For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the . RSS feeds are a free service.

© 2018 Cisco Systems, Inc. All rights reserved.
Configuring Automatic Bandwidth 49
Configuring Auto-Tunnel Mesh 50
Configuring an MPLS Traffic Engineering Interarea Tunneling 52
Configure Policy-Based Tunnel Selection 52
Configuring LDP over MPLS-TE 54
Configuring MPLS-TE Path Protection 55
MPLS-TE Features - Details 57
Policy-Based Tunnel Selection 60

CHAPTER 5 Implementing MPLS OAM 63
MPLS LSP Ping 63
MPLS LSP Traceroute 65
Implementing MPLS Label Distribution Protocol

MPLS (Multi Protocol Label Switching) is a forwarding mechanism based on label switching. In an MPLS network, data packets are assigned labels and packet-forwarding decisions are taken based on the contents of the label. To switch labeled packets across the MPLS network, predetermined paths are established for various source-destination pairs. These predetermined paths are known as Label Switched Paths (LSPs). To establish LSPs, MPLS signaling protocols are used. Label Distribution Protocol (LDP) is an MPLS signaling protocol used for establishing LSPs. This module provides information about how to configure MPLS LDP.

• Prerequisites for Implementing MPLS Label Distribution Protocol, on page 1
• Restrictions for MPLS LDP, on page 2
• Overview of Label Distribution Protocol, on page 2
• Configuring Label Distribution Protocol, on page 2
• MPLS Label Distribution Protocol : Details, on page 9

Prerequisites for Implementing MPLS Label Distribution Protocol

The following are the prerequisites to implement MPLS LDP:

• You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

• You must be running Cisco IOS XR software.

• You must install a composite mini-image and the MPLS package.

• You must activate IGP.

• We recommend to use a lower session holdtime bandwidth such as neighbors so that a session down occurs before an adjacency-down on a neighbor. Therefore, the following default values for the hello times are listed:

  • Holdtime is 15 seconds.

  • Interval is 5 seconds.

For example, the LDP session holdtime can be configured as 30 seconds by using the holdtime command.
Restrictions for MPLS LDP

- LDP statistics is not displayed in `show mpls forwarding` command output.

Overview of Label Distribution Protocol

In IP forwarding, when a packet arrives at a router the router looks at the destination address in the IP header, performs a route lookup, and then forwards the packet to the next hop. MPLS is a forwarding mechanism in which packets are forwarded based on labels. Label Distribution Protocols assign, distribute, and install the labels in an MPLS environment. It is the set of procedures and messages by which Label Switched Routers (LSRs) establish LSPs through a network by mapping network-layer routing information directly to data-link layer switched paths. These LSPs may have an endpoint at a directly attached neighbor (comparable to IP hop-by-hop forwarding), or may have an endpoint at a network egress node, enabling switching via all intermediary nodes.

LSPs can be created statically, by RSVP traffic engineering (TE), or by LDP. LSPs created by LDP perform hop-by-hop path setup instead of an end-to-end path. LDP enables LSRs to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information. Once label bindings are learned, the LDP is ready to setup the MPLS forwarding plane.

For more information about setting up LSPs, see MPLS Label Distribution Protocol: Details, on page 9.

Configuring Label Distribution Protocol

Depending on the requirements, LDP requires some basic configuration tasks described in the following topics:

Configuring Label Distribution Protocol

This section explains the basic LDP configuration. LDP should be enabled on all interfaces that connects the router to potential LDP peer routers. You can enable LDP on an interface by specifying the interface under `mpls ldp configuration` mode.

Configuration Example

This example shows how to enable LDP over an interface.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# router-id 192.168.70.1
RP/0/RP0/CPU0:Router(config-ldp)# interface HundredGigE 0/0/0/5
RP/0/RP0/CPU0:Router(config-ldp-if)# commit
```

Configuring Label Distribution Protocol Discovery Parameters

LSRs that are running LDP sends hello messages on all the LDP enabled interfaces to discover each other. So, the LSR that receives the LDP hello message on an interface is aware of the presence of the LDP router on that interface. If LDP hello messages are sent and received on an interface, there is an LDP adjacency
across the link between the two LSRs that are running LDP. By default, hello messages are sent every 5 seconds with a hold time of 15 seconds. If the LSR does not receive a discovery hello from peer before the hold time expires, the LSR removes the peer LSR from the list of discovered LDP neighbors. The LDP discovery parameters can be configured to change the default parameters.

LDP session between LSRs that are not directly connected is known as targeted LDP session. For targeted LDP sessions, LDP uses targeted hello messages to discover the extended neighbors. By default, targeted hello messages are sent every 10 seconds with a hold time of 90 seconds.

**Configuration Example**

This example shows how to configure the following LDP discovery parameters:

- hello hold time
- hello interval
- targeted hello hold time
- targeted hello interval

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# router-id 192.168.70.1
RP/0/RP0/CPU0:Router(config-ldp)# discovery hello holdtime 30
RP/0/RP0/CPU0:Router(config-ldp)# discovery hello interval 10
RP/0/RP0/CPU0:Router(config-ldp)# discovery targeted-hello holdtime 120
RP/0/RP0/CPU0:Router(config-ldp)# discovery targeted-hello interval 15
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

**Verification**

This section verifies the MPLS LDP discovery parameters configuration.

```
RP/0/RP0/CPU0:Router# show mpls ldp parameters
LDP Parameters:
  Role: Active
  Protocol Version: 1
  Router ID: 192.168.70.1
  Discovery:
   Link Hellos: Holdtime:30 sec, Interval:10 sec
   Targeted Hellos: Holdtime:120 sec, Interval:15 sec
  Quick-start: Enabled (by default)
  Transport address: IPv4: 192.168.70.1
```

**Label Distribution Protocol Discovery for Targeted Hellos**

LDP session between LSRs that are not directly connected is known as targeted LDP session. For LDP neighbors which are not directly connected, you should manually configure the LDP neighborship on both the routers.

**Configuration Example**

This example shows how to configure LDP for non-directly connected routers, Router1 and Router 2.

```
RP/0/RP0/CPU0:Router1(config)# mpls ldp
RP/0/RP0/CPU0:Router1(config-ldp)# router-id 192.168.70.1
```
Label Advertisement Control

LDP allows you to control the advertising and receiving of labels. You can control the exchange of label binding information by using label advertisement control (outbound filtering) or label acceptance control (inbound filtering).

Label Advertisement Control (Outbound Filtering)

Label Distribution Protocol advertises labels for all the prefixes to all its neighbors. When this is not desirable (for scalability and security reasons), you can configure LDP to perform outbound filtering for local label advertisement for one or more prefixes to one more peers. This feature is known as LDP outbound label filtering, or local label advertisement control. You can control the exchange of label binding information using the `mpls ldp label advertise` command. Using the optional keywords, you can advertise selective prefixes to all neighbors, advertise selective prefixes to defined neighbors, or disable label advertisement to all peers for all prefixes. Prefixes and peers advertised selectively are defined in the access list.

Configuration Example: Label Advertisement Control

This example shows how to configure outbound label advertisement control. In this example, neighbors are specified to advertise and receive label advertisements. Also an interface is specified for label advertisement.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)# label local advertise to 1.1.1.1:0 for pfx_ac11
RP/0/RP0/CPU0:Router(config-ldp-af)# label local advertise interface TenGigE 0/0/0/5
RP/0/RP0/CPU0:Router(config-ldp-af)# commit
```

Label Acceptance Control (Inbound Filtering)

LDP accepts labels (as remote bindings) for all prefixes from all peers. LDP operates in liberal label retention mode, which instructs LDP to keep remote bindings from all peers for a given prefix. For security reasons, or to conserve memory, you can override this behavior by configuring label binding acceptance for set of prefixes from a given peer. The ability to filter remote bindings for a defined set of prefixes is also referred to as LDP inbound label filtering or label acceptance control.

Configuration Example : Label Acceptance Control (Inbound Filtering)

This example shows how to configure label acceptance control. In this example, an LSR is configured to accept and retain label bindings from neighbors for prefixes defined in access list.

```
RP/0/RP0/CPU0:Router(config-ldp-if)# commit
```
Configuring Local Label Allocation Control

LDP creates label bindings for all IGP prefixes and receives label bindings for all IGP prefixes from all its peers. If an LSR receives label bindings from several peers for thousands of IGP prefixes, it consumes significant memory and CPU. In some scenarios, most of the LDP label bindings may not be useful for any application and you may require to limit the allocation of local labels. This is accomplished using LDP local label allocation control, where an access list can be used to limit allocation of local labels to a set of prefixes. Limiting local label allocation provides several benefits, including reduced memory usage requirements, fewer local forwarding updates, and fewer network and peer updates.

**Configuration Example**

This example shows how to configure local label allocation using an IP access list to specify a set of prefixes that local labels can allocate and advertise.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)# label remote accept from 192.168.1.1:0 for acl_1
RP/0/RP0/CPU0:Router(config-ldp-af)# label remote accept from 192.168.2.2:0 for acl_2
RP/0/RP0/CPU0:Router(config-ldp-af)# commit
```

Configuring Downstream on Demand

By default, LDP uses downstream unsolicited mode in which label advertisements for all routes are received from all LDP peers. The downstream on demand feature adds support for downstream-on-demand mode, where the label is not advertised to a peer, unless the peer explicitly requests it. At the same time, since the peer does not automatically advertise labels, the label request is sent whenever the next-hop points out to a peer that no remote label has been assigned.

In downstream on demand configuration, an ACL is used to specify the set of peers for downstream on demand mode. For downstream on demand to be enabled, it needs to be configured on both peers of the session. If only one peer in the session has downstream-on-demand feature configured, then the session does not use downstream-on-demand mode.

**Configuration Example**

This example shows how to configure LDP Downstream on Demand.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# address-family ipv4
RP/0/RP0/CPU0:Router(config-ldp-af)# label local allocate for pfx_acl_1
RP/0/RP0/CPU0:Router(config-ldp-af)# commit
```

Configuring Explicit Null Label

Cisco MPLS LDP uses implicit or explicit null label as local label for routes or prefixes that terminate on the given LSR. These routes include all local, connected, and attached networks. By default, the null label is
**Label Distribution Protocol Auto-configuration**

LDP auto-configuration allows you to automatically configure LDP on all interfaces for which the IGP protocol is enabled. Typically, LDP assigns and advertises labels for IGP routes and must often be enabled on all active interfaces by an IGP. During LDP manual configuration, you must define the set of interfaces under LDP which is a time-intensive procedure. LDP auto-configuration eliminates the need to specify the same list of interfaces under LDP and simplifies the configuration tasks.

**Configuration Example: Enabling LDP Auto-Configuration for OSPF**

This example shows how to enable LDP auto-configuration for a specified OSPF instance.

```
RP/0/RP0/CPU0:Router(config)# router ospf 190
RP/0/RP0/CPU0:Router(config-ospf)# mpls ldp auto-config
RP/0/RP0/CPU0:Router(config-ospf)# area 8
RP/0/RP0/CPU0:Router(config-ospf-area)# interface HundredGigE 0/0/0/5
RP/0/RP0/CPU0:Router(config-ospf-area-if)# commit
```

**Configuring Session Protection**

When a new link or node comes up after a link failure, IP converges earlier and much faster than MPLS LDP and may result in MPLS traffic loss until the MPLS convergence. If a link flaps, the LDP session also flaps due to loss of link discovery. LDP session protection minimizes traffic loss, provides faster convergence, and
protects existing LDP (link) sessions. When session protection is enabled for a peer, LDP starts sending targeted hello (directed discovery) in addition to basic discovery link hellos. When the direct link goes down, the targeted hellos can still be forwarded to the peer LSR over an alternative path as long as there is one. So, the LDP session stays up after the link goes down.

You can configure LDP session protection to automatically protect sessions with all or a given set of peers (as specified by peer-acl). When configured, LDP initiates backup targeted hellos automatically for neighbors for which primary link adjacencies already exist. These backup targeted hellos maintain LDP sessions when primary link adjacencies go down.

**Configuration Example**

This example shows how to configure LDP session protection for peers specified by the access control list peer-acl-1 for a maximum duration of 60 seconds.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# session protection for peer-acl-1 duration 60
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

### Configuring Label Distribution Protocol- Interior Gateway Protocol (IGP) Synchronization

Lack of synchronization between LDP and Interior Gateway Protocol (IGP) can cause MPLS traffic loss. Upon link up, for example, IGP can advertise and use a link before LDP convergence has occurred or, a link may continue to be used in IGP after an LDP session goes down.

LDP IGP synchronization coordinates LDP and IGP so that IGP advertises links with regular metrics only when MPLS LDP is converged on that link. LDP considers a link converged when at least one LDP session is up and running on the link for which LDP has sent its applicable label bindings and received at least one label binding from the peer. LDP communicates this information to IGP upon link up or session down events and IGP acts accordingly, depending on sync state.

LDP-IGP synchronization is supported for both OSPF and ISIS protocols and is configured under the corresponding IGP protocol configuration mode. Under certain circumstances, it might be required to delay declaration of re-synchronization to a configurable interval. LDP provides a configuration option to delay declaring synchronization up for up to 60 seconds. LDP communicates this information to IGP upon linkup or session down events.

**Configuring LDP IGP Synchronization: Open Shortest Path First (OSPF) Example**

This example shows how to configure LDP-IGP synchronization for an OSPF instance. The synchronization delay is configured as 30 seconds.

```
RP/0/RP0/CPU0:Router(config)# router ospf 100
RP/0/RP0/CPU0:Router(config-ospf)# mpls ldp sync
RP/0/RP0/CPU0:Router(config-ospf)# mpls ldp igp sync delay 30
RP/0/RP0/CPU0:Router(config-ospf)# commit
```

**Configuring LDP IGP Synchronization: Intermediate System to Intermediate System (IS-IS)**

This example shows how to configure LDP-IGP synchronization for IS-IS.

```
RP/0/RP0/CPU0:Router(config)# router isis 100
```
Configuring Label Distribution Protocol (LDP) Graceful Restart

LDP Graceful Restart provides a mechanism for LDP peers to preserve the MPLS forwarding state when the LDP session goes down. Without LDP Graceful Restart, when an established session fails, the corresponding forwarding states are cleaned immediately from the restart and peer nodes. In this case, LDP forwarding has to restart from the beginning, causing a potential loss of data and connectivity. If LDP graceful restart is configured, traffic can continue to be forwarded without interruption, even when the LDP session restarts. The LDP graceful restart capability is negotiated between two peers during session initialization time. During session initialization, a router advertises its ability to perform LDP graceful restart by sending the graceful restart typed length value (TLV). This TLV contains the reconnect time and recovery time. The values of the reconnect and recovery times indicate the graceful restart capabilities supported by the router. The reconnect time is the amount of time the peer router waits for the restarting router to establish a connection. When a router discovers that a neighboring router is restarting, it waits until the end of the recovery time before attempting to reconnect. Recovery time is the amount of time that a neighboring router maintains its information about the restarting router.

Configuration Example

This example shows how to configure LDP graceful restart. In this example, the amount of time that a neighboring router maintains the forwarding state about the gracefully restarting router is specified as 180 seconds. The reconnect time is configured as 169 seconds.

```bash
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# interface HundredGigE 0/0/0/5
RP/0/RP0/CPU0:Router(config-ldp-if)# exit
RP/0/RP0/CPU0:Router(config-ldp)# graceful-restart
RP/0/RP0/CPU0:Router(config-ldp)# graceful-restart forwarding-state-holdtime 180
RP/0/RP0/CPU0:Router(config-ldp)# graceful-restart reconnect-timeout 169
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Configuring Label Distribution Protocol (LDP) Nonstop Routing

LDP nonstop routing (NSR) functionality makes failures, such as Route Processor (RP) or Distributed Route Processor (DRP) failover, invisible to routing peers with minimal to no disruption of convergence performance. By default, NSR is globally enabled on all LDP sessions except AToM.

A disruption in service may include any of these events:

- Route processor (RP) or distributed route processor (DRP) failover
- LDP process restart
- Minimum disruption restart (MDR)
Unlike graceful restart functionality, LDP NSR does not require protocol extensions and does not force software upgrades on other routers in the network, nor does LDP NSR require peer routers to support NSR. L2VPN configuration is not supported on NSR. Process failures of active LDP results in session loss and, as a result, NSR cannot be provided unless RP switchover is configured as a recovery action.

**Configuration Example**

This example shows how to configure LDP Non-Stop Routing.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# nsr
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

**Verification**

```
RP/0/RP0/CPU0:Router# show mpls ldp nsr summary
Mon Dec 7 04:02:16.259 UTC
Sessions:
  Total: 1, NSR-eligible: 1, Sync-ed: 0
  (1 Ready)
```

**MPLS Label Distribution Protocol : Details**

This section provides detailed conceptual information about setting up LSPs, LDP graceful restart, and LDP session protection.

**Setting Up Label Switched Paths**

MPLS packets are forwarded between the nodes on the MPLS network using Label Switched Paths (LSPs). LSPs can be created statically or by using a label distribution protocol like LDP. Label Switched Paths created by LDP performs hop-by-hop path setup instead of an end-to-end path. LDP enables label switched routers (LSRs) to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information.

The following figure illustrates the process of label binding exchange for setting up LSPs.
For a given network (10.0.0.0), hop-by-hop LSPs are set up between each of the adjacent routers (or, nodes) and each node allocates a local label and passes it to its neighbor as a binding:

1. R4 allocates local label L4 for prefix 10.0.0.0 and advertises it to its neighbors (R3).
2. R3 allocates local label L3 for prefix 10.0.0.0 and advertises it to its neighbors (R1, R2, R4).
3. R1 allocates local label L1 for prefix 10.0.0.0 and advertises it to its neighbors (R2, R3).
4. R2 allocates local label L2 for prefix 10.0.0.0 and advertises it to its neighbors (R1, R3).
5. R1’s label information base (LIB) keeps local and remote labels bindings from its neighbors.
6. R2’s LIB keeps local and remote labels bindings from its neighbors.
7. R3’s LIB keeps local and remote labels bindings from its neighbors.
8. R4’s LIB keeps local and remote labels bindings from its neighbors.

**MPLS Forwarding**

Once the label bindings are learned, MPLS forwarding plane is setup and packets are forwarded as shown in the following figure.
1. Because R3 is next hop for 10.0.0.0 as notified by the FIB, R1 selects label binding from R3 and installs forwarding entry (Layer 1, Layer 3).

2. Because R3 is next hop for 10.0.0.0 (as notified by FIB), R2 selects label binding from R3 and installs forwarding entry (Layer 2, Layer 3).

3. Because R4 is next hop for 10.0.0.0 (as notified by FIB), R3 selects label binding from R4 and installs forwarding entry (Layer 3, Layer 4).

4. Because next hop for 10.0.0.0 (as notified by FIB) is beyond R4, R4 uses NO-LABEL as the outbound and installs the forwarding entry (Layer 4); the outbound packet is forwarded IP-only.

5. Incoming IP traffic on ingress LSR R1 gets label-imposed and is forwarded as an MPLS packet with label L3.

6. Incoming IP traffic on ingress LSR R2 gets label-imposed and is forwarded as an MPLS packet with label L3.

7. R3 receives an MPLS packet with label L3, looks up in the MPLS label forwarding table and switches this packet as an MPLS packet with label L4.

8. R4 receives an MPLS packet with label L4, looks up in the MPLS label forwarding table and finds that it should be Unlabeled, pops the top label, and passes it to the IP forwarding plane.

9. IP forwarding takes over and forwards the packet onward.

Details of Label Distribution Protocol Graceful Restart

LDP (Label Distribution Protocol) graceful restart provides a control plane mechanism to ensure high availability and allows detection and recovery from failure conditions while preserving Nonstop Forwarding (NSF) services. Graceful restart is a way to recover from signaling and control plane failures without impacting forwarding.
Without LDP graceful restart, when an established session fails, the corresponding forwarding states are 
cleaned immediately from the restarting and peer nodes. In this case LDP forwarding restarts from the 
beginning, causing a potential loss of data and connectivity.

The LDP graceful restart capability is negotiated between two peers during session initialization time, in FT 
SESSION TLV. In this typed length value (TLV), each peer advertises the following information to its peers:

**Reconnect time**

Advertises the maximum time that other peer will wait for this LSR to reconnect after control channel 
failure.

**Recovery time**

Advertises the maximum time that the other peer has on its side to reinstate or refresh its states with this 
LSR. This time is used only during session reestablishment after earlier session failure.

**FT flag**

Specifies whether a restart could restore the preserved (local) node state for this flag.

Once the graceful restart session parameters are conveyed and the session is up and running, graceful restart 
procedures are activated.

When configuring the LDP graceful restart process in a network with multiple links, targeted LDP hello 
adjacencies with the same neighbor, or both, make sure that graceful restart is activated on the session before 
any hello adjacency times out in case of neighbor control plane failures. One way of achieving this is by 
configuring a lower session hold time between neighbors such that session timeout occurs before hello 
adjacency timeout. It is recommended to set LDP session hold time using the following formula:

\[
\text{Session Holdtime} \leq (\text{Hello holdtime} - \text{Hello interval}) \times 3
\]

This means that for default values of 15 seconds and 5 seconds for link Hello holdtime and interval respectively, 
session hold time should be set to 30 seconds at most.

**Phases in Graceful Restart**

The graceful restart mechanism is divided into different phases:

**Control communication failure detection**

Control communication failure is detected when the system detects either:

- Missed LDP hello discovery messages
- Missed LDP keepalive protocol messages
- Detection of Transmission Control Protocol (TCP) disconnection a with a peer

**Forwarding state maintenance during failure**

Persistent forwarding states at each LSR are achieved through persistent storage (checkpoint) by the 
LDP control plane. While the control plane is in the process of recovering, the forwarding plane keeps 
the forwarding states, but marks them as stale. Similarly, the peer control plane also keeps (and marks 
as stale) the installed forwarding rewrites associated with the node that is restarting. The combination of 
local node forwarding and remote node forwarding plane states ensures NSF and no disruption in the 
traffic.
Control state recovery

Recovery occurs when the session is reestablished and label bindings are exchanged again. This process allows the peer nodes to synchronize and to refresh stale forwarding states.

Control Plane Failure

When a control plane failure occurs, connectivity can be affected. The forwarding states installed by the router control planes are lost, and the in-transit packets could be dropped, thus breaking NSF. The following figure illustrates control plane failure and recovery with graceful restart and shows the process and results of a control plane failure leading to loss of connectivity and recovery using graceful restart.

Figure 3: Control Plane Failure
Recovery with Graceful Restart

1. The R4 LSR control plane restarts.
2. LIB is lost when the control plane restarts.
3. The forwarding states installed by the R4 LDP control plane are immediately deleted.
4. Any in-transit packets flowing from R3 to R4 (still labeled with L4) arrive at R4.
5. The MPLS forwarding plane at R4 performs a lookup on local label L4 which fails. Because of this failure, the packet is dropped and NSF is not met.
6. The R3 LDP peer detects the failure of the control plane channel and deletes its label bindings from R4.
7. The R3 control plane stops using outgoing labels from R4 and deletes the corresponding forwarding state (rewrites), which in turn causes forwarding disruption.
8. The established LSPs connected to R4 are terminated at R3, resulting in broken end-to-end LSPs from R1 to R4.
9. The established LSPs connected to R4 are terminated at R3, resulting in broken LSPs end-to-end from R2 to R4.

When the LDP control plane recovers, the restarting LSR starts its forwarding state hold timer and restores its forwarding state from the checkpointed data. This action reinstates the forwarding state and entries and marks them as old.

The restarting LSR reconnects to its peer, indicated in the FT Session TLV, that it either was or was not able to restore its state successfully. If it was able to restore the state, the bindings are resynchronized.

The peer LSR stops the neighbor reconnect timer (started by the restarting LSR), when the restarting peer connects and starts the neighbor recovery timer. The peer LSR checks the FT Session TLV if the restarting
peer was able to restore its state successfully. It reinstates the corresponding forwarding state entries and receives binding from the restarting peer. When the recovery timer expires, any forwarding state that is still marked as stale is deleted.

If the restarting LSR fails to recover (restart), the restarting LSR forwarding state and entries will eventually timeout and is deleted, while neighbor-related forwarding states or entries are removed by the Peer LSR on expiration of the reconnect or recovery timers.

**Details of Session Protection**

LDP session protection lets you configure LDP to automatically protect sessions with all or a given set of peers (as specified by peer-acl). When configured, LDP initiates backup targeted hellos automatically for neighbors for which primary link adjacencies already exist. These backup targeted hellos maintain LDP sessions when primary link adjacencies go down.

The Session Protection figure illustrates LDP session protection between neighbors R1 and R3. The primary link adjacency between R1 and R3 is directly connected link and the backup; targeted adjacency is maintained between R1 and R3. If the direct link fails, LDP link adjacency is destroyed, but the session is kept up and running using targeted hello adjacency (through R2). When the direct link comes back up, there is no change in the LDP session state and LDP can converge quickly and begin forwarding MPLS traffic.

*Figure 5: Session Protection*

When LDP session protection is activated (upon link failure), protection is maintained for an unlimited period time.
MPLS Static Labeling

The MPLS static feature enables you to statically assign local labels to an IPv4 prefix. Also, Label Switched Paths (LSPs) can be provisioned for these static labels by specifying the next-hop information that is required to forward the packets containing static label.

If there is any discrepancy between labels assigned statically and dynamically, the router issues a warning message in the console log. By means of this warning message, the discrepancy can be identified and resolved.

The advantages of static labels over dynamic labels are:

- Improve security because the risk of receiving unwanted labels from peers (running a compromised MPLS dynamic labeling protocol) is reduced.
- Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

Restrictions

- Static labeling on IPv6 packets is not supported.
- The router does not prevent label discrepancy at the time of configuring static labels. Any generated discrepancy needs to be subsequently cleared.
- Equal-cost multi-path routing (ECMP) is not supported.
- Interfaces must be explicitly configured to handle traffic with static MPLS labels.
- Restrictions For MPLS, on page 17
- Define Label Range and Enable MPLS Encapsulation, on page 18
- Identify and Clear Label Discrepancy, on page 19
- Configuring Static LSP Next Hop Resolve with Recursive Prefix, on page 20
- Configuring MPLS Static over BVI, on page 21

Restrictions For MPLS

- MPLS statistics is not supported.
Define Label Range and Enable MPLS Encapsulation

By default, MPLS encapsulation is disabled on all interfaces. MPLS encapsulation has to be explicitly enabled on all ingress and egress MPLS interfaces through which the static MPLS labeled traffic travels.

Also, the dynamic label range needs to be defined. Any label that falls outside this dynamic range is available for manually allocating as static labels. The router does not verify statically-configured labels against the specified label range. Therefore, to prevent label discrepancy, ensure that you do not configure static MPLS labels that fall within the dynamic label range.

Configuration Example

You have to accomplish the following to complete the MPLS static labeling configuration. Values are provided as an example.

1. Define a dynamic label range, which in this task is set between 17000 and 18000.
2. Enable MPLS encapsulation on the required interface.
3. Setup a static MPLS LSP for a specific ingress label 24035.
4. Specify the forwarding information so that for packets that are received with the label, 24035, the MPLS protocol swaps labels and applies the label, 24036. After applying the new label, it forwards the packets to the next hop, 10.2.2.2, through the specified interface.

```
RP/0/RP0/CPU0:router(config)#mpls label range table 0 17000 18000
RP/0/RP0/CPU0:router(config)#commit

RP/0/RP0/CPU0:router(config)#mpls static

RP/0/RP0/CPU0:router(config-mpls-static)#address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-mpls-static-af)#local-label 24035 allocate
RP/0/RP0/CPU0:router(config-mpls-static-af-lbl)#forward
RP/0/RP0/CPU0:router(config-mpls-static-af-lbl-fwd)#

RP/0/RP0/CPU0:router(config-mpls-static-af-lbl-fwd)# commit
```

Verification

Verify the interfaces on which MPLS is enabled

```
RP/0/RP0/CPU0:router# show mpls interfaces
Mon May 12 06:21:30.937 DST
Interface LDP Tunnel Static Enabled
------------------------ ----------- ----------- --------
TenGigE0/0/0/5 No No Yes Yes
```

Verify that the status is "Created" for the specified label value.

```
RP/0/RP0/CPU0:router#show mpls static local-label all
Tue Apr 22 18:21:55.764 UTC
Label VRF Type Prefix RW Configured Status
----------- ----------- ----------- ----------------- --------
```
Check the dynamic range and ensure that the specified local-label value is outside this range.

```
RP/0/RP0/CPU0:router#show mpls label range
Mon Apr 28 19:56:00.596 IST
Range for dynamic labels: Min/Max: 17000/18000
```

Verify that the MPLS static configuration has taken effect, and the label forwarding is taking place.

```
RP/0/RP0/CPU0:router#show mpls lsd forwarding
Wed Nov 25 21:40:57.918 UTC
In_Label, (ID), Path_Info: <Type>
24035, (Static), 1 Paths
  1/1: IPv4, 'default':4U, BE1.2, nh=10.20.3.1, lbl=35001, flags=0x0, ext_flags=0x0
```

**Associated Commands**

- `mpls static`
- `mpls label range`
- `show mpls interfaces`

### Identify and Clear Label Discrepancy

During configuring or de-configuring static labels or a label range, a label discrepancy can get generated when:

- A static label is configured for an IP prefix that already has a binding with a dynamic label.
- A static label is configured for an IP prefix, when the same label value is dynamically allocated to another IP prefix.

**Verification**

Identify label discrepancy by using these show commands.

```
Router#show mpls static local-label discrepancy
Tue Apr 22 18:36:31.614 UTC
Label VRF Type Prefix RW Configured Status
------- --------------- ------------ ---------------- --------------- --------
24000 default X-Connect NA Yes Discrepancy
```

```
Router#show mpls static local-label all
Tue Apr 22 18:36:31.614 UTC
Label VRF Type Prefix RW Configured Status
------- --------------- ------------ ---------------- --------------- --------
24000 default X-Connect NA Yes Discrepancy
24035 default X-Connect N/A Yes Created
```

```
RP/0/RP0/CPU0:router#show log
Thu Apr 24 14:18:57.655 UTC
System logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Console logging: level warnings, 199 messages logged
  Monitor logging: level debugging, 0 messages logged
  Trap logging: level informational, 0 messages logged
```
Buffer logging: level debugging, 2 messages logged

Log Buffer (307200 bytes):

RP/0/RSP0/CPU0:Apr 24 14:18:53.743 : mpls_static[1043]:
%ROUTING-MPLS_STATIC-7-ERR_STATIC_LABEL_DISCREPANCY :
The system detected 1 label discrepancies (static label could not be allocated due to
conflict with other applications).
Please use 'clear mpls static local-label discrepancy' to fix this issue.
RP/0/RSP0/CPU0:Apr 24 14:18:53.937 : config[65762]: %MGBL-CONFIG-6-DB_COMMIT : Configuration committed by user 'cisco'.
Use 'show configuration commit changes 100000020' to view the changes.

Rectification

Label discrepancy is cleared by allocating a new label to those IP prefixes that are allocated dynamic label. The static label configuration takes precedence while clearing discrepancy. Traffic can be affected while clearing discrepancy.

Router# clear mpls static local-label discrepancy all

Verify that the discrepancy is cleared.

Router# show mpls static local-label all

<table>
<thead>
<tr>
<th>Label</th>
<th>VRF</th>
<th>Type</th>
<th>Prefix</th>
<th>RW Configured</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>24000</td>
<td>default</td>
<td>X-Connect</td>
<td>N/A</td>
<td>Yes</td>
<td>Created</td>
</tr>
<tr>
<td>24035</td>
<td>default</td>
<td>X-Connect</td>
<td>N/A</td>
<td>Yes</td>
<td>Created</td>
</tr>
</tbody>
</table>

Associated Commands

• show mpls static local-label discrepancy

• clear mpls static local-label discrepancy all

Configuring Static LSP Next Hop Resolve with Recursive Prefix

When a routing table entry references to another IP address and not to a directly connected exit interface, the next-hop IP address is resolved using another route with an exit interface. This is known as a recursive look up because multiple lookups are required to resolve the next-hop IP address. Static LSP next hop resolve with recursive prefix feature supports resolution of recursive routes for static LSPs. In this feature, you can specify a next hop which is not directly connected using the resolve-nexthop command for a static LSP.

Restrictions

The following restrictions apply for this feature:

• Only eBGP routes are supported.

Configuration Example

This example shows how to configure the static LSP next hop resolve with recursive prefix. Here 192.168.2.1 is a recursive route learnt through eBGP.
Configuring MPLS Static over BVI

A Bridge-group virtual interface (BVI) is a routed interface that represents a set of interfaces that gets bridged. By using BVI, you can convert multiple interfaces as members of a common broadcast domain. MPLS static over BVI feature allows you to specify a BVI interface as nexthop while setting up a static LSP.

Restrictions

• MPLS static with resolve nexthop to directly connected subnet prefix for BVI is not supported.
• Back up paths over BVI is not supported.
• Fast Reroute (FRR) is not supported.

Configuration Example

This example shows how to configure a BVI interface as next hop for a static LSP.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# interface TenGig 0/0/0/0
RP/0/0/CPU0:Router(config-mpls-static)# lsp bvi
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 5001 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 192.168.2.1 out-label pop
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
```
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).

- Overview of MPLS-TE Features, on page 24
- Restrictions For MPLS-TE, on page 24
- How MPLS-TE Works, on page 24
- Configuring MPLS-TE, on page 25
- MPLS-TE Features - Details, on page 36
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 MPLS-TE Features - Details, on page 36.

Restrictions For MPLS-TE

- MPLS-TE statistics is not supported.

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.

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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config)# router ospf area 1
RP/0/RP0/CPU0:router(config-ospf)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-ospf-ar)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1
RP/0/RP0/CPU0: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.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config)# router isis 1
RP/0/RP0/CPU0:router(config-isis)# net 47.0001.0000.0000.0002.00
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng level 1
RP/0/RP0/CPU0:router(config-isis-af)# exit
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-isis-if)# exit
RP/0/RP0/CPU0:router(config)# commit
```

**Related Topics**

- How MPLS-TE Works, on page 24
- Creating an MPLS-TE Tunnel, on page 26

**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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic
RP/0/RP0/CPU0:router(config-if)# autoroute announce or forwarding adjacency
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 100
RP/0/RP0/CPU0:router(config)# commit
```

Verification

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

```
RP/0/RP0/CPU0:router# show mpls traffic-engineering tunnels brief
```

Relationship 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
```

Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

Related Topics

- How MPLS-TE Works, on page 24
- Building MPLS-TE Topology, on page 25

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 MPLS-TE Features - Details, on page 36

**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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# fast-reroute
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# backup-path tunnel-te 2
RP/0/RP0/CPU0:router(config-if)# interface tunnel-te 2
RP/0/RP0/CPU0:router(config-if)# backup-bw global-pool 5000
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-path protected by 10
RP/0/RP0/CPU0:router(config)# commit
```

**Verification**

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

```
RP/0/RP0/CPU0:router# show mpls traffic-eng fast-reroute database
```

```
Tunnel head FRR information:
Tunnel Out intf/label FRR intf/label Status
---------- ---------------- ---------------- -------
tt4000 HundredGigabitEthernet 0/0/1/0:34 tt1000:34 Ready
tt4001 HundredGigabitEthernet 0/0/1/0:35 tt1001:35 Ready
tt4002 HundredGigabitEthernet 0/0/1/0:36 tt1001:36 Ready
```

**Related Topics**

- Configuring MPLS-TE, on page 25
- Configuring Auto-Tunnel Backup, on page 29
- Configuring Next Hop Backup Tunnel, on page 30
- MPLS-TE Features - Details, on page 36
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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup
RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)# attribute-set ab
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup timers removal unused 20
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
RP/0/RP0/CPU0:router(config)# commit
```

Verification

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

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

<table>
<thead>
<tr>
<th>TUNNEL NAME</th>
<th>DESTINATION</th>
<th>STATUS</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel-te0</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>tunnel-te1</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>tunnel-te2</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>tunnel-te50</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>*tunnel-te60</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>*tunnel-te70</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>*tunnel-te80</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>
```

Related Topics

- Configuring Fast Reroute, on page 27
- Configuring Next Hop Backup Tunnel, on page 30
- MPLS-TE Features - Details, on page 36
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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup nhop-only
RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)# attribute-set ab
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup timers removal unused 20
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
RP/0/RP0/CPU0:router(config)# commit
```

Related Topics
- Configuring Auto-Tunnel Backup, on page 29
- Configuring Fast Reroute, on page 27
- MPLS-TE Features - Details, on page 36

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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# backup-path tunnl-te 2
RP/0/RP0/CPU0:router(config-mpls-te-if)# exit
RP/0/RP0/CPU0:router(config)# interface tunnel-te 2
RP/0/RP0/CPU0:router(config)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg
RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg-nodep
```
**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.

**Configuration Example**

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

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# affinity-map red 1
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# attribute-names red
RP/0/RP0/CPU0:router(config)# interface tunnel-te 2
RP/0/RP0/CPU0:router(config-if)# affinity include red
RP/0/RP0/CPU0:router(config)# commit
```

**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.

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

**Table 1: Automatic Bandwidth Parameters**

<table>
<thead>
<tr>
<th>Bandwidth Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application frequency</td>
<td>Configures how often the tunnel bandwidths changed for each tunnel. The default value is 24 hours.</td>
</tr>
<tr>
<td>Bandwidth limit</td>
<td>Configures the minimum and maximum automatic bandwidth to set on a tunnel.</td>
</tr>
</tbody>
</table>
Bandwidth Parameters | Description
--- | ---
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.

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

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# auto-bw
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# application 1000
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# bw-limit min 30 max 1000
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# adjustment-threshold 50 min 800
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# overflow threshold 100 limit 1
RP/0/RP0/CPU0:router(config)# commit
```

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

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-bw brief
Tunnel Name | LSP ID | Last appl BW(kbps) | Requested BW(kbps) | Signalled BW(kbps) | Highest BW(kbps) | Application Time Left
-------------- | ------ | ------------------ | ------------------- | ----------------- | ----------------- | ------------------------
tunnel-te1 5 | 500 | 300 | 420 | 1h 10m
```  

Related Topics
- [MPLS-TE Features - Details, on page 36](#)

**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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel mesh
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# tunnel-id min 1000 max 2000
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# group 10
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# attribute-set 10
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# destination-list dl-65
RP/0/RP0/CPU0:router(config-mpls-te)# attribute-set auto-mesh 10
RP/0/RP0/CPU0:router(config-mpls-te-attribute-set)# autoroute announce
RP/0/RP0/CPU0:router(config-mpls-te-attribute-set)# auto-bw collect-bw-only
RP/0/RP0/CPU0:router(config)# commit
```

**Verification**

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

```
RP/0/RP0/CPU0: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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Tunnel ID</th>
<th>State</th>
<th>Unused timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.2</td>
<td>1000</td>
<td>up</td>
<td>Not running</td>
</tr>
<tr>
<td>192.168.0.3</td>
<td>1001</td>
<td>up</td>
<td>Not running</td>
</tr>
<tr>
<td>192.168.0.4</td>
<td>1002</td>
<td>up</td>
<td>Not running</td>
</tr>
</tbody>
</table>

Displayed 3 tunnels, 3 up, 0 down, 0 FRR enabled

Auto-mesh Cumulative Counters:
Total
Created: 3
Connected: 0
Removed (unused): 0
```
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.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# explicit-path name interarea1
RP/0/RP0/CPU0:router(config-expl-path)# index1 next-address loose ipv4 unicast 172.16.255.129
RP/0/RP0/CPU0:router(config-expl-path)# index 2 next-address loose ipv4 unicast 172.16.255.131
RP/0/RP0/CPU0:router(config)# interface tunnel-te1
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# destination 172.16.255.2
RP/0/RP0/CPU0:router(config-if)# path-option 10 explicit name interarea1
RP/0/RP0/CPU0:router(config)# commit
```

Related Topics

- **MPLS-TE Features - Details, on page 36**

Configure Policy-Based Tunnel Selection

Configuring PBTS is a process of directing incoming traffic into specific TE tunnels based on a classification criteria (DSCP). The traffic forwarding decisions are made based on the categorized traffic classes and the destination network addresses. The following section lists the steps to configure PBTS on a MPLS-TE Tunnel network:

1. Define a class-map based on a classification criteria.
2. Define a policy-map by creating rules for the classified traffic.
3. Associate a forward-class to each type of ingress traffic.
4. Enable PBTS on the ingress interface, by applying this service-policy.
5. Create one or more egress MPLS-TE Tunnels (to carry packets based on priority) to the destination.
6. Associate the egress MPLS-TE Tunnel to a forward-class.

For more information on PBTS, see **Policy-Based Tunnel Selection, on page 39** in the *Implementing MPLS Traffic Engineering* chapter.

Configuration Example

The following section illustrates PBTS implementation:
RP/0/RP0/CPU0:router#configure
/* Class-map; classification using DSCP */
RP/0/RP0/CPU0:router(config)# class-map match-any AF41-Class
RP/0/RP0/CPU0:router(config-cmap)# match dscp AF41
RP/0/RP0/CPU0:router(config-cmap)# exit
/* Policy-map */
RP/0/RP0/CPU0:router(config)# policy-map INGRESS-POLICY
RP/0/RP0/CPU0:router(config-pmap)# class AF41-Class
/* Associating forward class */
RP/0/RP0/CPU0:router(config-pmap-c)# set forward-class 1
RP/0/RP0/CPU0:router(config-pmap-c)# exit
RP/0/RP0/CPU0:router(config-pmap)# exit
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1
/* Applying service-policy to ingress interface */
RP/0/RP0/CPU0:router(config-if)# service-policy input INGRESS-POLICY
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/CPU0:router(config-if)# exit
/* Creating TE-tunnels to carry traffic based on priority */
RP/0/RP0/CPU0:router(config)# interface tunnel-te61
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 1000
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if)# destination 10.20.20.1
RP/0/RP0/CPU0:router(config-if)# record route
/* Associating egress TE tunnels to forward class */
RP/0/RP0/CPU0:router(config-if)# forward-class 1
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit identifier 61
RP/0/RP0/CPU0:router(config-if)# exit

Verification
Use show mpls forwarding tunnels command to verify the PBTS configuration:

RP/0/RP0/CPU0:ios# show mpls forwarding tunnels 10 detail
Tue May 16 01:18:19.681 UTC
Tunnel Outgoing Outgoing Next Hop Bytes
Name Label Interface Switched
----------------- ------------- -------------- --------------
tt10 Exp-Null-v4 Te0/0/0/16 20.20.17.21 0

Updated: May 11 19:31:54.716
Version: 483, Priority: 2
Label Stack (Top -> Bottom): { 0 }
NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
MAC/Encaps: 14/18, MTU: 1500
Packets Switched: 0

Interface:
Name: tunnel-te10 (ifhandle 0x0800005c)
Local Label: 64016, Forwarding Class: 1, Weight: 0

Packets/Bytes Switched: 0/0
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.

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
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**

![Interarea OSPF TE Network Diagram](image1)

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

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

![Interlevel IS-IS TE Network Diagram](image2)

As shown in the **Figure 9: Interlevel (IS-IS) TE Network Diagram**, on page 38, 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.
Policy-Based Tunnel Selection

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

PBTS works by selecting tunnels based on the classification criteria of the incoming packets, which are based on the IP precedence or differentiated services code point (DSCP), or the Type of Service (ToS) fields in the packets. The traffic forwarding decisions are made based on the traffic classes AND the destination network addresses instead of only considering the destination network.

Default-class configured for paths is always zero (0). If there is no TE for a given forward-class, then the default-class (0) will be tried. If there is no default-class, then the packet is tried against the lowest configured forward-class tunnels. PBTS supports up to seven (exp 1 - 7) EXP values associated with a single TE-tunnel.

The following figure illustrates PBTS Network Topology:

**Figure 10: Policy-Based Tunnel Selection Implementation**

- Tunnels are created between Ingress and Egress nodes through LSR 1-2 and LSR 1-3-4-2 paths.
- High priority traffic takes the path: Ingress->LSR1->LSR2->Egress.
- Low priority traffic takes the path: Ingress->LSR1->LSR3->LSR4->LSR2->Egress

**PBTS Function Details**

The following PBTS functions are supported on the Cisco NCS 540 Series Routers:

- Classify the Ingress traffic into different classes by creating rules using PBR configuration.
- Classify packets using DSCP/IP precedence for both IPv4 and IPv6 traffic.
- After classification, set the desired forward-class to each type of Ingress traffic.
- Define one or many MPLS-TE tunnels in the destination using Tunnel configuration.
- Associate the MPLS-TE tunnels to a specific forward-class under Tunnel configuration.
- Enable PBTS on the Ingress interface by applying the service policy that uses the configured classification rules.

The following list gives PBTS support information:

- PBTS is supported only on IPv4/IPv6 incoming traffic only.
- A maximum of eight forward-classes per destination prefix is supported.
- A maximum of 64 TE-tunnels within each forward class is supported.
• A maximum of 64 TE-tunnels can be configured on a given destination.
• Incoming labeled traffic is not supported.
• PBTS with L2VPN/L3VPN traffic is not supported.

PBTS Forward Class

A class-map is defined for various types of packets and these class-maps are associated with a forward-class. A class-map defines the matching criteria for classifying a particular type of traffic and a 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 traffic engineering (TE) tunnel interfaces that the packets should take for each forwarding-class is specified by associating the TE interface explicitly (or implicitly in case of default value) with the forward-group.

When the TE interfaces are associated with the forward-class, they can be exported to the routing protocol module using the **auto-route** command. This will then associate the route in the FIB database with these tunnels. If the TE interface is not explicitly associated with a forward-class, it gets associated with a default-class (0). All non-TE interfaces will be routed to the forwarding plane (with forward-class set to default-class) by the routing protocol.
CHAPTER 4

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).

- Overview of MPLS-TE Features, on page 42
- Restrictions For MPLS-TE, on page 42
- How MPLS-TE Works, on page 42
- Configuring MPLS-TE, on page 43
- MPLS-TE Features - Details, on page 57
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 backup 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 MPLS-TE Features - Details, on page 36.

Restrictions For MPLS-TE

- MPLS-TE statistics is not supported.

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.

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.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config)# router ospf area 1
RP/0/RP0/CPU0:router(config-ospf)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-ospf-ar)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1
RP/0/RP0/CPU0: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.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config)# router isis 1
RP/0/RP0/CPU0:router(config-isis)# net 47.0001.0000.0000.0002.00
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide
RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng level 1
RP/0/RP0/CPU0:router(config-isis-af)# exit
RP/0/RP0/CPU0:router(config-isis)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-isis-if)# exit
RP/0/RP0/CPU0:router(config)# commit
```

Related Topics

• How MPLS-TE Works, on page 24
• Creating an MPLS-TE Tunnel, on page 26

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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic
RP/0/RP0/CPU0:router(config-if)# autoroute announce or forwarding adjacency
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 100
RP/0/RP0/CPU0:router(config)# commit
```

Verification

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

```
RP/0/RP0/CPU0: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

Related Topics

- How MPLS-TE Works, on page 24
- Building MPLS-TE Topology, on page 25

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 MPLS-TE Features - Details, on page 36

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.

```plaintext
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# fast-reroute
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# backup-path tunnel-te 2
RP/0/RP0/CPU0:router(config)# interface tunnel-te 2
RP/0/RP0/CPU0:router(config-if)# backup-bw global-pool 5000
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125
RP/0/RP0/CPU0:router(config-if)# path-option l explicit name backup-path protected by 10
RP/0/RP0/CPU0:router(config)# commit
```

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

```plaintext
RP/0/RP0/CPU0:router# show mpls traffic-eng fast-reroute database
```

<table>
<thead>
<tr>
<th>Tunnel head FRR information:</th>
<th>FRR intf/label</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel Out intf/label</td>
<td>FRR intf/label</td>
<td>--------</td>
</tr>
<tr>
<td>tt4000 HundredGigabitEthernet 0/0/1/0:34 tt1000:34 Ready</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tt4001 HundredGigabitEthernet 0/0/1/0:35 tt1001:35 Ready</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tt4002 HundredGigabitEthernet 0/0/1/0:36 tt1001:36 Ready</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
- Configuring MPLS-TE, on page 25
- Configuring Auto-Tunnel Backup, on page 29
- Configuring Next Hop Backup Tunnel, on page 30
- MPLS-TE Features - Details, on page 36
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.

RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup
RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)# attribute-set ab
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup timers removal unused 20
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
RP/0/RP0/CPU0:router(config)# commit

Verification

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

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels brief

<table>
<thead>
<tr>
<th>TUNNEL NAME</th>
<th>DESTINATION</th>
<th>STATUS</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel-te0</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>tunnel-te1</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>tunnel-te2</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>tunnel-te50</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>*tunnel-te60</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>*tunnel-te70</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>*tunnel-te80</td>
<td>200.0.0.3</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

Related Topics

- Configuring Fast Reroute, on page 27
- Configuring Next Hop Backup Tunnel, on page 30
- MPLS-TE Features - Details, on page 36
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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup nhop-only
RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)# attribute-set ab
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup timers removal unused 20
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
RP/0/RP0/CPU0:router(config)# commit
```

Related Topics

- Configuring Auto-Tunnel Backup, on page 29
- Configuring Fast Reroute, on page 27
- MPLS-TE Features - Details, on page 36

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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# backup-path tunnl-te 2
RP/0/RP0/CPU0:router(config-mpls-te-if)# exit
RP/0/RP0/CPU0:router(config)# interface tunnel-te 2
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg
RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg-nodep
```
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.

Configuration Example

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

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# affinity-map red 1
RP/0/RP0/CPU0:router(config-mpls-te)# interface HundredGigabitEthernet0/0/1/0
RP/0/RP0/CPU0:router(config-mpls-te-if)# attribute-names red
RP/0/RP0/CPU0:router(config)# interface tunnel-te 2
RP/0/RP0/CPU0:router(config-if)# affinity include red
RP/0/RP0/CPU0:router(config)# commit

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.

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

<table>
<thead>
<tr>
<th>Bandwidth Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application frequency</td>
<td>Configures how often the tunnel bandwidths changed for each tunnel. The default value is 24 hours.</td>
</tr>
<tr>
<td>Bandwidth limit</td>
<td>Configures the minimum and maximum automatic bandwidth to set on a tunnel.</td>
</tr>
</tbody>
</table>
### Bandwidth Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth collection frequency</td>
<td>Enables bandwidth collection without adjusting the automatic bandwidth. The default value is 5 minutes.</td>
</tr>
<tr>
<td>Overflow threshold</td>
<td>Configures tunnel overflow detection.</td>
</tr>
<tr>
<td>Adjustment threshold</td>
<td>Configures the tunnel-bandwidth change threshold to trigger an adjustment.</td>
</tr>
</tbody>
</table>

### 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

```bash
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# auto-bw
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# application 1000
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# bw-limit min 30 max 1000
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# adjustment-threshold 50 min 800
RP/0/RP0/CPU0:router(config-if-tunte-autobw)# overflow threshold 100 limit 1
RP/0/RP0/CPU0:router(config)# commit
```

### Verification

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

```bash
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-bw brief
```

### Related Topics

- [MPLS-TE Features - Details, on page 36](#)

### 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.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel mesh
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# tunnel-id min 1000 max 2000
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# group 10
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# attribute-set 10
RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# destination-list dl-65
RP/0/RP0/CPU0:router(config-mpls-te)# attribute-set auto-mesh 10
RP/0/RP0/CPU0:router(config-mpls-te-attribute-set)# autoroute announce
RP/0/RP0/CPU0:router(config-mpls-te-attribute-set)# auto-bw collect-bw-only
RP/0/RP0/CPU0:router(config)# commit
```

**Verification**

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

```
RP/0/RP0/CPU0: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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Tunnel ID</th>
<th>State</th>
<th>Unused timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.2</td>
<td>1000</td>
<td>up</td>
<td>Not running</td>
</tr>
<tr>
<td>192.168.0.3</td>
<td>1001</td>
<td>up</td>
<td>Not running</td>
</tr>
<tr>
<td>192.168.0.4</td>
<td>1002</td>
<td>up</td>
<td>Not running</td>
</tr>
</tbody>
</table>

Displayed 3 tunnels, 3 up, 0 down, 0 FRR enabled

Auto-mesh Cumulative Counters:
Total
Created: 3
Connected: 0
Removed (unused): 0
```
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.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# explicit-path name interarea1
RP/0/RP0/CPU0:router(config-expl-path)# index1 next-address loose ipv4 unicast 172.16.255.129
RP/0/RP0/CPU0:router(config-expl-path)# index2 next-address loose ipv4 unicast 172.16.255.131
RP/0/RP0/CPU0:router(config)# interface tunnel-te1
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# destination 172.16.255.2
RP/0/RP0/CPU0:router(config-if)# path-option 10 explicit name interarea1
RP/0/RP0/CPU0:router(config)# commit
```

Related Topics

- MPLS-TE Features - Details, on page 36

Configure Policy-Based Tunnel Selection

Configuring PBTS is a process of directing incoming traffic into specific TE tunnels based on a classification criteria (DSCP). The traffic forwarding decisions are made based on the categorized traffic classes and the destination network addresses. The following section lists the steps to configure PBTS on a MPLS-TE Tunnel network:

1. Define a class-map based on a classification criteria.
2. Define a policy-map by creating rules for the classified traffic.
3. Associate a forward-class to each type of ingress traffic.
4. Enable PBTS on the ingress interface, by applying this service-policy.
5. Create one or more egress MPLS-TE Tunnels (to carry packets based on priority) to the destination.
6. Associate the egress MPLS-TE Tunnel to a forward-class.

For more information on PBTS, see Policy-Based Tunnel Selection, on page 39 in the Implementing MPLS Traffic Engineering chapter.

Configuration Example

The following section illustrates PBTS implementation:
RP/0/RP0/CPU0:router#configure
/* Class-map; classification using DSCP */
RP/0/RP0/CPU0:router(config)# class-map match-any AF41-Class
RP/0/RP0/CPU0:router(config-cmap)# match dscp AF41
RP/0/RP0/CPU0:router(config-cmap)# exit

/* Policy-map */
RP/0/RP0/CPU0:router(config)# policy-map INGRESS-POLICY
RP/0/RP0/CPU0:router(config-pmap)# class AF41-Class
/* Associating forward class */
RP/0/RP0/CPU0:router(config-pmap-c)# set forward-class 1
RP/0/RP0/CPU0:router(config-pmap-c)# exit
RP/0/RP0/CPU0:router(config-pmap)# exit
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1
/* Applying service-policy to ingress interface */
RP/0/RP0/CPU0:router(config-if)# service-policy input INGRESS-POLICY
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/CPU0:router(config-if)# exit

/* Creating TE-tunnels to carry traffic based on priority */
RP/0/RP0/CPU0:router(config)# interface tunnel-te61
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 1000
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if)# destination 10.20.20.1
RP/0/RP0/CPU0:router(config-if)# record route

/* Associating egress TE tunnels to forward class */
RP/0/RP0/CPU0:router(config-if)# forward-class 1
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit identifier 61
RP/0/RP0/CPU0:router(config-if)# exit

Verification
Use `show mpls forwarding tunnels` command to verify the PBTS configuration:

```
RP/0/RP0/CPU0:ios# show mpls forwarding tunnels 10 detail
Tue May 16 01:18:19.681 UTC
Tunnel Outgoing Outgoing Next Hop Bytes
Name Label Interface Switched
------------- ----------- --------------- --------------
tt10 Exp-Null-v4 Te0/0/0/16 20.20.17.21 0

Updated: May 11 19:31:54.716
Version: 483, Priority: 2
Label Stack (Top -> Bottom): { 0 }
NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
MAC/Encaps: 14/18, MTU: 1500
Packets Switched: 0

Interface:
Name: tunnel-te10 (ifhandle 0x0800005c)
Local Label: 64016, Forwarding Class: 1, Weight: 0

Packets/Bytes Switched: 0/0
```
Configuring LDP over MPLS-TE

LDP and RSVP-TE are signaling protocols used for establishing LSPs in MPLS networks. While LDP is easy to configure and reliable, it lacks the traffic engineering capabilities of RSVP that helps to avoid traffic congestions. LDP over MPLS-TE feature combines the benefits of both LDP and RSVP. In LDP over MPLS-TE, an LDP signalled label-switched path (LSP) runs through a TE tunnel established using RSVP-TE.

The following diagram explains a use case for LDP over MPLS-TE. In this diagram, LDP is used as the signalling protocol between provider edge (PE) router and provider (P) router. RSVP-TE is used as the signalling protocol between the P routers to establish an LSP. LDP is tunneled over the RSVP-TE LSP.

![Diagram showing LDP over MPLS-TE](image)

Restrictions and Guidelines for LDP over MPLS-TE

The following restrictions and guidelines apply for this feature in Cisco IOS-XR release 6.3.2:

- MPLS services over LDP over MPLS-TE are supported when BGP neighbours are on the head or tail node of the TE tunnel.
- MPLS services over LDP over MPLS-TE are supported when the TE headend router is acting as transit point for that service.
- If MPLS services are originating from the TE headend, but the TE tunnel is ending before the BGP peer, LDP over MPLS-TE feature is not supported.
- If LDP optimization is enabled using the `hw-module fib mpls ldp lsr-optimized` command, the following restrictions apply:
  - EVPN is not supported.
  - For any prefix or label all outgoing paths has to be LDP enabled.

Configuration Example:

This example shows how to configure an MPLS-TE tunnel from provider router P1 to P2 and then enable LDP over MPLS-TE. In this example, the destination of the tunnel from P1 is configured as the loop back for P2.
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.

```
RP/0/RP0/CPU0:router # configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic
RP/0/RP0/CPU0:router(config-if)# exit
RP/0/RP0/CPU0:router(config)# mpls ldp
RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.1.1
RP/0/RP0/CPU0:router(config-ldp)# interface TenGigE 0/0/0/0
RP/0/RP0/CPU0:router(config-ldp-if)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-ldp-if)# exit
```
Verifying MPLS-TE Path Protection

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

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 0
Fri Oct 13 16:24:39.379 UTC
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
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:
      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

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.

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 12: 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.
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 14: Interarea (OSPF) TE Network Diagram*

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

*Figure 15: Interlevel (IS-IS) TE Network Diagram*

As shown in the *Figure 15: Interlevel (IS-IS) TE Network Diagram*, on page 59, 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.

**Policy-Based Tunnel Selection**

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

PBTS works by selecting tunnels based on the classification criteria of the incoming packets, which are based on the IP precedence or differentiated services code point (DSCP), or the Type of Service (ToS) fields in the packets. The traffic forwarding decisions are made based on the traffic classes AND the destination network addresses instead of only considering the destination network.

Default-class configured for paths is always zero (0). If there is no TE for a given forward-class, then the default-class (0) will be tried. If there is no default-class, then the packet is tried against the lowest configured forward-class tunnels. PBTS supports up to seven (exp 1 - 7) EXP values associated with a single TE-tunnel.

The following figure illustrates PBTS Network Topology:

*Figure 16: Policy-Based Tunnel Selection Implementation*

- Tunnels are created between Ingress and Egress nodes through LSR 1-2 and LSR 1-3-4-2 paths.
- High priority traffic takes the path: Ingress->LSR1->LSR2->Egress.
- Low priority traffic takes the path: Ingress->LSR1->LSR3->LSR4->LSR2->Egress

**PBTS Function Details**

The following PBTS functions are supported on the Cisco NCS 540 Series Routers:
• Classify the Ingress traffic into different classes by creating rules using PBR configuration.
• Classify packets using DSCP/IP precedence for both IPv4 and IPv6 traffic.
• After classification, set the desired forward-class to each type of Ingress traffic.
• Define one or many MPLS-TE tunnels in the destination using Tunnel configuration.
• Associate the MPLS-TE tunnels to a specific forward-class under Tunnel configuration.
• Enable PBTS on the Ingress interface by applying the service policy that uses the configured classification rules.

The following list gives PBTS support information:

• PBTS is supported only on IPv4/IPv6 incoming traffic only.
• A maximum of eight forward-classes per destination prefix is supported.
• A maximum of 64 TE-tunnels within each forward class is supported.
• A maximum of 64 TE-tunnels can be configured on a given destination.
• Incoming labeled traffic is not supported.
• PBTS with L2VPN/L3VPN traffic is not supported.

**PBTS Forward Class**

A class-map is defined for various types of packets and these class-maps are associated with a forward-class. A class-map defines the matching criteria for classifying a particular type of traffic and a 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 traffic engineering (TE) tunnel interfaces that the packets should take for each forwarding-class is specified by associating the TE interface explicitly (or implicitly in case of default value) with the forward-group.

When the TE interfaces are associated with the forward-class, they can be exported to the routing protocol module using the **auto-route** command. This will then associate the route in the FIB database with these tunnels. If the TE interface is not explicitly associated with a forward-class, it gets associated with a default-class (0). All non-TE interfaces will be routed to the forwarding plane (with forward-class set to default-class) by the routing protocol.
Policy-Based Tunnel Selection
Implementing MPLS OAM

MPLS Operations, Administration, and Maintenance (OAM) helps service providers to monitor label-switched paths (LSPs) and quickly isolate MPLS forwarding problems to assist with fault detection and troubleshooting in an MPLS network. This module describes MPLS LSP Ping and Traceroute features which can be used for failure detection and troubleshooting of MPLS networks.

- MPLS LSP Ping, on page 63
- MPLS LSP Traceroute, on page 65

MPLS LSP Ping

The MPLS LSP Ping feature is used to check the connectivity between Ingress LSR and egress LSRs along an LSP. MPLS LSP ping uses MPLS echo request and reply messages, similar to Internet Control Message Protocol (ICMP) echo request and reply messages, to validate an LSP. While ICMP echo request and reply messages validate IP networks, MPLS echo and reply messages validate MPLS networks. The MPLS echo request packet is sent to a target router through the use of the appropriate label stack associated with the LSP to be validated. Use of the label stack causes the packet to be forwarded over the LSP itself. The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.x.y.z/8 address and it prevents the IP packet from being IP switched to its destination, if the LSP is broken.

An MPLS echo reply is sent in response to an MPLS echo request. The reply is sent as an IP packet and it is forwarded using IP, MPLS, or a combination of both types of switching. The source address of the MPLS echo reply packet is an address obtained from the router generating the echo reply. The destination address is the source address of the router that originated the MPLS echo request packet. The MPLS echo reply destination port is set to the echo request source port.

The following figure shows MPLS LSP ping echo request and echo reply paths.
Configuration Examples

This example shows how to use MPLS LSP ping to test the connectivity of an IPv4 LDP LSP. The destination is specified as a Label Distribution Protocol (LDP) IPv4 prefix and Forwarding Equivalence Class (FEC) type is specified as generic.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 fec-type generic
```

```
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32, timeout is 2 seconds, send interval is 0 msec:


Type escape sequence to abort.

!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

This example shows how to use MPLS LSP ping to test the connectivity when the destination is specified as a MPLS traffic engineering (TE) tunnel.

```
RP/0/RP0/CPU0:router# ping mpls traffic-eng tunnel-te 4003 source 10.1.1.2
```

```
Sending 5, 100-byte MPLS Echos to tunnel-te4003, timeout is 2 seconds, send interval is 0 msec:


Type escape sequence to abort.

!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/3/4 ms
```

This example shows how to use the `show mpls oam` command to display the MPLS OAM information.
RP/0/RP0/CPU0:router# show mpls oam counters packet

Wed Nov 25 03:38:07.397 UTC Global Packet Statistics:

<table>
<thead>
<tr>
<th>Pkt</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive Counts:</td>
<td></td>
</tr>
<tr>
<td>Good Requests:</td>
<td>0 0</td>
</tr>
<tr>
<td>Good Replies:</td>
<td>10 760</td>
</tr>
<tr>
<td>Unknown Pkt Types:</td>
<td>0 0</td>
</tr>
<tr>
<td>IP header error:</td>
<td>0 0</td>
</tr>
<tr>
<td>UDP header error:</td>
<td>0 0</td>
</tr>
<tr>
<td>Runts:</td>
<td>0 0</td>
</tr>
<tr>
<td>Dropped (Q full):</td>
<td>0 0</td>
</tr>
<tr>
<td>General error:</td>
<td>0 0</td>
</tr>
<tr>
<td>Error, no IF:</td>
<td>0 0</td>
</tr>
<tr>
<td>Error, no memory:</td>
<td>0 0</td>
</tr>
<tr>
<td>Transmit Counts:</td>
<td></td>
</tr>
<tr>
<td>Good:</td>
<td>10 960</td>
</tr>
<tr>
<td>Dropped:</td>
<td>0 0</td>
</tr>
</tbody>
</table>

MPLS LSP Traceroute

The MPLS LSP Traceroute feature is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP Traceroute feature relies on the expiration of the Time to Live (TTL) value of the packet that carries the echo request. When the MPLS echo request message hits a transit node, it checks the TTL value and if it is expired, the packet is passed to the control plane, else the message is forwarded. If the echo message is passed to the control plane, a reply message is generated based on the contents of the request message.

The following figure shows an MPLS LSP traceroute example with an LSP from LSR1 to LSR4.

Figure 18: MPLS LSP Traceroute

Configuration Examples

This example shows how to use the traceroute command to trace to a destination with Forwarding Equivalence Class (FEC) type specified as generic.

RP/0/RP0/CPU0:router# traceroute mpls ipv4 3.3.3.3/32 fec-type generic
Mon Nov 30 17:48:45.585 UTC
Tracing MPLS Label Switched Path to 3.3.3.3/32, timeout is 2 seconds

Codes: '!': success, 'Q': request not sent, '.': timeout,
    'L': labeled output interface, 'B': unlabeled output interface,
    'D': DS Map mismatch, 'F': no FEC mapping, 'f': FEC mismatch,
    'M': malformed request, 'm': unsupported tlv, 'N': no rx label,
    'P': no rx intf label prot, 'p': premature termination of LSP,
    'R': transit router, 'I': unknown upstream index,
    'X': unknown return code, 'x': return code 0

Type escape sequence to abort.

0 11.1.1.57 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 11.1.1.58 7 ms 23:19