



Configuring MPLS TE Class-Based Tunnel Selection

This chapter describes how to configure Multiprotocol Label Switching (MPLS) traffic engineering (TE) class-based tunnel selection (CBTS) on Cisco NX-OS devices.

This chapter includes the following sections:

- [Finding Feature Information, page 18-248](#)
- [Information About MPLS TE Class-Based Tunnel Selection, page 18-248](#)
- [Licensing Requirements for MPLS TE Class-Based Tunnel Selection, page 18-256](#)
- [Prerequisites for MPLS TE Class-Based Tunnel Selection, page 18-256](#)
- [Guidelines and Limitations for MPLS TE Class-Based Tunnel Selection, page 18-257](#)
- [Configuring MPLS TE Class-Based Tunnel Selection, page 18-257](#)
- [Verifying the MPLS TE Class-Based Tunnel Selection Configuration, page 18-260](#)
- [Configuration Examples for MPLS TE Class-Based Tunnel Selection, page 18-262](#)
- [Additional References MPLS TE Class-Based Tunnel Selection, page 18-268](#)
- [Feature History for MPLS TE Class-Based Tunnel Selection, page 18-268](#)

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at <https://tools.cisco.com/bugsearch/> and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” chapter or the Feature History table below.

Information About MPLS TE Class-Based Tunnel Selection

CBTS 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, CBTS dynamically routes and forwards each packet into the tunnel that does the following:

- Carries the CoS of the packet
- Contains 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.

This section includes the following topics:

- [Incoming Traffic Supported by Class-Based Tunnel Selection, page 18-249](#)
- [CoS Attributes for Class-Based Tunnel Selection, page 18-249](#)
- [Routing Protocols and Class-Based Tunnel Selection, page 18-250](#)
- [Tunnel Selection with Class-Based Tunnel Selection, page 18-250](#)
- [Fast Reroute and Class-Based Tunnel Selection, page 18-255](#)
- [DS-TE Tunnels and Class-Based Tunnel Selection, page 18-256](#)
- [Reoptimization and Class-Based Tunnel Selection, page 18-256](#)

Incoming Traffic Supported by Class-Based Tunnel Selection

CBTS supports the following types of incoming packets:

- At a provider edge (PE) router—Unlabeled packets that enter a virtual private network (VPN) routing and forwarding (VRF) instance interface
- At a provider core (P) router—Unlabeled and MPLS-labeled packets that enter a non-VRF interface

CoS Attributes for Class-Based Tunnel Selection

CBTS supports tunnel selection based on the value of the EXP field that the headend router imposes on the packet. Before imposing this value, the router 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, operators use input MQC to apply their own DiffServ policies and mark imposed EXP values accordingly. CBTS can operate consistently for all customers by considering the EXP values marked by the operator.



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 Class-Based Tunnel Selection

CBTS routes and forwards packets to MPLS TE tunnels for specified destinations through 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 next hop with packets forwarded on the tunnel through IS-IS, OSPF, or static routing

Tunnel Selection with Class-Based Tunnel Selection

This section includes the following topics:

- [EXP Mapping Configuration, page 18-250](#)
- [Tunnel Selection for EXP Values, page 18-251](#)
- [Tunnel Failure Handling, page 18-253](#)
- [Misordering of Packets, page 18-255](#)

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 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 (all EXP values not currently allocated to any up-tunnel) enables you to avoid explicitly listing all possible EXP values. The default property also enables you to indicate tunnel preferences on which to force certain EXP values, if the tunnel carrying those EXP values goes down.

You can configure each tunnel independently of any other tunnel. CBTS does not try to perform any consistency check for EXP configuration.

This feature allows configurations for the following situations:

- 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

This section includes the following topics:

- [Tunnel Selection Process, page 18-251](#)
- [Tunnel Selection Examples, page 18-251](#)
- [Multipath with Non-TE Paths and MPLS TE CBTS, page 18-253](#)
- [MPLS TE Class-Based Tunnel Selection and Policy-Based Routing, page 18-253](#)

Tunnel Selection Process

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 router 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:
 - If a given EXP value is configured, CBTS does the following:
 - If only one of the tunnels is configured in the selected set, CBTS maps the EXP value onto that tunnel.
 - If two or more of the tunnels in the selected set are configured, CBTS arbitrarily maps the EXP value onto one of these tunnels. CBTS selects the tunnel on which the lowest EXP value is explicitly configured and picks the tunnel that has the lowest tunnel ID.
 - If a given EXP value is not configured on any of the tunnels in the selected set, CBTS does the following:
 - If only one of the tunnels in the selected set is configured as a default, CBTS maps the EXP value onto that tunnel.
 - If 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.
 - If no tunnel in the selected set of tunnels is configured as a default, CBTS does not map the EXP value onto any specific tunnel. Instead, CBTS performs CoS-unaware load balancing of that EXP information across all tunnels in the selected set.

CBTS relies on autoroute, which selects only tunnels that are on the Shortest Path First (SPF) to the destination. Therefore, CBTS does not introduce any risk of routing loops.

Tunnel Selection Examples

The following examples show various tunnel configurations and indicate how CBTS maps packets that carry EXP values onto these tunnels. Each example describes a different configuration.

Example 1—Default Tunnel Configured

The following example shows the following parameters on tunnels T1 and T2:

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

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

- 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

The following example shows the following parameters on tunnels T1, T2, and T3:

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

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

- 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

The following example shows the following parameters on tunnels T1, T2, and T3:

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

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

- 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

The following example shows the following parameters on tunnels T1 and T2:

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

If prefix P is behind the T1 and T2 tailend router, CBTS maps the packets onto the tunnels as follows:

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

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

Example 5—Metrics Configured on Tunnels

The following example shows the following parameters on tunnels T1 and T2:

- T1: exp = 5, autoroute, relative metric -2
- T2: exp = 3, autoroute, relative metric -3

CBTS maps the packets onto the tunnels as follows:

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

The autoroute tunnel selection algorithm selects the tunnel with the best metric; therefore, the router selects only T2 as the selected set of tunnels.

Example 6—No Default or Metric Configuration

The following example shows the following parameters on tunnels T1 and T2:

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

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

- 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 configured with the lowest EXP value.

Multipath with Non-TE Paths and MPLS TE CBTS

For a given prefix in the routing process, the router 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 as follows:

- 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, 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 performs CoS-unaware load balancing of that EXP value across all the possible paths, including all of the TE tunnels of the selected set and the non-TE paths.
- If the routing process allocates all EXP values to tunnels or if a default is used, 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 overrides CBTS. PBR is an input process that the router performs ahead of regular forwarding.

Tunnel Failure Handling

This section includes the following topics:

- [Tunnel Up or Down, page 18-254](#)
- [Behavior When a Tunnel Goes Down, page 18-254](#)

Tunnel Up or Down

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.

Behavior When a Tunnel Goes Down

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

The following example shows the following parameters on tunnels T1, T2, and T3:

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

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

- 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

The following example shows the following parameters on tunnels T1, T2, and T3:

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

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

- 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 and T2, following existing CoS-unaware load balancing

Example 3—Two Default Tunnels Are Configured

The following example shows the following parameters on tunnels T1, T2, and T3:

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

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

- 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 as follows:

- 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 as follows:

- 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

Example 3 shows that the EXP default option is configured 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. These values of EXP bits are referred to as EXP-in and EXP-out.

If the packets for EXP-in are sent on a different tunnel than packets for EXP-out, the packets could get misordered within the same flows. For that reason, CBTS enables you to ensure that EXP-in and EXP-out never get mapped onto different tunnels.

CBTS enables you to configure EXP-in and EXP-out to be transported on the same tunnel when that tunnel is up to ensure that the feature does not introduce misordering of packets. If a tunnel failure occurs, the tunnel selection algorithm ensures that if EXP-in and EXP-out were carried on the same tunnel before the failure, that they are still carried on a single tunnel after the failure. Therefore, CBTS protects against nontransient misordering even in the event of a tunnel failure.



Note

CBTS does not try to force EXP-in and EXP-out packets to be carried on the same tunnel. You must configure CBTS so that EXP-in and EXP-out packets are carried on the same tunnel. This configuration is comparable to the regular DiffServ situation, where you must ensure that EXP-in and EXP-out packets are configured to go in the same queue.

Fast Reroute and Class-Based Tunnel Selection

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

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. The independence of these features includes the following considerations:

- 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.

FRR protects a given tunnel in exactly the same way as if CBTS were not configured on the tunnel.

DS-TE Tunnels and Class-Based Tunnel Selection

CBTS operates over tunnels using DS-TE. 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 Class-Based Tunnel Selection

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

Licensing Requirements for MPLS TE Class-Based Tunnel Selection

Product	License Requirement
Cisco NX-OS	CBTS requires an MPLS license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the <i>Cisco NX-OS Licensing Guide</i> .

Prerequisites for MPLS TE Class-Based Tunnel Selection

CBTS has the following prerequisites:

- You must enable the MPLS TE feature. You can enable or disable MPLS TE by entering the **[no] feature mpls traffic-eng** command.
- You must enable MPLS on all tunnel interfaces.

Guidelines and Limitations for MPLS TE Class-Based Tunnel Selection

CBTS has the following configuration guidelines and limitations:

- For a given destination, all CoS values are carried in tunnels that terminate at the same tailend. Either all CoS values are carried in tunnels or no values are carried in tunnels. 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.
- Any traffic that is coming from an M1-Series or M2-Series I/O module interface and going towards a Traffic Engineering (TE) Class-based Tunnel Selection (CBTS) tunnel will be dropped if you have configured the **ip wccp redirect exclude in** command on the inbound M1-Series or M2-Series I/O module interface or Switch Virtual Interface (SVI).

Configuring MPLS TE Class-Based Tunnel Selection

This section includes the following topics:

- [Creating Multiple MPLS TE Tunnels from the Same Headend to the Same Tailend, page 18-257](#)
- [Making the MPLS TE Tunnels Visible for Routing, page 18-259](#)



Note

Configure the CBTS feature only on the tunnel headend. No CBTS configuration is required on the tailend or transit LSR.

Creating Multiple MPLS TE Tunnels from the Same Headend to the Same Tailend

You can create multiple MPLS TE tunnels from the same headend to the same tailend. The first tunnel configured from the headend to the tailend is the master tunnel, which you can configure by entering the **ip unnumbered loopback 0** command. Within the master tunnel, you can create a CBTS bundle of member tunnels.

Prerequisites

You must have the MPLS TE feature enabled (see the [“Configuring MPLS TE” section on page 11-145](#)). Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel-te *number***
3. **ip unnumbered loopback *number***

4. **destination** {*hostname* | *ip-address*}
5. **cbts-member tunnel-te** *number*
6. (Optional) **exp** [*list-of-exp-values*] [**default**]
7. (Optional) **path-option** [**protect**] *preference-number* {**dynamic** | **explicit** {**identifier** *id* | **name** *name*} [**verbatim**]} [**lockdown**] [**bandwidth** *kbps*] [**attributes** *listname*]
8. (Optional) **bandwidth** *kbps*
9. **exit**
10. Repeat steps 5 through 9 to create additional member tunnels in this CBTS bundle.
11. **no shut**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface tunnel-te <i>number</i> Example: switch(config)# interface tunnel-te 65 switch(config-if-te)#	Enters TE interface configuration mode. The <i>number</i> argument identifies the tunnel number to be configured.
Step 3	ip unnumbered loopback <i>number</i> Example: switch(config-if-te)# ip unnumbered loopback 0	Configures an unnumbered IP interface for the master tunnel.
Step 4	destination { <i>hostname</i> <i>ip-address</i> } Example: switch(config-cbts-member)# destination 10.1.1.4	Specifies the destination for the tunnel. The <i>host-name</i> argument is the name of the host destination. The <i>ip-address</i> argument is the IPv4 address of the host destination.
Step 5	cbts-member tunnel-te <i>number</i> Example: switch(config-if-te)# cbts-member tunnel-te 251 switch(config-cbts-member)#	Configures a CBTS tunnel interface type and enters interface configuration mode. The <i>number</i> argument identifies the tunnel number to be configured. Note You can configure any existing MPLS TE command on these CBTS member tunnels.
Step 6	exp [<i>list-of-exp-values</i>] [default] Example: switch(config-cbts-member)# exp 5	(Optional) Specifies the EXP bits that are forwarded over the member tunnel.
Step 7	path-option [protect] <i>preference-number</i> { dynamic explicit { identifier <i>id</i> name <i>name</i> } [verbatim]} [lockdown] [bandwidth <i>kbps</i>] [attributes <i>listname</i>] Example: switch(config-cbts-member)# path-option 10 explicit Link5	(Optional) Configures the member tunnel to use an explicit path or a path dynamically calculated from the TE topology database. Note A dynamic path is used if an explicit path is currently unavailable.

	Command	Purpose
Step 8	bandwidth <i>kbps</i> Example: switch(config-cbts-member)# bandwidth 3000	(Optional) Configures the bandwidth for the member tunnel. The <i>kbps</i> argument is the number of kilobits per second set aside for the path option. The range is from 1 to 4294967295.
Step 9	exit Example: switch(config-cbts-member)# exit switch(config-if-te)#	Returns to TE interface configuration mode.
Step 10	Repeat steps 5 through 9 to create additional member tunnels in this CBTS bundle.	—
Step 11	no shut Example: switch(config-if-te)# no shut	Activates the bundle.

Making the MPLS TE Tunnels Visible for Routing

You can make the MPLS TE tunnels visible for routing.



Note

Alternatively, you can use static routing instead of autoroute to make the TE tunnels visible for routing.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel-te** *number*
3. **cbts-member tunnel-te** *number*
4. **autoroute announce**
5. **autoroute metric {absolute | relative} value**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface tunnel-te <i>number</i> Example: switch(config)# interface tunnel-te 65 switch(config-if-te)#	Enters TE interface configuration mode. The <i>number</i> argument identifies the tunnel number to be configured.

	Command	Purpose
Step 3	cbts-member tunnel-te <i>number</i> Example: switch(config-if-te)# cbts-member tunnel-te 65 switch(config-cbts-member)#	Configures a CBTS tunnel interface type and enters interface configuration mode. The <i>number</i> argument identifies the tunnel number to be configured.
Step 4	autoroute announce Example: switch(config-cbts-member)# autoroute announce	Specifies that the Interior Gateway Protocol (IGP) should use the tunnel (if the tunnel is up) in its enhanced SPF calculation.
Step 5	autoroute metric { absolute relative } <i>value</i> Example: switch(config-cbts-member)# autoroute metric relative 2	Specifies the MPLS TE tunnel metric that the IGP enhanced SPF calculation uses. Note Even though the value for a relative metric can be from –10 to +10, configuring a tunnel metric with a negative value is considered a misconfiguration. If the metric to the tunnel tailend appears to be 4 from the routing table, then the cost to the tunnel tailend router is actually 3 because 1 is added to the cost for getting to the loopback address. In this instance, the lowest value that you can configure for the relative metric is –3.

Verifying the MPLS TE Class-Based Tunnel Selection Configuration

To verify that the MPLS TE tunnels are operating and announced to the IGP, perform one of the following tasks.

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**] **protect**
3. **show mpls traffic-eng autoroute**

DETAILED STEPS

	Command	Purpose
Step 1	show mpls traffic-eng topology {ip-address igp-id {isis nsap-address ospf ip-address} [brief] Example: switch# show mpls traffic-eng topology	Displays the MPLS TE global topology currently known at this node.
Step 2	show mpls traffic-eng tunnels number [brief] [protection] Example: switch# show mpls traffic-eng tunnels 500 brief protection	Displays information for a specified tunneling interface.
Step 3	show mpls traffic-eng autoroute Example: switch# show mpls traffic-eng autoroute	Displays tunnels that are announced to the IGP, including interface, destination, and bandwidth.

Examples

The following example shows how to display the MPLS TE global topology:

```
switch# 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
    globalpoolsubpool
    total allocatedreservable reservable
    -----
  bw[0]: 0 1000500
  bw[1]:10 990490
  bw[2]: 600 390390
  bw[3]: 0 390390
  bw[4]: 0 390390
  bw[5]: 0 390390
```

The following example shows how to display information for a specified tunneling interface:

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

switch#_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

```

The following example shows how to display tunnels that are announced to the IGP, including interface, destination, and bandwidth:

```

switch# show mpls traffic-eng autoroute

MPLS TE autorouting enabled
destination 0002.0002.0002.00 has 2 tunnels
  Tunnel1021 (traffic share 10000, nexthop 10.2.2.2, absolute metric 11)
  Tunnel1022 (traffic share 3333, nexthop 10.2.2.2, relative metric -3)
destination 0003.0003.0003.00 has 2 tunnels
  Tunnel1032 (traffic share 10000, nexthop 172.16.3.3)
  Tunnel1031 (traffic share 10000, nexthop 172.16.3.3, relative metric -1)

```

Configuration Examples for MPLS TE Class-Based Tunnel Selection

This section includes the following configuration examples:

- [Example: Creating Multiple MPLS TE Tunnels from the Same Headend to the Same Tailend, page 18-262](#)
- [Example: Making the MPLS TE Tunnels Visible for Routing, page 18-263](#)
- [Example: Verifying that the MPLS TE Tunnels are Operating and Visible to the IGP, page 18-263](#)

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

The following example shows how to create a CBTS bundle of four MPLS TE tunnels from the same headend to the same tailend. Three explicit tunnels carry traffic with EXP values of 6, 0, and 3, respectively, while the fourth tunnel is dynamic and carries traffic with the remaining EXP values.

```

! create explicit paths for member tunnels
mpls traffic-eng configuration
  explicit-path name path1
    index 3 next-address 192.0.45.4
    index 5 next-address 10.1.1.4
  exit
  explicit-path name path2
    index 3 next-address 192.1.45.4
    index 5 next-address 10.1.1.4

```

```

    exit
    explicit-path name path3
    index 2 next-address 192.0.15.1
    index 3 next-address 192.0.14.4
    index 5 next-address 10.1.1.4
    exit

! create master tunnel
interface tunnel-te 25
 ip unnumbered loopback0
 destination 10.1.1.4
! create member tunnels
cbts-member tunnel-te 251
 exp 6 7
  path-option 1 explicit name path1
 exit
cbts-member tunnel-te 252
 exp 0 1
  path-option 1 explicit name path2
 exit
cbts-member tunnel-te 253
 exp 3
  path-option 1 explicit name path3
 exit
cbts-member tunnel-te 254
 exp default
  path-option 1 dynamic
 exit
no shut
exit

```

Example: Making the MPLS TE Tunnels Visible for Routing

The following example shows how to make MPLS TE tunnels visible for routing:

```

interface tunnel-te 65
cbts-member tunnel-te 65
 autoroute announce
 autoroute metric relative -2
 exit
exit
interface tunnel-te 66
cbts-member tunnel-te 66
 autoroute announce
 autoroute metric relative -2
end

```

Example: Verifying that the MPLS TE Tunnels are Operating and Visible to the IGP

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

The following example shows how to display the MPLS TE global topology:

```

switch# 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
physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)

```

	Total Allocated BW (kbps)	Global Pool Reservable BW (kbps)
	-----	-----
bw[0]:	0	1000
bw[1]:	0	1000
bw[2]:	0	1000
bw[3]:	0	1000
bw[4]:	0	1000
bw[5]:	0	1000
bw[6]:	0	1000
bw[7]:	100	900

```

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
physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)

```

	Total Allocated BW (kbps)	Global Pool Reservable BW (kbps)
	-----	-----
bw[0]:	0	1000
bw[1]:	0	1000
bw[2]:	0	1000
bw[3]:	0	1000
bw[4]:	0	1000
bw[5]:	0	1000
bw[6]:	0	1000
bw[7]:	300	700

switch#

The following example shows how to display the MPLS TE topology for 10.0.0.9:

switch# **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
physical_bw: 155000 (kbps), max_reservable_bw_global: 1000 (kbps)

```

	Total Allocated BW (kbps)	Global Pool Reservable BW (kbps)
	-----	-----
bw[0]:	0	1000
bw[1]:	0	1000
bw[2]:	0	1000
bw[3]:	0	1000
bw[4]:	0	1000
bw[5]:	0	1000
bw[6]:	0	1000
bw[7]:	0	1000

```

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
physical_bw: 155000 (kbps), max_reservable_bw_global: 1000 (kbps)

```

	Total Allocated BW (kbps)	Global Pool Reservable BW (kbps)
	-----	-----
bw[0]:	0	1000
bw[1]:	0	1000
bw[2]:	0	1000
bw[3]:	0	1000
bw[4]:	0	1000
bw[5]:	0	1000
bw[6]:	0	1000
bw[7]:	0	1000

switch#

The following example shows how to display display information about a tunnel:

```
switch# 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 bw-based
  auto-bw: disabled

Active Path Option Parameters:
  State: explicit path option 1 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled

InLabel  : -
OutLabel : Ethernet6/0, 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
```

The following example shows how to display information about a tunnel:

```
switch# 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 bw-based
  auto-bw: disabled
Active Path Option Parameters:
  State: explicit path option 1 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled

InLabel : -
OutLabel : Ethernet6/1, 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

```

The following example shows how to display information about a tunnel:

```

switch# 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 bw-based
  auto-bw: disabled
Active Path Option Parameters:
  State: explicit path option 1 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled

InLabel : -
OutLabel : Ethernet6/1, 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

```

The following example shows how to display information about a tunnel:

```
switch# 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 bw-based
  auto-bw: disabled
Active Path Option Parameters:
  State: explicit path option 1 is active
  BandwidthOverride: disabled LockDown: disabled Verbatim: disabled

InLabel  : -
OutLabel  : Ethernet6/1, 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 following example shows how to display tunnels that are announced to the IGP:

```
switch# 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)

```

```
switch# Tunnel4 (flags: Announce)
          (load balancing metric 20000000, nexthop 10.0.0.9)
          (flags: Announce)
```

Additional References MPLS TE Class-Based Tunnel Selection

The following sections provide references related to the MPLS TE—Class-based Tunnel Selection feature.

Related Documents

Related Topic	Document Title
MPLS TE commands	Cisco NX-OS Multiprotocol Label Switching Command Reference

MIBs

MIB	MIBs Link
<ul style="list-style-type: none">CISCO-IETF-FRR-MIBMPLS TE-STD-MIB	To locate and download Cisco MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Feature History for MPLS TE Class-Based Tunnel Selection

[Table 18-1](#) lists the release history for this feature.

Table 18-1 Feature History for Class-Based Tunnel Selection

Feature Name	Releases	Feature Configuration Information
Class-based tunnel selection	5.2(1)	This feature was introduced.

