing the next-hop node. They protect LSPs if a node along their path fails by enabling the node upstream of the failure to reroute the LSPs and their traffic around the failed node to the next-next hop. NNHOP backup tunnels also provide protection from link failures, because they bypass the failed link and the node.

The following figure illustrates node protection.

Figure 7: Node Protection



Differentiated Services Traffic Engineering

MPLS Differentiated Services 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), you can configure multiple bandwidth constraints. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

Cisco IOS XR software supports two DS-TE modes: pre-standard and IETF. The pre-standard DS-TE mode uses the Cisco proprietary mechanisms for RSVP signaling and IGP advertisements. This DS-TE mode does not interoperate with third-party vendor equipment. Pre-standard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces. Pre-standard DS-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 Pre-standard DS-TE mode.

IETF DS-TE mode uses IETF-defined extensions for RSVP and IGP. This mode inter-operates with third-party vendor equipment. IETF mode supports multiple bandwidth constraint models, including RDM and Maximum Allocation Bandwidth Constraint Model (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.

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.

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 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.

MPLS-TE Forwarding Adjacency

MPLS TE forwarding adjacency allows you to handle a TE label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network that is based on the Shortest Path First (SPF) algorithm. Both Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) are supported as the IGP. A forwarding adjacency can be created between routers regardless of their location in the network. The routers can be located multiple hops from each other.

As a result, a TE tunnel is advertised as a link in an IGP network with the tunnel's cost associated with it. Routers outside of the TE domain see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network. TE tunnel interfaces are advertised in the IGP network just like any other links. Routers can then use these advertisements in their IGPs to compute the SPF even if they are not the headend of any TE tunnels.

Automatic Bandwidth

Automatic bandwidth allows you to dynamically adjust bandwidth reservation based on measured traffic. MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every headend router. MPLS-TE automatic bandwidth monitors the traffic rate on a tunnel interface and 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 on either the largest average output rate that was noticed during a certain interval, or a configured maximum bandwidth value.

While re-optimizing the LSP with the new bandwidth, a new path request is generated. If the new bandwidth is not available, the last good LSP remains 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.

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 re-optimized. Then, the bandwidth samples are cleared to record the new largest output rate at the next interval. If a tunnel is shut down, and is later brought again, the adjusted bandwidth is lost, and the tunnel is brought back with the initially configured bandwidth. When the tunnel is brought back, the application period is reset.

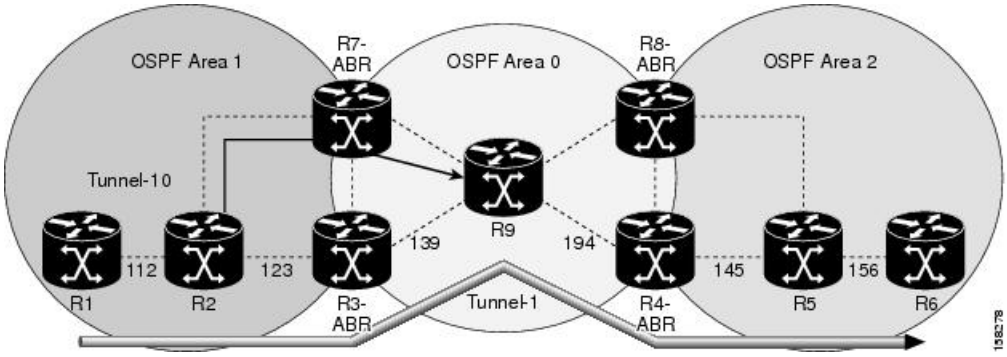
MPLS Traffic Engineering Interarea Tunneling

The MPLS-TE interarea tunneling feature allows you to establish TE tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thus 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. Customers running multiple IGP area backbones (primarily for scalability reasons) requires Multiarea and Interarea TE . 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.

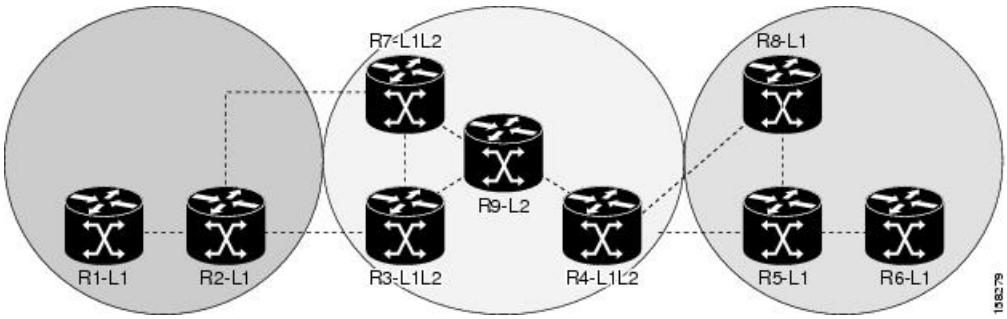
The following figure shows a typical interarea TE network using OSPF.

Figure 8: Interarea (OSPF) TE Network Diagram



The following figure shows a typical interlevel (IS-IS) TE Network.

Figure 9: Interlevel (IS-IS) TE Network Diagram



As shown in the topology, 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).

Loose hop optimization allows the re-optimization 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. Then it is 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 router). The same operation is performed by the last ABR connected to the tailend area to reach the tailend LSR.

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

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

Configuring Performance Measurement

Network performance metrics such as packet loss, delay, delay variation, and bandwidth utilization is a critical measure for traffic engineering (TE) in service provider networks. These network performance metrics provide network operators information about the performance characteristics of their networks for performance evaluation and helps to ensure compliance with service level agreements. The service-level agreements (SLAs) of service providers depend on the ability to measure and monitor these network performance metrics. Network operators can use performance measurement (PM) feature to monitor the network metrics for links as well as end-to-end TE label switched paths (LSPs).

Path Calculation Metric Type

To configure the metric type to be used for path calculation for a given tunnel, use the **path-selection metric** command in either the MPLS-TE configuration mode or under the tunnel interface configuration mode.

The metric type specified per interface takes the highest priority, followed by the MPLS-TE global metric type.

Note If the delay metric is configured, CSPF finds a path with optimized*minimum* link delay metric. See the *Configuring Performance Measurement* chapter in the Segment Routing Configuration Guide for information on configuring interface performance delay measurement.

Configuration Example

The following example shows how to set the path-selection metric to use the IGP metric under a specific tunnel interface:

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# path-selection metric igp
Router(config-if)# commit
```

The following example shows how to set the path-selection metric to use the delay metric under the MPLS-TE configuration mode:

```
Router# configure
Router(config)# mpls traffic-eng
```

```
Router(config-mpls-te)# path-selection metric delay
Router(config-mpls-te)# commit
```

Path-Selection Delay Limit

Apply the **path-selection delay-limit** configuration to set the upper limit on the path aggregate delay when computing paths for MPLS-TE LSPs. After you configure the **path-selection delay-limit** value, if the sum of minimum-delay metric from all links that are traversed by the path exceeds the specified delay-limit, CSPF will not return any path. The periodic path verification checks if the delay-limit is crossed.

The **path-selection delay-limit** value can be configured at the global MPLS-TE, per-interface tunnel, and per path-option attribute set. The path-selection delay-limit per path-option attribute set takes the highest priority, followed by per-interface, and then the MPLS-TE global path-selection delay-limit values.

The delay limit range is a value from 1 to 4294967295 microseconds.

Note See the *Configuring Performance Measurement* chapter in the Segment Routing Configuration Guide for information on configuring interface performance delay measurement.

Configuration Example

The following example shows how to set the path-selection delay limit under a specific tunnel interface:

```
Router# configure
Router(config)# interface tunnel-te2000
Router(config-if)# path-selection metric delay
Router(config-if)# path-selection delay-limit 200
Router(config-if)# commit
```

The following example shows how to set the path-selection delay limit under a path-option attribute set:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# attribute-set path-option test
Router(config-te-attribute-set)# path-selection delay-limit 300
Router(config-te-attribute-set)# root
Router(config)# interface tunnel-te1000
Router(config-if)# path-option 10 dynamic attribute-set test
Router(config-if)# commit
```

The following example shows how to set the path-selection delay limit under the global MPLS-TE configuration mode:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# path-selection metric delay
Router(config-mpls-te)# path-selection delay-limit 150
Router(config-mpls-te)# commit
```

Additional References

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

Related Documents

Related Topic	Document Title
MPLS-TE commands	*MPLS Traffic Engineering Commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.

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

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport