

MPLS Traffic Engineering Class-based Tunnel Selection

The MPLS Traffic Engineering (TE): Class-based Tunnel Selection feature enables you to dynamically route and forward traffic with different class of service (CoS) values onto different TE tunnels between the same tunnel headend and the same tailend. The TE tunnels can be regular TE or DiffServ-aware TE (DS-TE) tunnels.

The set of TE (or DS-TE) tunnels from the same headend to the same tailend that you configure to carry different CoS values is referred to as a "tunnel bundle." After configuration, Class-Based Tunnel Selection (CBTS) dynamically routes and forwards each packet into the tunnel that:

- Is configured to carry the CoS of the packet
- Has the right headend for the destination of the packet

Because CBTS offers dynamic routing over DS-TE tunnels and requires minimum configuration, it greatly eases deployment of DS-TE in large-scale networks.

CBTS can distribute all CoS values on eight different tunnels.

CBTS also allows the TE tunnels of a tunnel bundle to exit headend routers through different interfaces.

- Prerequisites for MPLS Traffic Engineering Class-based Tunnel Selection, on page 1
- Restrictions for MPLS Traffic Engineering Class-based Tunnel Selection, on page 2
- Information About MPLS Traffic Engineering Class-based Tunnel Selection, on page 2
- How to Configure MPLS Traffic Engineering Class-based Tunnel Selection, on page 9
- Configuration Examples for MPLS Traffic Engineering Class-based Tunnel Selection, on page 17
- Additional References, on page 24
- Feature Information for MPLS Traffic Engineering Class-based Tunnel Selection, on page 25
- Glossary, on page 25

Prerequisites for MPLS Traffic Engineering Class-based Tunnel Selection

- Multiprotocol Label Switching (MPLS) must be enabled on all tunnel interfaces.
- Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled in global configuration mode.

Restrictions for MPLS Traffic Engineering Class-based Tunnel Selection

- For a given destination, all CoS values are carried in tunnels terminating at the same tailend. Either all CoS values are carried in tunnels or no values are carried in tunnels. In other words, for a given destination, you cannot map some CoS values in a DS-TE tunnel and other CoS values in a Shortest Path First (SPF) Label Distribution Protocol (LDP) or SPF IP path.
- CBTS does not allow load-balancing of a given experimental (EXP) value in multiple tunnels. If two or more tunnels are configured to carry a given EXP value, CBTS picks one of those tunnels to carry this EXP value.
- The operation of CBTS is not supported MPLS TE Automesh or label-controlled (LC)-ATM.
- For Any Transport over MPLS (AToM), the operation of CBTS is supported only with Ethernet over MPLS (EoMPLS).
- With Cisco IOS XE Release 3.6S and later releases, you must configure a primary tunnel to make CBTS work. For configuration information, see the “Configuring a Primary Tunnel” section.

Information About MPLS Traffic Engineering Class-based Tunnel Selection

Incoming Traffic Supported by MPLS TE Class-based Tunnel Selection

The CBTS feature supports the following kinds of incoming packets:

- At a provider edge (PE) device—Unlabeled packets that enter a Virtual Private Network (VPN) routing and forwarding (VRF) instance interface
- At a provider core (P) device—Unlabeled and MPLS-labeled packets that enter a non-VRF interface
- At a PE device in a Carrier Supporting Carrier (CSC) or interautonomous system (Inter-AS)—MPLS-labeled packets that enter a VRF interface

CoS Attributes for MPLS TE Class-based Tunnel Selection

CBTS supports tunnel selection based on the value of the EXP field that the headend device imposes on the packet. Before imposing this value, the device considers the input modular quality of service (QoS) command-line interface (CLI) (MQC). If the input MQC modifies the EXP field value, CBTS uses the modified value for its tunnel selection.

Packets may enter the headend from multiple incoming interfaces. These interfaces can come from different customers that have different DiffServ policies. In such cases, service providers generally use input MQC to apply their own DiffServ policies and mark imposed EXP values accordingly. Thus, CBTS can operate consistently for all customers by considering the EXP values marked by the service provider.

Note If the output MQC modifies the EXP field, CBTS ignores the change in the EXP value.

CBTS allows up to eight different tunnels on which it can distribute all classes of service.

Routing Protocols and MPLS TE Class-based Tunnel Selection

CBTS routes and forwards packets to MPLS TE tunnels for specified destinations through use of the following routing protocols:

- Intermediate System-to-Intermediate System (IS-IS) with Autoroute configured
- Open Shortest Path First (OSPF) with Autoroute configured
- Static routing
- Border Gateway Protocol (BGP) with recursion configured on the BGP next hop with packets forwarded on the tunnel through the use of IS-IS, OSPF, or static routing

Tunnel Selection with MPLS TE Class-based Tunnel Selection

This section contains the following topics related to tunnel selection:

EXP Mapping Configuration

With CBTS, you can configure each tunnel with any of the following:

- The same EXP information configured as it was before the CBTS feature was introduced, that is, with no EXP-related information
- One or more EXP values for the tunnel to carry
- A property that allows the carrying of all EXP values not currently allocated to any up-tunnel (default)
- One or more EXP values for the tunnel to carry, and the default property that allows the carrying of all EXP values not currently allocated to any up-tunnel

The default property (the carrying of all EXP values not currently allocated to any up-tunnel) effectively provides a way for the operator to avoid explicitly listing all possible EXP values. Even more important, the default property allows the operator to indicate tunnel preferences onto which to “bump” certain EXP values, should the tunnel carrying those EXP values go down. (See the **tunnel mpls traffic-eng exp** command for the command syntax.)

The configuration of each tunnel is independent of the configuration of any other tunnel. CBTS does not attempt to perform any consistency check for EXP configuration.

This feature allows configurations where:

- Not all EXP values are explicitly allocated to tunnels.
- Multiple tunnels have the default property.
- Some tunnels have EXP values configured and others do not have any values configured.

- A given EXP value is configured on multiple tunnels.

Tunnel Selection for EXP Values

Tunnel selection with this feature is a two-step process:

1. For a given prefix, routing (autoroute, static routes) occurs exactly as it did without the CBTS feature. The device selects the set of operating tunnels that have the best metrics, regardless of the EXP-related information configured on the tunnel.
2. CBTS maps all of the EXP values to the selected set of tunnels.
3. If a given EXP value is configured:
   - On only one of the tunnels in the selected set, CBTS maps the EXP value onto that tunnel.
   - On two or more of the tunnels in the selected set, CBTS arbitrarily maps the EXP value onto one of these tunnels.
4. If a given EXP value is not configured on any of the tunnels in the selected set:
   - And only one of the tunnels in the selected set is configured as a default, CBTS maps the EXP value onto that tunnel.
   - And two or more of the tunnels in the selected set are configured as defaults, CBTS arbitrarily maps the EXP value onto one of these tunnels.
   - And no tunnel in the selected set of tunnels is configured as a default, CBTS arbitrarily maps the EXP value onto one of these tunnels.

CBTS relies on autoroute to select the tunnel bundle. Autoroute selects only tunnels that are on the SPF to the destination. Therefore, similar to Autoroute, CBTS does not introduce any risk of routing loops.

Tunnel Selection Examples

The following examples show various tunnel configurations that are set up by an operator and indicate how CBTS maps packets carrying EXP values onto these tunnels. Each example describes a different configuration: a default tunnel configured, more than one tunnel configured with the same EXP value, and so on.

Example 1—Default Tunnel Configured

An operator configures the following parameters on tunnels T1 and T2:

- T1: exp = 5
- T2: exp = default

If T1 and T2 are next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1
- Packets with <Dest = P, exp = anything-other-than-5> onto T2

Example 2—EXP Values Configured on Two Tunnels; One Default Tunnel

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5

- T2: exp = 3 and 4
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1
- Packets with <Dest = P, exp = 3 or 4> onto T2
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3

Example 3—More than One Tunnel with the Same EXP

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 5
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (arbitrary selection)
- Packets with <Dest = P, exp = anything-other-than-5> onto T3
- No packets onto T2

Example 4—Static Route Configured

An operator configures the following parameters on tunnels T1 and T2:

- T1: exp = 5
- T2: exp = 3
- Static route to P on T2

If prefix P is behind the T1 and T2 tailend device, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = anything> onto T2
- No packets onto T1

Static routes are preferred over dynamic routes; therefore, the device chooses only T2 as the "selected set" of tunnels.

Example 5—No Default or Metric Configuration

An operator configures the following parameters on tunnels T1 and T2:

- T1: exp = 5
- T2: exp = 3

If T1 and T2 are the next-hop interfaces for prefix P, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1
- Packets with <Dest = P, exp = 3> onto T2
- Packets with <Dest = P, exp = anything-other-than-3-or-5> onto T2

If a packet arrives with an EXP value that is different from any value configured for a tunnel, the packet goes in to the default tunnel. If no default tunnel is configured, the packet goes in to the tunnel that is arbitrarily selected by CBTS.

Multipath with Non-TE Paths and MPLS TE Class-Based Tunnel Selection

For a given prefix in the routing process, the device might select a set of paths that includes both TE tunnels and non-TE-tunnel paths (SPF paths). For example, internal Border Gateway Protocol (iBGP) Multipath might be activated and result in multiple BGP next hops for that prefix, where one BGP next hop is reachable through TE tunnels and other BGP next hops are reachable through non-TE-tunnel paths.

An equal cost IGP path might also exist over TE tunnels and over a non-TE tunnel path. For example, a TE tunnel metric might be modified to be equal to the SPF path.

In these situations, CBTS maps traffic in the following manner:

- If a given EXP value is configured on one or more of the tunnels in the selected set, CBTS maps the EXP value onto that tunnel or one of those tunnels.
- If a given EXP value is not configured on any of the tunnels in the selected set but one or more of the tunnels is configured as a default in the selected set, then CBTS maps the EXP value onto that tunnel or one of those tunnels.
- If a given EXP value is not configured on any of the tunnels from the selected set and no tunnel in the selected set is configured as a default, CBTS arbitrarily maps the EXP value onto one of the tunnels in the selected set, and performs CoS-unaware load-balancing with other non-TE paths.
- If the routing process allocates all EXP values to tunnels or if a default is used, then routing does not use the non-TE paths unless all TE tunnels are down.

MPLS TE Class-Based Tunnel Selection and Policy-Based Routing

If you configure both policy-based routing (PBR) over TE tunnels (in non-VRF environments) and CBTS, the PBR decision overrides the CBTS decision. PBR is an input process that the device performs ahead of regular forwarding.

Tunnel Failure Handling

For CBTS operation, the important question is whether the tunnel interface is up or down, not whether the current TE label switched path (LSP) is up or down. For example, a TE LSP might go down but is reestablished by the headend because another path option exists. The tunnel interface does not go down during the transient period while the TE LSP is reestablished. Because the tunnel interface does not go down, the corresponding EXP does not get rerouted onto another tunnel during the transient period.

When a tunnel used by CBTS for forwarding goes down, the feature adjusts its tunnel selection for the affected EXP values. It reapplies the tunnel selection algorithm to define the behavior of packets for all EXP values, as shown in the examples that follow.

Example 1—Tunnel Other than the Default Tunnel Goes Down

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 3 and 4
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P and Tunnel T1 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 3, 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3 (as before)
- Packets with <Dest = P, exp = 5> onto T3

Example 2—Default Tunnel Goes Down

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 3 and 4
- T3: exp = default

If T1, T2, and T3 are next-hop interfaces for prefix P and Tunnel T3 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (as before)
- Packets with <Dest = P, exp = 3, 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T1

Example 3—Two Default Tunnels Are Configured

An operator configures the following parameters on tunnels T1, T2, and T3:

- T1: exp = 5
- T2: exp = 3, 4, and default
- T3: exp = 0, 1, 2, 6, 7, and default

If T1, T2, and T3 are next-hop interfaces for prefix P and Tunnel T3 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (as before)
- Packets with <Dest = P, exp = 3, 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T2

If tunnel T2 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 5> onto T1 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3 (as before)
- Packets with <Dest = P, exp = 3, or 4> onto T3

If tunnel T1 goes down, CBTS maps the packets onto the tunnels in this way:

- Packets with <Dest = P, exp = 3, or 4> onto T2 (as before)
- Packets with <Dest = P, exp = 0, 1, 2, 6, or 7> onto T3 (as before)
- Packets with <Dest = P, exp = 5> onto either T2 or T3, but not both

In Example 3, the operator configures the EXP default option on two tunnels to ensure that nonvoice traffic is never redirected onto the voice tunnel (T1).

Misordering of Packets

In DiffServ, packets from a given flow might get marked with EXP values that are different from each other but belong to the same CoS value because of in-contract and out-of-contract marking of packets. We can refer to these values of EXP bits as EXP-in and EXP-out.

If packets for EXP-in are sent on a different tunnel than packets for EXP-out, then misordering of packets within the same flows could occur. For that reason, CBTS allows operators to ensure that EXP-in and EXP-out never get mapped onto different tunnels.

The CBTS feature allows the operator to configure EXP-in and EXP-out to be transported on the same tunnel when that tunnel is up. This ensures that the feature does not introduce misordering of packets. In case of tunnel failure, the tunnel selection algorithm ensures that if EXP-in and EXP-out were carried on the same tunnel before the failure, they are still carried on a single tunnel after the failure. Thus, CBTS protects against nontransient misordering even in the event of tunnel failure.



Note CBTS does not attempt to force EXP-in and EXP-out to be carried on the same tunnel. The operator must configure CBTS so that EXP-in and EXP-out are carried on the same tunnel. This is comparable to the regular DiffServ situation, where the operator must ensure that EXP-in and EXP-out are configured to go in the same queue.

Fast Reroute and MPLS TE Class-based Tunnel Selection

CBTS allows Fast Reroute (FRR) protection on tunnels for which you configure CoS-based selection.



Note You cannot configure FRR on a primary tunnel.

CBTS operation with FRR does not change the number of or the way in which FRR backup tunnels might be used. The operation of FRR is the same as when CBTS is not activated. After you configure primary tunnels from a given headend to a given tailend, you can use FRR in the same way whether you activate CoS-based tunnel selection or not. This includes the following possibilities:

- None of the tunnels use FRR.

- All of the *x* tunnels are FRR-protected and share the same backup tunnel, if the traffic goes out the same interface.
- Some of the *x* tunnels are not FRR-protected; the remaining tunnels are FRR-protected and share the same backup tunnel, if the traffic goes out the same interface.
- Some of the *x* tunnels are not FRR-protected; the remaining tunnels are FRR-protected and are protected by different backup tunnels (for example, if the traffic goes out different interfaces, or if the traffic goes out the same interface). Bandwidth guarantees exist on the backup tunnels.

The important question for CBTS operation is only whether a tunnel interface goes down or stays up. FRR protects a given tunnel in exactly the same way as if CBTS were not configured on the tunnel.

DS-TE Tunnels and MPLS TE Class-based Tunnel Selection

CBTS operates over tunnels using DS-TE. Therefore, the tunnels on which CoS-based selection is performed can each arbitrarily and independently use a bandwidth from the global pool or the subpool.

Reoptimization and MPLS TE Class-based Tunnel Selection

CBTS allows tunnels on which CoS-based selection is performed to be reoptimized. Reoptimization does not affect CBTS operation.

Interarea and Inter-AS and MPLS TE Class-based Tunnel Selection

The CBTS operates over tunnels that are interarea when the interarea tunnels use static routes on destination prefixes or on the BGP next hops.

ATM PVCs and MPLS TE Class-based Tunnel Selection

CBTS operates over ATM permanent virtual circuits (PVCs). This means that TE or DS-TE tunnels handled by CBTS can span links that are ATM PVCs. ATM PVCs might be used on the headend device that is running CBTS and on transit label switch routers (LSRs).

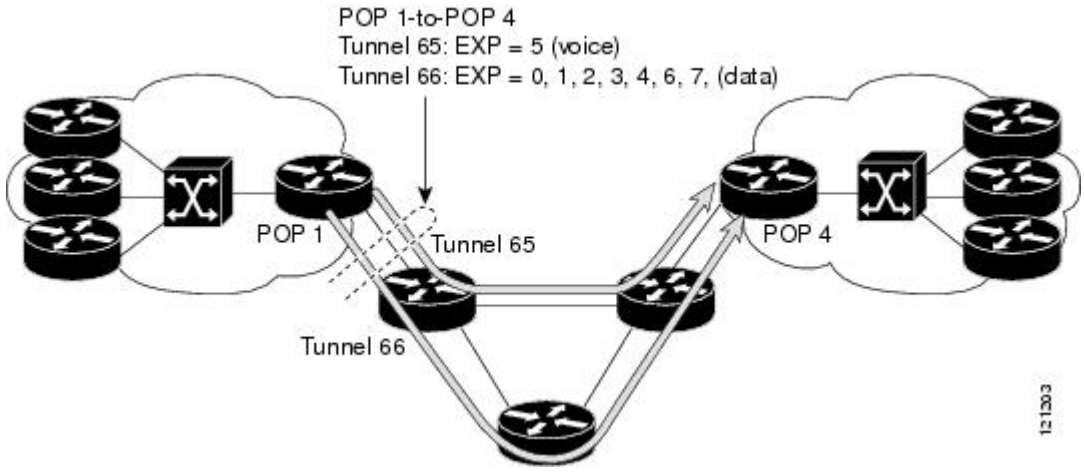
How to Configure MPLS Traffic Engineering Class-based Tunnel Selection

You need to configure the CBTS feature only on the tunnel headend. No CBTS configuration is required on the tailend or transit LSR.

Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend

The figure below shows an example of two tunnels, Tunnel 65 and Tunnel 66, transporting different classes of traffic between the same headend and the same tailend.

Figure 1: Tunnels Transporting Different Classes of Service Between the Same Headend and Tailend



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** {*hostname* | *ip-address*}
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** [**sub-pool** | **global**] *bandwidth*
8. **exit**
9. Repeat steps 3 through 8 on the same headend device to create additional tunnels from this headend to the same tailend.
10. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 65`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 5	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Device(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Device(config-if)# tunnel mode mpls traffic-eng`	Sets the mode of a tunnel to MPLS for TE.
Step 7	**tunnel mpls traffic-eng bandwidth** [**sub-pool** | **global**] *bandwidth* **Example:** `Device(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 3000`	Configures the bandwidth for the MPLS TE tunnel. If automatic bandwidth is configured for the tunnel, use the **tunnel mpls traffic-eng bandwidth** command to configure the initial tunnel bandwidth, which is adjusted by the autobandwidth mechanism. **Note** You can configure any existing MPLS TE command on these TE or DS-TE tunnels.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 9	Repeat steps 3 through 8 on the same headend device to create additional tunnels from this headend to the same tailend.	--
Step 10	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring EXP Values to Be Carried by Each MPLS TE or DS-TE Tunnel

For each tunnel that you create, you must indicate which EXP values the tunnel carries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel mpls traffic-eng eng** [*list-of-exp-values*] [**default**]
5. **exit**
6. Repeat steps 3 through 5 for all MPLS TE tunnels that you created in the Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend, on page 9.
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface tunnel 65`	Configures an interface type and enters interface configuration mode.
Step 4	**tunnel mpls traffic-eng eng** [*list-of-exp-values*] [**default**] **Example:** `Device(config-if)# tunnel mpls traffic-eng exp 5`	Specifies the EXP bits that will be forwarded over a member tunnel that is part of the CBTS bundle.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 6	Repeat steps 3 through 5 for all MPLS TE tunnels that you created in the Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend, on page 9.	--
Step 7	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

Verifying That the MPLS TE or DS-TE Tunnels Are Operating and Announced to the IGP

SUMMARY STEPS

1. **show mpls traffic-eng topology** {*ip-address* | **igp-id** {**isis** *nsap-address* | **ospf** *ip-address*} [**brief**]
2. **show mpls traffic-eng tunnels** *number* [**brief**] [**protection**]
3. **show ip cef summary**
4. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface*| **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
5. **show mpls traffic-eng autoroute**

DETAILED STEPS

Procedure

Step 1 **show mpls traffic-eng topology** {*ip-address* | **igp-id** {**isis** *nsap-address* | **ospf** *ip-address*} [**brief**]

Use this command to display the MPLS TE global topology currently known at this node:

Example:

```
Device# show mpls traffic-eng topology
My_System_id: 0000.0025.0003.00

IGP Id: 0000.0024.0004.00, MPLS TE Id:172.16.4.4 Router Node
      link[0 ]:Intf Address: 10.1.1.4
                  Nbr IGP Id: 0000.0024.0004.02,
                admin_weight:10, affinity_bits:0x0
                max_link_bw:10000 max_link_reservable: 10000
   globalpool subpool
                total allocated reservable   reservable
                ---------------  ----------  ----------
 bw[0]:  0  1000 500
 bw[1]: 10  990 490
 bw[2]:  600  390 390
 bw[3]:  0  390 390
 bw[4]:  0  390 390
 bw[5]:  0  390 390
```

Step 2 **show mpls traffic-eng tunnels** *number* [**brief**] [**protection**]

Use this command to display information for a specified tunneling interface:

Example:

```
Device# show mpls traffic-eng tunnels 500 brief protection
```

```
Device#_t500
  LSP Head, Tunnel500, Admin: up, Oper: up
  Src 172.16.0.5, Dest 172.16.0.8, Instance 17
  Fast Reroute Protection: None
  Path Protection: 1 Common Link(s) , 1 Common Node(s)
    Primary lsp path:192.168.6.6 192.168.7.7
                     192.168.8.8 192.168.0.8
    Protect lsp path:172.16.7.7 192.168.8.8
                     10.0.0.8
    Path Protect Parameters:
      Bandwidth: 50       kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
      Metric Type: TE (default)
    InLabel  :  -
    OutLabel : Serial5/3, 46
    RSVP Signalling Info:
         Src 172.16.0.5, Dst 172.16.0.8, Tun_Id 500, Tun_Instance 18
      RSVP Path Info:
        My Address: 172.16.0.5
        Explicit Route: 192.168.7.7 192.168.8.8
        Record   Route:   NONE
        Tspec: ave rate=50 kbits, burst=1000 bytes, peak rate=50 kbits
      RSVP Resv Info:
        Record   Route:   NONE
        Fspec: ave rate=50 kbits, burst=1000 bytes, peak rate=50 kbits
```

Step 3 **show ip cef summary**

Use this command to display a summary of the IP CEF table:

Example:

```
Device# show ip cef summary
IP Distributed CEF with switching (Table Version 25), flags=0x0
 21 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 1
 21 leaves, 16 nodes, 19496 bytes, 36 inserts, 15 invalidations
 0 load sharing elements, 0 bytes, 0 references
 universal per-destination load sharing algorithm, id 5163EC15
 3(0) CEF resets, 0 revisions of existing leaves
 Resolution Timer: Exponential (currently 1s, peak 1s)
 0 in-place/0 aborted modifications
 refcounts:  4377 leaf, 4352 node
 Table epoch: 0 (21 entries at this epoch)
 Adjacency Table has 9 adjacencies
```

Step 4 **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface*| **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]

Use this command to display the contents of the MPLS Label Forwarding Information Base (LFIB):

Example:

```
Device# show mpls forwarding-table
Local Outgoing      Prefix            Bytes tag Outgoing       Next Hop
Label Label or VC   or Tunnel Id      switched  interface
26    No Label      10.253.0.0/16     0         Et4/0/0        10.27.32.4
28    1/33          10.15.0.0/16      0         AT0/0.1        point2point
29    Pop Label     10.91.0.0/16      0         Hs5/0          point2point
      1/36          10.91.0.0/16      0         AT0/0.1        point2point
30    32            10.250.0.97/32    0         Et4/0/2        10.92.0.7
      32            10.250.0.97/32    0         Hs5/0          point2point
34    26            10.77.0.0/24      0         Et4/0/2        10.92.0.7
      26            10.77.0.0/24      0         Hs5/0          point2point
35    No Label[T]   10.100.100.101/32 0         Tu301          point2point
```

```
36    Pop Label     10.1.0.0/16      0         Hs5/0         point2point
      1/37          10.1.0.0/16      0         AT0/0.1       point2point
[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

Step 5 **show mpls traffic-eng autoroute**

Use this command to display tunnels that are announced to the IGP, including interface, destination, and bandwidth:

Example:

```
Device# show mpls traffic-eng autoroute
MPLS TE autorouting enabled
destination 10.0.0.9, area ospf 10  area 0, has 4 tunnels
    Tunnel1     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel2     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel3     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel4     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
```

Configuring a Primary Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** {*hostname* | *ip-address*}
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng autoroute announce**
8. **tunnel mpls traffic-eng exp-bundle master**
9. **tunnel mpls traffic-eng exp-bundle member** *tunnel-number*
10. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 65`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Device(config-if)# ip unnumbered loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 5	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Device(config-if)# tunnel destination 10.10.10.12`	Specifies the destination of the tunnel for this path option.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Device(config-if)# tunnel mode mpls traffic-eng`	Sets the mode of a tunnel to MPLS for TE.
Step 7	**tunnel mpls traffic-eng autoroute announce** **Example:** `Device(config-if)# tunnel mpls traffic-eng autoroute announce`	Specifies that the IGP should use the tunnel in its enhanced SPF calculation if the tunnel is up
Step 8	**tunnel mpls traffic-eng exp-bundle master** **Example:** `Device(config-if)# tunnel mpls traffic-eng exp-bundle master`	Configures a primary tunnel.
Step 9	**tunnel mpls traffic-eng exp-bundle member** *tunnel-number* **Example:** `Device(config-if)# tunnel mpls traffic-eng exp-bundle member tunnel1`	Identifies which tunnel is a member of a primary tunnel.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.

Configuration Examples for MPLS Traffic Engineering Class-based Tunnel Selection

Example: Creating Multiple MPLS TE or DS-TE Tunnels from the Same Headend to the Same Tailend

The following example shows how to create multiple MPLS TE or DS-TE tunnels from the same headend to the same tailend:

```
Device(config)# interface Tunnel 65

Device(config-if)# ip numbered loopback 0
Device(config-if)# tunnel destination 10.1.1.1

Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000

Device(config-if)# ^Z
Device(config)# interface Tunnel 66

Device(config-if)# ip numbered loopback 0
Device(config-if)# tunnel destination 10.1.1.1

Device(config-if)# tunnel mode mpls traffic-eng
Device(config-if)# tunnel mpls traffic-eng bandwidth 50000
Device(config-if)# end
Device#
```

Example: Configuring EXP Values to Be Carried by Each MPLS TE or DS-TE Tunnel

The following example shows how to configure EXP values to be carried by each MPLS TE or DS-TE tunnel that you created:

```
Device(config)# interface Tunnel 65

Device(config-if)# tunnel mpls traffic-eng exp 5
Device(config-if)# ^Z
Device(config)#
Device(config)# interface Tunnel 66

Device(config-if)# tunnel mpls traffic-eng exp 0 1 2 3 4 6 7
Device(config-if)# end
Device#
```

Example: Verifying That the MPLS TE or DS-TE Tunnels Are Operating and Announced to the IGP

The output for each of the following examples helps verify that the MPLS TE or DS-TE tunnels are operating and visible.

The **show mpls traffic-eng topology** command output displays the MPLS TE global topology:

```
Device# show mpls traffic-eng topology 10.0.0.1
IGP Id: 10.0.0.1, MPLS TE Id:10.0.0.1 Router Node  (ospf 10  area 0) id 1
      link[0]: Broadcast, DR: 10.0.1.2, nbr_node_id:6, gen:18
   frag_id 0, Intf Address:10.1.1.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                          Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:            0             1000               0
 bw[1]:            0             1000               0
 bw[2]:            0             1000               0
 bw[3]:            0             1000               0
 bw[4]:            0             1000               0
 bw[5]:            0             1000               0
 bw[6]:            0             1000               0
 bw[7]:          100              900               0
      link[1]: Broadcast, DR: 10.0.2.2, nbr_node_id:7, gen:19
   frag_id 1, Intf Address:10.0.2.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                          Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:            0             1000               0
 bw[1]:            0             1000               0
 bw[2]:            0             1000               0
 bw[3]:            0             1000               0
 bw[4]:            0             1000               0
 bw[5]:            0             1000               0
 bw[6]:            0             1000               0
 bw[7]:          300              700               0
Device#
Device# show mpls traffic-eng topology 10.0.0.9
IGP Id: 10.0.0.9, MPLS TE Id:10.0.0.9 Router Node  (ospf 10  area 0) id 3
      link[0]: Point-to-Point, Nbr IGP Id: 10.0.0.5, nbr_node_id:5, gen:9
   frag_id 1, Intf Address:10.0.5.2, Nbr Intf Address:10.0.5.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 155000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                          Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:            0             1000               0
 bw[1]:            0             1000               0
```

```
 bw[2]:            0             1000               0
 bw[3]:            0             1000               0
 bw[4]:            0             1000               0
 bw[5]:            0             1000               0
 bw[6]:            0             1000               0
 bw[7]:            0             1000               0
      link[1]: Point-to-Point, Nbr IGP Id: 10.0.0.7, nbr_node_id:4, gen:9
   frag_id 0, Intf Address:10.0.6.2, Nbr Intf Address:10.0.6.1
   TE metric:1, IGP metric:1, attribute_flags:0x0
   SRLGs: None
   physical_bw: 155000 (kbps), max_reservable_bw_global: 1000 (kbps)
   max_reservable_bw_sub: 0 (kbps)
                       Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
        ---------------   -----------       ----------
 bw[0]:            0             1000               0
 bw[1]:            0             1000               0
 bw[2]:            0             1000               0
 bw[3]:            0             1000               0
 bw[4]:            0             1000               0
 bw[5]:            0             1000               0
 bw[6]:            0             1000               0
 bw[7]:            0             1000               0
Device#
```

The **show mpls traffic-eng tunnels** command output displays information about a tunnel:

```
Device# show mpls traffic-eng tunnels tunnel1
Name: Router_t1                           (Tunnel1) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path1 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/0, 12304
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 1, Tun_Instance 10
    RSVP Path Info:
      My Address: 10.0.1.1
      Explicit Route: 10.0.1.2 10.0.3.2 10.0.5.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 180.0.2.2 10.0.3.2 180.0.5.2
                    10.0.0.9
  History:
    Tunnel:
      Time since created: 15 minutes, 18 seconds
      Time since path change: 15 minutes, 5 seconds
    Current LSP:
      Uptime: 15 minutes, 5 seconds
Device# show mpls traffic-eng tunnel tunnel2
```

```
Name: Router_t2                           (Tunnel2) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path2 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/1, 12305
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 2, Tun_Instance 10
    RSVP Path Info:
      My Address: 10.0.2.1
      Explicit Route: 10.0.2.2  10.0.4.2  10.0.6.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 10.0.2.2 10.0.3.2 10.0.5.2
                    10.0.0.9
  History:
    Tunnel:
      Time since created: 15 minutes, 19 seconds
      Time since path change: 15 minutes, 6 seconds
    Current LSP:
      Uptime: 15 minutes, 6 seconds
Device# show mpls traffic-eng tunnels tunnel3
Name: Router_t3                           (Tunnel3) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path2 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/1, 12306
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 3, Tun_Instance 8
    RSVP Path Info:
      My Address: 10.0.2.1
'     Explicit Route: 10.0.2.2 10.0.4.2 10.0.6.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 10.0.2.2 10.0.3.2 10.0.5.2
                    10.0.0.9
  History:
```

```
    Tunnel:
      Time since created: 15 minutes, 19 seconds
      Time since path change: 15 minutes, 7 seconds
    Current LSP:
      Uptime: 15 minutes, 7 seconds
Device# show mpls traffic-eng tunnels tunnel4
Name: Router_t4                           (Tunnel4) Destination: 10.0.0.9
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit path2 (Basis for Setup, path weight 3)
  Config Parameters:
    Bandwidth: 100      kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 100      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: explicit path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
  InLabel  :  -
  OutLabel : FastEthernet6/1, 12307
  RSVP Signalling Info:
       Src 10.0.0.1, Dst 10.0.0.9, Tun_Id 4, Tun_Instance 6
    RSVP Path Info:
      My Address: 10.0.2.1
      Explicit Route: 10.0.2.2 10.0.4.2 10.0.6.2 10.0.0.9
      Record   Route:   NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=17179869 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 3 (TE)
    Explicit Route: 10.0.2.1 10.0.2.2 10.0.3.2 10.0.5.2
                    10.0.0.9
  History:
    Tunnel:
      Time since created: 15 minutes, 20 seconds
      Time since path change: 15 minutes, 8 seconds
    Current LSP:
      Uptime: 15 minutes, 8 seconds
```

The **show ip cef detail** command output displays detailed FIB entry information for a tunnel:

```
Device# show ip cef tunnel1 detail
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel1, 0 dependencies
```

```
    traffic share 1
    next hop 0.0.0.0, Tunnel1
    valid adjacency
    tag rewrite with Tu1, point2point, tags imposed {12304}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
Device# show ip cef tunnel2 detail
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel2, 0 dependencies
    traffic share 1
    next hop 0.0.0.0, Tunnel2
    valid adjacency
    tag rewrite with Tu2, point2point, tags imposed {12305}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
Device# show ip cef tunnel3 detail
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel3, 0 dependencies
    traffic share 1
    next hop 0.0.0.0, Tunnel3
    valid adjacency
    tag rewrite with Tu3, point2point, tags imposed {12306}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
Device# show ip cef tunnel4 detail
```

```
IP CEF with switching (Table Version 46), flags=0x0
  31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
  2 instant recursive resolutions, 0 used background process
  8 load sharing elements, 8 references
  6 in-place/0 aborted modifications
  34696 bytes allocated to the FIB table data structures
  universal per-destination load sharing algorithm, id 9EDD49E1
  1(0) CEF resets
  Resolution Timer: Exponential (currently 1s, peak 1s)
  Tree summary:
   8-8-8-8 stride pattern
   short mask protection disabled
   31 leaves, 23 nodes using 26428 bytes
  Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, all rewrites inherited
    local tag: tunnel head
  via 0.0.0.0, Tunnel4, 0 dependencies
    traffic share 1
    next hop 0.0.0.0, Tunnel4
    valid adjacency
    tag rewrite with Tu4, point2point, tags imposed {12307}
  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
```

The **show mpls forwarding-table detail** command output displays detailed information from the MPLS LFIB:

```
Device# show mpls forwarding-table detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
Device#
Device# show mpls forwarding-table 10.0.0.9 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
Tun hd Untagged    10.0.0.9/32       0          Tu1        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12304}, via Fa6/0
 00027D884000000ED70178A88847 03010000
 No output feature configured
        Per-exp selection: 1
       Untagged    10.0.0.9/32       0          Tu2        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12305}, via Fa6/1
 00027D884001000ED70178A98847 03011000
 No output feature configured
        Per-exp selection: 2  3
       Untagged    10.0.0.9/32       0          Tu3        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12306}, via Fa6/1
 00027D884001000ED70178A98847 03012000
 No output feature configured
        Per-exp selection: 4  5
       Untagged    10.0.0.9/32       0          Tu4        point2point
 MAC/Encaps=14/18, MRU=1500, Tag Stack{12307}, via Fa6/1
 00027D884001000ED70178A98847 03013000
 No output feature configured
        Per-exp selection: 0  6  7
Device#
```

The **show mpls traffic-eng autoroute** command output displays tunnels that are announced to the IGP:

```
Device# show mpls traffic-eng autoroute
```

```
MPLS TE autorouting enabled
  destination 10.0.0.9, area ospf 10  area 0, has 4 tunnels
    Tunnel1     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel2     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel3     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
    Tunnel4     (load balancing metric 20000000, nexthop 10.0.0.9)
                (flags: Announce)
Device#
```

Example: Configuring a Primary Tunnel

The following example specifies that there is a primary tunnel that includes tunnels Tunnel20000 through Tunnel20005:

```
interface Tunnel 200
 ip unnumbered Loopback 0
 tunnel destination 10.10.10.10
 tunnel mode mpls traffic-eng
 tunnel mode mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng exp-bundle master
 tunnel mpls traffic-eng exp-bundle member Tunnel20000
 tunnel mpls traffic-eng exp-bundle member Tunnel20001
 tunnel mpls traffic-eng exp-bundle member Tunnel20002
 tunnel mpls traffic-eng exp-bundle member Tunnel20003
 tunnel mpls traffic-eng exp-bundle member Tunnel20004
 tunnel mpls traffic-eng exp-bundle member Tunnel20005
```

Additional References

Related Documents

Related Topic	Document Title
MPLS traffic engineering commands	*Multiprotocol Label Switching Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS Traffic Engineering Class-based Tunnel Selection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for MPLS Traffic Engineering Class-based Tunnel Selection

Feature Name	Releases	Feature Configuration Information
MPLS Traffic Engineering : Class-based Tunnel Selection	12.0(29)S 12.2(33)SRA 12.2(32)SY 12.2(33)SXH 12.4(20)T Cisco IOS XE Release 3.6S	The MPLS Traffic Engineering (TE): Class-based Tunnel Selection feature enables you to dynamically route and forward traffic with different class of service (CoS) values onto different TE tunnels between the same tunnel headend and the same tailend. The TE tunnels can be regular TE or DiffServ-aware TE (DS-TE) tunnels. In 12.0(29)S, this feature was introduced. In 12.2(33)SRA, this feature was integrated and the following commands were added: • **tunnel mpls traffic-eng exp-bundle master** • **tunnel mpls traffic-eng exp-bundle member** • **show mpls traffic-eng exp** 12.0(32)SY, support for this feature was added on the Cisco 12000 family of routers. In 12.2(33)SXH, this feature was integrated. In 12.4(20)T, this feature was integrated. In Cisco IOS XE Release 3.6S, this feature was integrated.

Glossary

BGP --Border Gateway Protocol. Interdomain routing protocol that replaces External Gateway Protocol (EGP). BGP exchanges reachability information with other BGP systems. It is defined by RFC 116.3

bundled tunnels--Members of a primary tunnel. You define the EXP bits that will be forwarded over each bundled tunnel.

Cisco Express Forwarding--An advanced Layer 3 IP switching technology. Cisco Express Forwarding optimizes network performance and scalability for networks with large and dynamic traffic patterns, such as the Internet and networks characterized by intensive web-based applications or interactive sessions.

CoS --class of service. An indication of how an upper-layer protocol requires a lower-layer protocol to treat its messages. In Systems Network Architecture (SNA) subarea routing, CoS definitions are used by subarea nodes to determine the optimal route for establishing a given session. A CoS definition comprises a virtual route number and a transmission priority field. Also called type of service (ToS).

DS-TE --DiffServ-aware traffic engineering. The configuring of two bandwidth pools on each link, a global pool and a subpool. Multiprotocol Label Switching (MPLS) traffic engineering tunnels using the subpool bandwidth can be configured with quality of service (QoS) mechanisms to deliver guaranteed bandwidth services end-to-end across the network. Simultaneously, tunnels using the global pool can convey DiffServ traffic.

EXP --experimental field or bits. A 3-bit field in the Multiprotocol Label Switching (MPLS) header widely known as the EXP field or EXP bits because, according to RFC 3032, that field is reserved for experimental use. However, the most common use of those bits is for quality of service (QoS) purposes.

headend --The upstream, transmitting end of a tunnel. This is the first device in the label switched path (LSP).

LSP --label switched path. A sequence of hops (R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A label switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

master tunnel--A set of tunnels that have the same destination.

MPLS traffic engineering--Multiprotocol Label Switching traffic engineering. A constraint-based routing algorithm for routing label switched path (LSP) tunnels.

MQC --modular quality of service (QoS) command-line interface (CLI). A CLI structure that allows users to create traffic polices and attach those polices to interfaces.

PBR --policy-based routing. A routing scheme in which packets are forwarded to specific interfaces based on user-configured policies. A policy might specify, for example, that traffic sent from a particular network should be forwarded out one interface, and all other traffic should be forwarded out another interface.

tailend --The downstream, receiving end of a tunnel. The device that terminates the traffic engineering label switched path (LSP).

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

ToS --type of service. See CoS.

tunnel --A secure communication path between two peers. A traffic engineering tunnel is a label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.