Configuring CRS-1 Series Virtual Interfaces

A virtual interface is defined as representing a logical packet switching entity within the Cisco CRS-1 Series router. Virtual Interfaces have a global scope and do not have an associated location. The Cisco IOS-XR software uses the *r/s/m/p* notation for identifying physical interfaces, but virtual interfaces have instead a globally unique numerical ID after their names—for example, Loopback 0 Loopback 1... Loopback 99999. The ID is unique per virtual interface type to make the entire name string unique such that you can have both Loopback 0 and Null 0.

Virtual interfaces currently have their control plane presence on the active RP. The configuration and control plane is mirrored onto the standby RP, and in the event of a failover, the virtual interfaces will move to the ex-standby, which then becomes the newly active RP.

In future releases of the Cisco CRS-1 Series, virtual interfaces will be distributed across other RPs, but this is not yet supported.

Note Subinterfaces can be physical or virtual, depending on their parent interface.

CRS-1 Series supports the following virtual interface types:

- Loopback 0—Virtual interfaces that provide a consistent interface/IP address pair used by routing protocols to advertise as the IP address for the router on which they are running.
- Null 0—Packets routed to a null interface are dropped.
- Tunnel-TE—Tunnel interfaces permit encapsulating packets of one protocol inside a different protocol.

Note Tunnels do not have a one-to-one line card association.

Contents

This chapter includes the following sections:

- “Configuring CRS-1 Series MPLE-TE Tunnels” section on page 67
- “Implementing CRS-1 Series MPLS Traffic Engineering” section on page 71

Virtual Interface Naming Convention

Virtual interface names never use the physical interface naming notation *r/s/m/p* for identifying an interface’s rack/slot/module/port because they are not tied to any physical location.

Virtual interfaces use a globally unique numerical identifier (per virtual interface type).

Examples of naming notation for virtual interfaces:

```
Interface                      IP-Address      Status  Protocol
Loopback0                      127.0.0.1         Up            Up
Loopback10                     127.0.0.1         Up            Up
Tunnel-TE5000                  92.166.255.255    Down          Down
Null10                         10.8.0.0          Up            Up
```

Active and Standby RPs and Virtual Interface Configuration

The standby RP is available and in a state in which it can take over the work from the active RP should that prove necessary. Conditions that necessitate the standby RP to become the active RP and assume the active RP’s duties include:

- Failure detection by a watchdog.
- Administrative command to take over.
- Removal of the active RP from the chassis.

If a second RP is not present in the chassis while the first is in operation, a second RP may be inserted and will automatically become the standby RP. The standby RP may also be removed from the chassis with no effect on the CRS-1 Series system other than loss of RP redundancy.

After failover, the virtual interfaces will all be present on the standby (now active) RP. Their state and configuration will be unchanged and there will have been no loss of forwarding (in the case of tunnels) over the interfaces during the failover. The CRS-1 Series system uses NonStop Forwarding (NSF) over tunnels through the failover of the host RP.

For more information, see the sections “Cisco CRS-1 Series Virtual Interfaces and Failover or Failback” and “Cisco CRS-1 Series Management Interfaces on the RP” in this guide.

Note The user does not need to configure anything in order to guarantee that the standby interface configurations are maintained.

Configuring CRS-1 Series Loopback Interfaces

You can specify a software-only interface called a loopback interface to emulate a physical interface. A loopback interface is a virtual interface that is always up and allows BGP sessions, for example, to stay up even if the outbound interface is down.

You can use the loopback interface as the termination address for OSPF or BGP sessions, or to establish a Telnet session from the CRS-1 Series system's console to its auxiliary port when all other interfaces are down. In applications where other routers or access servers attempt to reach this loopback interface, you should configure a routing protocol to distribute the subnet assigned to the loopback address.

Packets routed to the loopback interface are rerouted back to the router or access server and processed locally. IP packets routed out the loopback interface but not destined to the loopback interface are dropped. This means that the loopback interface also serves as the Null 0 interface.

SUMMARY STEPS

Creation of a Loopback virtual interface is shown in the following steps:

Step 1 Configure and name the loopback interface, for example:

```
interface Loopback 1
```

Step 2 Configure an IPV4 address for the loopback, for example:

```
ipv4 address 10.2.3.0 255.255.255.0
```

Step 3 Commit the configuration to the router running configuration:

```
Commit
```

Step 4 Verify the loopback, for example:

```
show interfaces Loopback 1
```

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure terminal`
	Example:
	`Router> configure terminal`
		Enters global configuration mode.

Step 2	`interface Loopback` *`identifier`*
	Example:
	`Router>(config)# interface Loopback 3`
		Configure and name the new loopback interface.
		This **interface** command will enter you into the interface submode, from where interface specific config commands are entered. Use **exit** to exit from the interface submode back to the normal config mode.

Step 3	`ipv4 address` *`ip address`*
	Example:
	`Router(config-if)# ipv4 address 10.0.0.1`
		Assigns an IP address and subnet mask to the virtual loopback interface using the **ip address** configuration subcommand.

Step 4	`commit`
	Example:
	`Router# commit`
		Commits the target configuration to the router running configuration

Step 5	`show interfaces` *`name`*
	Example:
	`Router# show interfaces Loopback 3`
		Verifies the configuration of the Loopback interface.

Configuring CRS-1 Series Null Interfaces

Cisco CRS-1 Series supports null interface configurations. A null interface functions similarly to the null devices available on most operating systems. This interface is always up and can never forward or receive traffic; encapsulation always fails. The null interface provides an alternative method of filtering traffic. You can avoid the overhead involved with using access lists by directing undesired network traffic to the null interface.

The only interface configuration command that you can specify for the null interface is the **ipv4 unreachables** command. With the **ipv4 unreachables** command, if the software receives a nonbroadcast packet destined for itself that uses a protocol it does not recognize, it sends an Internet Control Message Protocol (ICMP) protocol unreachable message to the source. If the software receives a datagram that it cannot deliver to its ultimate destination because it knows of no route to the destination address, it replies to the originator of that datagram with an ICMP host-unreachable message.

To use configuration commands, you must be in a user group associated with a task group that includes the proper task IDs. To use the **ipv4 unreachables** command, you must be a member of a user group associated with the network task ID. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

SUMMARY STEPS

Null 0 interface is created by default on the RP during boot, and cannot be removed. The **ipv4 unreachables** command can be configured for this interface, but most configuration is unnecessary because this interface just discards all the packets sent to it.

The Null o interface can be displayed as follows:

```
show interfaces Null 0
```

Configuration of a Null virtual interface involves the following steps:

Step 1 Configure and name the null interface.

```
interface Null 0
```

Step 2 Enable the generation of IPv4 Internet Control Message Protocol (ICMP) unreachable messages by using the **ipv4 unreachables disable** command in interface configuration mode.

```
ipv4 unreachables disable
```

Step 3 Commit the configuration to the router running configuration:

```
Commit
```

Step 4 Verify the Null interface:

```
show interfaces Null 0
```

Note The null interface can be used in any command that has an interface type as an argument.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`configure terminal`**
	Example:
	`Router> configure terminal`
		Enters global configuration mode.

Step 2	**`interface Null`** *`identifier`*
	Example:
	`Router>(config)#` **`interface Null 0`**
		Configure and name the new Null 0 interface.

Step 3	**`ipv4 unreachables disable`**
	Example:
	`Router>(config-null0)#` **`ipv4 unreachables disable`**
		Generates IPv4 ICMP unreachables messages. This command has no arguments or keywords.

Step 4	**`commit`**
	Example:
	`Router>(config-null0)#` **`commit`**
		Commits the target configuration to the router running configuration.

Step 5	**`show interfaces`** *`Null identifier`*
	Example:
	`Router#` **`show interfaces Null0`**
		Verifies the configuration of the Null interface.

Example

The following example configures a null interface for IPv4 route 10.2.0.0:

```
ipv4 route 10.2.0.0 255.0.0.0 null0
```

Tunnel Interfaces Overview

Tunneling provides a way to encapsulate arbitrary packets inside of a transport protocol. This feature is implemented as a virtual interface to provide a simple interface for configuration. The CRS-1 Series tunnel interfaces are not tied to specific “passenger” or “transport” protocols, but, rather, they represent an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. Because tunnels are point-to-point links, you must configure a separate tunnel for each link.

There are three necessary steps in configuring a tunnel interface:

1. Specify the tunnel interface—**interface tunnel-te** *number*.
2. Configure the tunnel source—**tunnel source** {ip-address | type number}.
3. Configure the tunnel destination—**tunnel destination** {hostname | ip-address}.

The CRS-1 Series system supports MPLS-TE for MPLS traffic-engineering in its initial release.

CRS-1 Series Tunnel Naming Conventions

The new tunnel interface naming notation for CRS-1 Series MPLS-TE tunnels are:

- tunnel-te *nn*

where *nn* are unique identifiers.

The new CRS-1 Series tunnel naming convention employs a new method for configuring tunnels by embedding the tunnel mode in the tunnel name.

The tunnel interface name embeds the tunnel mode via the following command format:

```
Tunnel mode-substring tunnelid
```

For example:

```
interface tunnel-te 1000
```

Note that tunnel names now have different modes but can have the same tunnel id.

For example,

```
configure terminal
    interface tunnel-te 0
    path-option 1 dynamic
```

The advantage of embedding the tunnel mode in the name is that it boosts performance in the control plane due to the removal of redundant tunnel mode commands, which in turn avoids base encapsulation replacement on each tunnel. It also allows for future distribution of tunnels across multiple RPs based on their encapsulation, which will allow the number of tunnels supported to increase much further.

All mode-specific commands are configured from the new CRS-1 Series CLI tunnel mode submode to the interface submode.

Note The new CRS-1 Series IOS-XR tunnel naming convention is backward compatible with the Cisco IOS method for configuring tunnels.

The following sections describe how to configure each of the CRS-1 Series tunnel types in turn.

Configuring CRS-1 Series MPLE-TE Tunnels

MPLS is a label-switching technology that creates and uses a Virtual Circuit (VC) switching function. MPLS integrates Layer 2 and Layer 3 technologies by making traditional Layer 2 features available to Layer 3.

Your network must support the following Cisco features before you can enable MPLS traffic engineering:

- MPLS
- IP Cisco Express Forwarding (CEF)
- Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF) routing protocol

With the creation of the new tunnel names (containing the encapsulation) there is no need for a tunnel mode submode when configuring TE tunnels—the commands are now available in the tunnel-te interface submode directly.

For example:

```
configure terminal
int Tunnel?
Tunnel-GRE Tunnel-TE Tunnel-IPSec
interface tunnel-te 100
priority 6 6
 bandwidth 1000
 path-option 1 dynamic
 path-option 2 explicit name PATH2
 record-route
 tunnel destination 172.19.120.40
```

Differences Between Cisco IOS and Cisco IOS-XR MPLS-TE Configuration

The following characteristics and features of MPLS Traffic Engineering (MPLS-TE) are similar on both Cisco IOS Software and Cisco IOS-XR Software:

- MPLS-TE topology.
- Path calculation (PCALC).
- Differentiated Services Traffic Engineering (DS-TE).
- Fast reroute.
- Flooding.

The following characteristics and features of MPLS-TE are new in Cisco IOS-XR Software:

- New configuration modes.
- Protocol-based command-line interface (CLI).
- Task IDs are now implemented for a new level of system control. To configure MPLS-TE tunnels, you must be a member of a user group associated with the mpls-te task ID.

> **Note** To use configuration commands, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

- Router IDs are configured globally, unless they are overridden using the **mpls traffic-eng router-id** command.
- New show commands to facilitate Cisco IOS-XR software operation.
- While MPLS-TE tunnel functionality is similar to Cisco IOS software, Cisco IOS-XR software features a new **interface tunnel-te** command and eliminates the tunnel mode mpls traffic-eng mode.
- Elimination of the mpls traffic-eng tunnels command.

Prerequisites

The following prerequisites are needed to implement MPLS-TE on the Cisco CRS-1 Series router:

- Installation of a full image or a composite mini-image plus an MPLS package.
- IGP must be running.

- To configure MPLS-TE, you must be a member of a user group associated with the mpls-te task ID set up through AAA configuration.
  - For show and debug commands you need “read” privileges.
  - For any action command you need “read/write” privileges.

Note To use the configuration commands, you must be in a user group associated with a task group that includes the proper task IDs. For detailed information about user groups and task IDs, refer to the *Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*.

Essential Conceptual Information

To configure MPLS-TE, you must understand the following concepts:

- MPLS Traffic Engineering
- Differentiated Services
- Fast Reroute
- Flooding

MPLS Traffic Engineering

MPLS traffic engineering on the CRS-1 Series provides the advantage of Layer 2 and Layer 3 integration without needing to configure a separate network. Traffic engineering tunnels are calculated at the link state protocol (LSP) head based on a fit between required and available resources (constraint-based routing). The Interior Gateway Protocol (IGP) automatically routes the traffic onto these LSPs.

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

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

Differentiated Services

MPLS Differentiated Services (diff-serv) aware Traffic Engineering (DS-TE) is an extension of the regular MPLS Traffic Engineering (TE) feature. Regular traffic engineering does not provide bandwidth guarantees to different traffic classes. A single bandwidth pool (global pool) is used in regular TE that is shared by all traffic. In order to support various classes of service (CoS), you must have the ability to provide multiple bandwidth pools. These bandwidth pools then can be treated differently based on the requirement for the traffic class using that pool.

MPLS diff-serv traffic engineering provides the ability to configure global and subpool(s) to provide reservable bandwidths on an interface basis.

When a TE tunnel is configured to use one of these bandwidth pools, available bandwidth from all configured bandwidth pools is advertised via IGP. Path calculation and admission control then takes the bandwidth pool type into consideration. RSVP is used to signal the TE tunnel with bandwidth pool requirements.

Fast Reroute

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

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

Note With the ability to configure FRR on a per-LSP basis, it is possible to effectively provide different levels of fast restoration to tunnels from different bandwidth pools.

Flooding

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

Flooding Triggers

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

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

Flooding Thresholds

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

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

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

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

Implementing CRS-1 Series MPLS Traffic Engineering

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

- Building MPLS-TE Topology
- Creating an MPLS-TE Tunnel
- Forwarding over the MPLS-TE Tunnel

Building MPLS-TE Topology

This task explains how to configure MPLS-TE topology, which is required for traffic engineering tunnel operations. Building the MPLS-TE topology is done by performing the following basic steps:

- Enabling MPLS-TE on the port interface.
- Enabling RSVP on the port interface.
- Enabling an IGP such as OSPF or IS-IS for MPLS-TE.

Prerequisites

The following are the requirements for traffic engineering:

- You must have a router ID for the neighbor router being linked in order to configure discovery for the local router.
- A stable router ID is required at either end of the link to ensure the link will be successful. If you do not assign a router ID to the routers, the system will default to the global router ID as it does in Cisco IOS software. Default router IDs are subject to change causing an unstable link.
- If you are going to use nondefault holdtime or intervals, you must decide the values to which they will be set.

SUMMARY STEPS

Step 1 Enters configuration mode

```
configure terminal
```

Step 2 Enables mpls traffic-eng, and enters the mpls traffic-eng configuration submode.

```
mpls traffic-eng
```

Step 3 Enables mpls traffic-eng on a particular interface on the originating node. In this case, on the POS interface 0/6/0/0.

interface *type number*

Step 4 Specifies the router ID of the local node. In Cisco IOS-XR software, the router ID can be specified with an interface name or an IP address. By default, MPLS uses the global router ID, the same as Cisco IOS software.

router-id {*interface-name* | *ip-address*}

Step 5 Configures an OSPF routing instance. Enters the IGP submode and configures the area and interface for an IGP such as OSPF or IS-IS.

router ospf *instance-name*

Step 6 Configures a router ID for the OSPF process using an IP address.

router-id {*ip-address* | *interface-name*}

Step 7 Configures an area for the OSPF process. Backbone areas have an area ID of 0. Non-backbone areas have a non-zero area ID.

area *area-id*

Step 8 Configures one or more interfaces for the area configured in Step 7.

interface *type number*

Step 9 Enables IGP on the Loopback 0 MPLS router ID.

interface Loopback 0

Step 10 Sets the MPLS traffic engineering loopback interface.

mpls traffic-eng router-id loopback0

Step 11 Sets the MPLS traffic engineering area.

mpls traffic-eng area *area-id*

Step 12 Enables RSVP, and enters the RSVP configuration submode.

rsvp

Step 13 Enters RSVP interface submode, and enables RSVP on a particular interface on the originating node. In this case, on the POS interface 0/6/0/0.

interface *type number*

Step 14 Sets the reserved RSVP bandwidth available on this interface. Physical interface bandwidth is not used by MPLS traffic-engineering.

bandwidth *bandwidth*

Step 15 Saves configuration changes.

end or **commit**

Step 16 Verifies the traffic engineering topology.

show mpls traffic topology

Step 17 Displays all the link-management advertisements for the links on this node.

show mpls traffic-eng link-management advertisements

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** router(config)# **configure terminal**	Enters the configuration mode.
Step 2	**mpls traffic-eng** **Example:** router(config)# **mpls traffic-eng**	Enters the MPLS traffic-engineering interface configuration mode.
Step 3	**interface** *type number* **Example:** router(config-mpls-te)# **interface POS 0/6/0/0**	Enables traffic engineering on a particular interface on the originating node. In this case, POS interface 0/6/0/0.
Step 4	**router id** *{interface-name | ip-address}* **Example:** router(config)# **router id Loopback 0**	Specifies the router ID of the local node. In Cisco IOS-XR software, the router ID can be specified with an interface name or an IP address. By default, MPLS uses the global router ID, the same as Cisco IOS software.
Step 5	**router ospf** *instance-name* **Example:** router(config)# **router ospf 100**	Configures an OSPF routing instance. Enters the IGP submode and configures the area and interface for an IGP such as OSPF or IS-IS.
Step 6	**router-id** *{ip-address | interface-name}* **Example:** router(config-router)# **mpls traffic-eng router-id 192.168.25.66**	Configures a router ID for the OSPF process using an IP address. It is also possible to configure a router ID using a Loopback interface, as in: router(config-router)# **mpls traffic-eng router-id Loopback 0**
Step 7	**area** *area-id* **Example:** router(config-router)# **mpls traffic-eng area 0**	Configures an area for the OSPF process. Backbone areas have an area ID of 0. Non-backbone areas have a non-zero area ID.
Step 8	**interface** *type number* **Example:** router(config-ospf-ar)# **interface pos 0/6/0/0**	Configures one or more interfaces for the area configured in Step 7.
Step 9	**interface** *interface-name* **Example:** router(config-ospf-ar)# **interface Loopback 0**	Enables IGP on the Loopback 0 MPLS router ID.

	Command or Action	Purpose
Step 10	**mpls traffic-eng router-id Loopback 0** **Example:** router(config-router)# **mpls traffic-eng router-id Loopback 0**	Sets the MPLS traffic engineering loopback interface.
Step 11	**mpls traffic-eng area** *area-id* **Example:** router(config-router)# **mpls traffic-eng area 0**	Sets the MPLS traffic engineering area.
Step 12	**rsvp** **Example:** router(config)# **rsvp**	Enables RSVP, and enters the RSVP configuration submode.
Step 13	**interface** *type number* **Example:** router(config-rsvp)# **pos 0/6/0/0**	Enters RSVP interface submode, and enables RSVP on a particular interface on the originating node. In this case, on the POS interface 0/6/0/0.
Step 14	**bandwidth** *bandwidth* **Example:** :router(config-rsvp-if)# **bandwidth 100**	Sets the reserved RSVP bandwidth available on this interface. Physical interface bandwidth is not used by MPLS traffic-engineering.
Step 15	**end** or **commit** **Example:** router(config-rsvp-if)# **end** or router(config-rsvp-if)# **commit**	Saves configuration changes. When you enter the end command, the system prompts you to commit changes: Uncommitted changes found. Commit them? —Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. —Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. When you enter the commit command, the system saves the configuration changes to the running configuration file and remains within the configuration session.
Step 16	**show mpls traffic topology** **Example:** router# **show mpls traffic topology**	Verifies the traffic engineering topology.
Step 17	**show mpls traffic-eng link-management advertisements** **Example:** router# **show mpls traffic-eng link-management advertisements**	Displays all the link-management advertisements for the links on this node.

Creating an MPLS-TE Tunnel

After the traffic engineering topology is built, the next task is to create an MPLS-TE tunnel. Creating an MPLS-TE tunnel is a process of customizing the traffic engineering to fit your network topology.

Prerequisites

The following are the requirements for traffic engineering:

- You must have a router ID for the neighbor router being linked in order to configure discovery for the local router.
- A stable router ID is required at either end of the link to ensure the link will be successful. If you do not assign a router ID to the routers, the system will default to the global router ID as it does in Cisco IOS software. Default router IDs are subject to change, causing an unstable link.
- If you are going to use non-default holdtime or intervals, you must decide the values to which they will be set.
- Configure an MPLS-TE tunnel only after you have created a traffic engineering topology.

Perform the following steps to create an MPLS-TE tunnel.

SUMMARY STEPS

Use the following steps to create an MPLS-TE tunnel:

Step 1 Enter configuration mode and create the new tunnel with the **tunnel-te interface** command and number keyword.

```
configure terminal
interface tunnel-te number
```

Step 2 Specify the tunnel destination.

```
tunnel destination {ip-address | tunnel-id}
```

Step 3 Set an unnumbered loopback interface.

```
ipv4 unnumbered Loopback number
```

Step 4 Set the tunnel path option.

```
path-option {path-id} {dynamic | explicit path_name}
```

Step 5 Set the tunnel bandwidth.

```
bandwidth bandwidth
```

Step 6 Commit the configuration to the running configuration.

```
commit
```

Step 7 Verify the MPLS-TE tunnel configuration.

```
show mpls traffic-eng tunnels
```

Step 8 Enable forwarding on the MPLS-TE tunnel you just created.

```
route ipv4 {ip-address} {bits} tunnel {tunnel-id}
```

Step 9 Commit the target configuration to the running configuration.

```
commit
```

Step 10 Verify forwarding.

```
show mpls forwarding tunnels
show ip cef
```

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters configuration mode.
Step 2	**interface tunnel-te** *number* **Example:** `Router>(config-if)# interface tunnel-te 1000`	Enters the MPLS traffic-engineering tunnel interface submode, and enables traffic engineering on a particular interface on the originating node; in this case, on tunnel-te interface 1000.
Step 3	**autoroute announce** **Example:** `Router>(config-if)# autoroute announce`	Specifies that the Interior Gateway Protocol (IGP) should use the tunnel (if the tunnel is up) in its enhanced shortest path first (SPF) calculation. To specify that the IGP does not use the tunnel in its enhanced SPF calculations, use the **no** form of this command.
Step 4	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.255.255`	Assigns an interface destination address to the new tunnel. The destination address is the remote node's MPLS traffic-engineering router ID.
Step 5	**ipv4 unnumbered Loopback** *number ipv4 address* **Example:** `Router(config-if)# ipv4 unnumbered Loopback 0`	Assigns an interface source address so that forwarding can be performed on the new tunnel. This is required for the tunnel to actually be used.

Step 6	**path-option** *path-id* {**dynamic** | **explicit** *path_name*}
	Example:
	`Router(config-if)# path-option 1 dynamic`
		Sets the path option to dynamic and also assigns the path to path 1.

Step 7	**bandwidth** *bandwidth [subpool]* **Example:** `Router(config-if)# bandwidth 100`	Sets the bandwidth required on the interface; the range is 0–4294967295.
Step 8	**commit** **Example:** `Router(config-if)# commit`	Commits the target configuration to the router running configuration.
Step 9	**show mpls traffic-eng tunnels** **show ipv4 interface brief** **show mpls traffic-wng autoroute** **show mpls forwarding tunnels** **show ip cef** **Example:** `Router# show mpls traffic-eng tunnels`	Verifies the configuration of the new MPLS-TE tunnel.

What To Do Next

After creating MPLS-TE tunnel interfaces, you may want to protect the tunnels with fast reroute, and create differentiated services MPLS-TE traffic engineering tunnels. For information about these tasks, refer to the *Cisco IOS-XR Multiprotocol Label Switching Command Reference* and the *Cisco IOS-XR Multiprotocol Label Switching Configuration Guide.*

Additional References

The following sections provide references related to interface configuration.

Related Documents

Related Topic	Document Title
Cisco IOS-XR master command reference	*Cisco IOS-XR Master Commands List, Initial Release*
Cisco IOS-XR interface configuration commands	*Cisco IOS-XR Interface and Hardware Component Command Reference*
Information about forwarding over MPLE-TE tunnels, and protecting MPLS-TE tunnels with fast reroute.	*Cisco IOS-XR Multiprotocol Label Switching Command Reference* and *Cisco IOS-XR Multiprotocol Label Switching Configuration Guide*
Information about user groups and task IDs	*Configuring AAA Services on Cisco IOS-XR Software* module of the *Cisco IOS-XR System Security Configuration Guide*
Information about configuring interfaces and other components on the Cisco CRS-1 Series router from a remote Craft Works Interface (CWI) client management application.	*Cisco CRS-1 Series Carrier Routing System Craft Works Interface Configuration Guide*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Glossary

See the book-level glossary at the end of this guide.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in the glossary.