THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, 
INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, 
EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH 
THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, 
CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version 
of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. 
CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF 
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT 
LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS 
HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network 
topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional 
and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http:// 
www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership 
relationship between Cisco and any other company. (1110R)

© 2013 Cisco Systems, Inc. All rights reserved.
CONTENTS

Preface xv
  Changes to This Document xv
  Obtaining Documentation and Submitting a Service Request xv

CHAPTER 1 New and Changed Feature Information in Cisco IOS XR Release 4.3.x 1
  New and Changed MPLS Feature Information 1
  New and Changed Feature Information in Cisco IOS XR Release 4.3.x 2

CHAPTER 2 Implementing MPLS Label Distribution Protocol 3
  Prerequisites for Implementing Cisco MPLS LDP 4
  Information About Implementing Cisco MPLS LDP 5
    Overview of Label Distribution Protocol 5
      Label Switched Paths 5
      LDP Control Plane 5
      Exchanging Label Bindings 5
      LDP Forwarding 6
    LDP Graceful Restart 8
      Control Plane Failure 8
      Phases in Graceful Restart 9
      Recovery with Graceful-Restart 11
    Label Advertisement Control (Outbound Filtering) 12
    Label Acceptance Control (Inbound Filtering) 12
    Local Label Allocation Control 13
    Session Protection 13
    IGP Synchronization 14
    IGP Auto-configuration 15
      IGP Synchronization Process Restart Delay 15
LDP Nonstop Routing 16
IP LDP Fast Reroute Loop Free Alternate 16
Downstream on Demand 18

How to Implement MPLS LDP 18
Configuring LDP Discovery Parameters 18
Configuring LDP Discovery Over a Link 20
Configuring LDP Discovery for Active Targeted Hellos 22
Configuring LDP Discovery for Passive Targeted Hellos 24
Configuring Label Advertisement Control (Outbound Filtering) 26
Setting Up LDP Neighbors 28
Setting Up LDP Forwarding 30
Setting Up LDP NSF Using Graceful Restart 31
Configuring Label Acceptance Control (Inbound Filtering) 33
Configuring Local Label Allocation Control 35
Configuring Session Protection 36
Configuring LDP IGP Synchronization: OSPF 36
Configuring LDP IGP Synchronization: ISIS 37
Configuring LDP IGP Synchronization Delay Interval 39
Configuring LDP IGP Synchronization Process Restart Delay 39
Enabling LDP Auto-Configuration for a Specified OSPF Instance 40
Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance 42
Disabling LDP Auto-Configuration 43
Configuring LDP Nonstop Routing 44
Configuring LDP Downstream on Demand mode 45
Redistributing MPLS LDP Routes into BGP 46

Configuration Examples for Implementing MPLS LDP 47
Configuring LDP with Graceful Restart: Example 48
Configuring LDP Discovery: Example 48
Configuring LDP Link: Example 48
Configuring LDP Discovery for Targeted Hellos: Example 48
Configuring Label Advertisement (Outbound Filtering): Example 49
Configuring LDP Neighbors: Example 49
Configuring LDP Forwarding: Example 50
Configuring LDP Nonstop Forwarding with Graceful Restart: Example 50
Configuring Label Acceptance (Inbound Filtering): Example 50
### Implementing RSVP for MPLS-TE and MPLS O-UNI

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites for Implementing RSVP for MPLS-TE and MPLS O-UNI</td>
<td>60</td>
</tr>
<tr>
<td>Information About Implementing RSVP for MPLS-TE and MPLS O-UNI</td>
<td>60</td>
</tr>
<tr>
<td>Overview of RSVP for MPLS-TE and MPLS O-UNI</td>
<td>60</td>
</tr>
<tr>
<td>LSP Setup</td>
<td>61</td>
</tr>
<tr>
<td>High Availability</td>
<td>62</td>
</tr>
<tr>
<td>Graceful Restart</td>
<td>62</td>
</tr>
<tr>
<td>Graceful Restart: Standard and Interface-Based</td>
<td>62</td>
</tr>
<tr>
<td>Graceful Restart: Figure</td>
<td>63</td>
</tr>
<tr>
<td>ACL-based Prefix Filtering</td>
<td>64</td>
</tr>
<tr>
<td>RSVP MIB</td>
<td>65</td>
</tr>
<tr>
<td>Information About Implementing RSVP Authentication</td>
<td>65</td>
</tr>
<tr>
<td>RSVP Authentication Functions</td>
<td>65</td>
</tr>
<tr>
<td>RSVP Authentication Design</td>
<td>65</td>
</tr>
<tr>
<td>Global, Interface, and Neighbor Authentication Modes</td>
<td>66</td>
</tr>
<tr>
<td>Security Association</td>
<td>67</td>
</tr>
<tr>
<td>Key-source Key-chain</td>
<td>69</td>
</tr>
<tr>
<td>Guidelines for Window-Size and Out-of-Sequence Messages</td>
<td>69</td>
</tr>
<tr>
<td>Caveats for Out-of-Sequence</td>
<td>70</td>
</tr>
<tr>
<td>How to Implement RSVP</td>
<td>70</td>
</tr>
<tr>
<td>Configuring Traffic Engineering Tunnel Bandwidth</td>
<td>70</td>
</tr>
<tr>
<td>Confirming DiffServ-TE Bandwidth</td>
<td>70</td>
</tr>
<tr>
<td>Configuring MPLS O-UNI Bandwidth</td>
<td>71</td>
</tr>
<tr>
<td>Enabling Graceful Restart</td>
<td>72</td>
</tr>
<tr>
<td>Configuring ACL-based Prefix Filtering</td>
<td>73</td>
</tr>
<tr>
<td>Configuring ACLs for Prefix Filtering</td>
<td>73</td>
</tr>
</tbody>
</table>
Configuring RSVP Packet Dropping  74
Verifying RSVP Configuration  75
Enabling RSVP Traps  78
How to Implement RSVP Authentication  79
  Configuring Global Configuration Mode RSVP Authentication  79
    Enabling RSVP Authentication Using the Keychain in Global Configuration Mode  79
    Configuring a Lifetime for RSVP Authentication in Global Configuration Mode  80
    Configuring the Window Size for RSVP Authentication in Global Configuration Mode  81
  Configuring an Interface for RSVP Authentication  82
    Specifying the RSVP Authentication Keychain in Interface Mode  82
    Configuring a Lifetime for an Interface for RSVP Authentication  83
    Configuring the Window Size for an Interface for RSVP Authentication  84
  Configuring RSVP Neighbor Authentication  85
    Specifying the Keychain for RSVP Neighbor Authentication  85
    Configuring a Lifetime for RSVP Neighbor Authentication  86
    Configuring the Window Size for RSVP Neighbor Authentication  87
Verifying the Details of the RSVP Authentication  88
Eliminating Security Associations for RSVP Authentication  89
Configuration Examples for RSVP  89
  Bandwidth Configuration (Prestandard): Example  89
  Bandwidth Configuration (MAM): Example  89
  Bandwidth Configuration (RDM): Example  90
  Refresh Reduction and Reliable Messaging Configuration: Examples  90
    Refresh Interval and the Number of Refresh Messages Configuration: Example  90
    Retransmit Time Used in Reliable Messaging Configuration: Example  90
    Acknowledgement Times Configuration: Example  90
    Summary Refresh Message Size Configuration: Example  91
    Disable Refresh Reduction: Example  91
  Configure Graceful Restart: Examples  91
    Enable Graceful Restart: Example  91
    Enable Interface-Based Graceful Restart: Example  91
    Change the Restart-Time: Example  92
    Change the Hello Interval: Example  92
  Configure ACL-based Prefix Filtering: Example  92
Set DSCP for RSVP Packets: Example  92
Enable RSVP Traps: Example  93
Configuration Examples for RSVP Authentication  93
  RSVP Authentication Global Configuration Mode: Example  93
  RSVP Authentication for an Interface: Example  94
  RSVP Neighbor Authentication: Example  94
  RSVP Authentication by Using All the Modes: Example  95
Additional References  95

CHAPTER 4
Implementing MPLS Forwarding  99
Prerequisites for Implementing Cisco MPLS Forwarding  99
Restrictions for Implementing Cisco MPLS Forwarding  100
Information About Implementing MPLS Forwarding  100
  MPLS Forwarding Overview  100
  Label Switching Functions  100
  Distribution of Label Bindings  101
  MFI Control-Plane Services  101
  MFI Data-Plane Services  102
  Time-to-Live Propagation in Hierarchical MPLS  102
  MPLS Maximum Transmission Unit  102
How to Implement MPLS Forwarding  102
  Configuring the Time-to-Live Propagation in Hierarchical MPLS  102
  Configuring the Size of the Local Label  103
Additional References  104

CHAPTER 5
Implementing MPLS Traffic Engineering  107
Prerequisites for Implementing Cisco MPLS Traffic Engineering  109
Information About Implementing MPLS Traffic Engineering  109
  Overview of MPLS Traffic Engineering  109
  Benefits of MPLS Traffic Engineering  110
  How MPLS-TE Works  110
MPLS Traffic Engineering  111
  Backup AutoTunnels  111
    AutoTunnel Attribute-set  112
    Link Protection  112
Node Protection 113
Backup AutoTunnel Assignment 113
Explicit Paths 114
Periodic Backup Promotion 114
Protocol-Based CLI 115
Differentiated Services Traffic Engineering 115
Prestandard DS-TE Mode 116
IETF DS-TE Mode 116
Bandwidth Constraint Models 116
Maximum Allocation Bandwidth Constraint Model 116
Russian Doll Bandwidth Constraint Model 117
TE Class Mapping 117
Flooding 118
Flooding Triggers 118
Flooding Thresholds 118
Fast Reroute 119
IS-IS IP Fast Reroute Loop-free Alternative 119
MPLS-TE and Fast Reroute over Link Bundles 120
Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE 120
DWDM Transponder Integration 121
GMPLS Benefits 121
GMPLS Support 122
GMPLS Protection and Restoration 122
1:1 LSP Protection 123
Shared Mesh Restoration and M:N Path Protection 123
End-to-end Recovery 123
GMPLS Protection Requirements 123
GMPLS Prerequisites 123
Flexible Name-based Tunnel Constraints 123
MPLS Traffic Engineering Interarea Tunneling 124
Interarea Support 124
Multiarea Support 125
Loose Hop Expansion 126
Loose Hop Reoptimization 126
ABR Node Protection 126
Configuring Numbered and Unnumbered Optical TE Tunnels 188
Configuring an Optical TE Tunnel Using Dynamic Path Option 188
Configuring an Optical TE Tunnel Using Explicit Path Option 191
Configuring LSP Hierarchy 192
Configuring Border Control Model 193
Configuring Path Protection 193
Configuring an LSP 194
Forcing Reversion of the LSP 196
Configuring Flexible Name-based Tunnel Constraints 197
Assigning Color Names to Numeric Values 197
Associating Affinity-Names with TE Links 198
Associating Affinity Constraints for TE Tunnels 199
Configuring IS-IS to Flood MPLS-TE Link Information 200
Configuring an OSPF Area of MPLS-TE 201
Configuring Explicit Paths with ABRs Configured as Loose Addresses 202
Configuring MPLS-TE Forwarding Adjacency 203
Configuring Unequal Load Balancing 204
Setting Unequal Load Balancing Parameters 204
Enabling Unequal Load Balancing 205
Configuring a Path Computation Client and Element 206
Configuring a Path Computation Client 206
Configuring a Path Computation Element Address 207
Configuring PCE Parameters 208
Configuring Policy-based Tunnel Selection 211
Configuring the Automatic Bandwidth 212
Configuring the Collection Frequency 212
Forcing the Current Application Period to Expire Immediately 214
Configuring the Automatic Bandwidth Functions 214
Configuring the Shared Risk Link Groups 217
Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link 217
Creating an Explicit Path With Exclude SRLG 219
Using Explicit Path With Exclude SRLG 220
Creating a Link Protection on Backup Tunnel with SRLG Constraint 222
Creating a Node Protection on Backup Tunnel with SRLG Constraint 225
Configuring Point-to-Multipoint TE 228
Contents

Enabling Multicast Routing on the Router 228
Configuring the Static Group for the Point-to-Multipoint Interface 230
Configuring Destinations for the Tunnel Interface 231
Disabling Destinations 234
Logging Per Destinations for Point-to-Multipoint 236
Enabling Soft-Preemption on a Node 238
Enabling Soft-Preemption on a Tunnel 239
Configuring Attributes within a Path-Option Attribute 240
Configuring Auto-Tunnel Mesh Tunnel ID 241
Configuring Auto-tunnel Mesh Unused Timeout 242
Configuring Auto-Tunnel Mesh Group 243
Configuring Tunnel Attribute-Set Templates 245
Enabling LDP on Auto-Tunnel Mesh 246
Implementing Associated Bidirectional Label Switched Paths 247
    Signaling Methods and Object Association for Bidirectional LSPs 248
    Associated Bidirectional Non Co-routed and Co-routed LSPs 249
    Configure Associated Bidirectional Co-routed LSPs 251
Path Protection 253
    Configure Path Protection for Associated Bidirectional LSPs 253
OAM Support for Associated Bidirectional LSPs 255
    Generate Fault OAM Messages at Mid-point 256
    Generate Fault OAM Messages at End-point 256
Pseudowire Call Admission Control 257
Configuration Examples for Cisco MPLS-TE 257
    Configure Fast Reroute and SONET APS: Example 257
    Build MPLS-TE Topology and Tunnels: Example 258
    Configure IETF DS-TE Tunnels: Example 259
    Configure MPLS-TE and Fast-Reroute on OSPF: Example 259
    Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example 260
    Configure GMPLS: Example 260
    Configure Flexible Name-based Tunnel Constraints: Example 262
    Configure an Interarea Tunnel: Example 264
    Configure Forwarding Adjacency: Example 264
    Configure Unequal Load Balancing: Example 265
    Configure PCE: Example 266
Configure Policy-based Tunnel Selection: Example 267
Configure Tunnels for Path Protection: Example 267
Configure Tunnels for Explicit Path Protection: Example 267
Configure Tunnels for Co-existence of Path Protection with Fast Reroute: Example 268
Configure Automatic Bandwidth: Example 268
Configure the MPLS-TE Shared Risk Link Groups: Example 268
Configure the MPLS-TE Auto-Tunnel Backup: Example 270
Configure Point-to-Multipoint TE: Examples 277
  Configure Point-to-Multipoint for the Source: Example 277
  Configure the Point-to-Multipoint Tunnel: Example 278
  Disable a Destination: Example 279
  Configure the Point-to-Multipoint Solution: Example 279
Additional References 282

CHAPTER 6
Implementing GMPLS UNI 285
  Prerequisites for Implementing GMPLS UNI 285
  Restrictions for Implementing GMPLS UNI 286
  Information About Implementing GMPLS UNI 286
    GMPLS UNI vs GMPLS NNI 286
    GMPLS LSP Signaling 286
      Path Message without an ERO 287
      XRO Attribute-set 287
      Connection Diversity 287
    DWDM Transponder Integration 287
  How to Implement GMPLS UNI 288
    Configuring TE for GMPLS UNI 288
      Enabling GMPLS UNI Submode 288
      Configuring GMPLS UNI Controller 289
    Configuring GMPLS UNI Controller as a Tunnel Head 290
    Configuring Other Tunnel Properties for a GMPLS UNI Tunnel 291
    Configuring LSP Diversity 293
      Configuring XRO Attribute-set 293
      Configuring Connection Diversity 294
    Configuring LMP for GMPLS UNI 295
    Configuring Optical Router ID 295
Preface

The preface contains these sections:

- Changes to This Document, page xv
- Obtaining Documentation and Submitting a Service Request, page xv

Changes to This Document

This table lists the technical changes made to this document since it was first printed.

Table 1: Changes to This Document

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>OL-28406-02</td>
<td>May 2013</td>
<td>Republished with documentation updates for Cisco IOS XR Release 4.3.1 features.</td>
</tr>
<tr>
<td>OL-28406-01</td>
<td>December 2012</td>
<td>Initial release of this document.</td>
</tr>
</tbody>
</table>

Obtaining Documentation and Submitting a Service Request

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

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.
New and Changed Feature Information in Cisco IOS XR Release 4.3.x

This table summarizes the new and changed feature information for the Cisco IOS XR MPLS Configuration Guide for the Cisco CRS Router, and tells you where they are documented.

For a complete list of new and changed features in Cisco IOS XR Software, Release 4.3.x, see the New and Changed Features in Cisco IOS XR Software, Release 4.3.x for Cisco CRS Router document.

- New and Changed MPLS Feature Information, page 1
- New and Changed Feature Information in Cisco IOS XR Release 4.3.x, page 2

### New and Changed MPLS Feature Information

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associated Bidirectional Label Switched Paths</td>
<td>This feature was introduced.</td>
<td>Release 5.2.0</td>
<td>Implementing MPLS Traffic Engineering chapter: Implementing Associated Bidirectional Label Switched Paths, on page 247</td>
</tr>
</tbody>
</table>
# New and Changed Feature Information in Cisco IOS XR Release 4.3.x

Table 2: New and Changed Features in Cisco IOS XR Software

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS OAM Support for BGP 3107</td>
<td>This feature was introduced.</td>
<td>Release 4.3.1</td>
<td>Implementing MPLS OAM chapter: MPLS OAM Support for BGP 3107, on page 309</td>
</tr>
<tr>
<td>Policy-Based Tunnel Selection</td>
<td>This feature was introduced.</td>
<td>Release 4.3.1</td>
<td>Implementing MPLS Traffic Engineering chapter: Policy-Based Tunnel Selection, on page 130</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Refer MPLS Traffic Engineering Commands chapter in Cisco IOS XR MPLS Command Reference for the Cisco CRS Router for information on the commands used for configuring Label Security for BGP Inter-AS Option-B.</td>
</tr>
<tr>
<td>GMPLS UNI</td>
<td>This feature was introduced.</td>
<td>Release 4.3.0</td>
<td>Implementing GMPLS UNI chapter:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Implementing GMPLS UNI, on page 285</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Refer GMPLS UNI Commands chapter in Cisco IOS XR MPLS Command Reference for the Cisco CRS Router for information on the commands used for configuring GMPLS UNI.</td>
</tr>
</tbody>
</table>
CHAPTER 2

Implementing MPLS Label Distribution Protocol

The Multiprotocol Label Switching (MPLS) is a standards-based solution driven by the Internet Engineering Task Force (IETF) that was devised to convert the Internet and IP backbones from best-effort networks into business-class transport mediums.

MPLS, with its label switching capabilities, eliminates the need for an IP route look-up and creates a virtual circuit (VC) switching function, allowing enterprises the same performance on their IP-based network services as with those delivered over traditional networks such as Frame Relay or ATM.

Label Distribution Protocol (LDP) performs label distribution in MPLS environments. LDP provides the following capabilities:

- LDP performs hop-by-hop or dynamic path setup; it does not provide end-to-end switching services.
- LDP assigns labels to routes using the underlying Interior Gateway Protocols (IGP) routing protocols.
- LDP provides constraint-based routing using LDP extensions for traffic engineering.

Finally, LDP is deployed in the core of the network and is one of the key protocols used in MPLS-based Layer 2 and Layer 3 virtual private networks (VPNs).

Feature History for Implementing MPLS LDP

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 3.2</td>
<td>Support was added for conceptual and configuration information about LDP label advertisement control (Outbound label filtering).</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>Support was added for these features:</td>
</tr>
<tr>
<td></td>
<td>• Inbound Label Filtering</td>
</tr>
<tr>
<td></td>
<td>• Local Label Allocation Control</td>
</tr>
<tr>
<td></td>
<td>• Session Protection</td>
</tr>
<tr>
<td></td>
<td>• LDP-IGP Synchronization</td>
</tr>
<tr>
<td>Release 3.5.0</td>
<td>Support was added for LDP Auto-configuration.</td>
</tr>
</tbody>
</table>
Prerequisites for Implementing Cisco MPLS LDP

These prerequisites are required to implement MPLS LDP:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must be running Cisco IOS XR software.
- You must install a composite mini-image and the MPLS package.
- You must activate IGP.
- We recommend to use a lower session holdtime bandwidth such as neighbors so that a session down occurs before an adjacency-down on a neighbor. Therefore, the following default values for the hello times are listed:
  - Holdtime is 15 seconds.
  - Interval is 5 seconds.

For example, the LDP session holdtime can be configured as 30 seconds by using the `holdtime` command.
Information About Implementing Cisco MPLS LDP

To implement MPLS LDP, you should understand these concepts:

**Overview of Label Distribution Protocol**

LDP performs label distribution in MPLS environments. LDP uses hop-by-hop or dynamic path setup, but does not provide end-to-end switching services. Labels are assigned to routes that are chosen by the underlying IGP routing protocols. The Label Switched Paths (LSPs) that result from the routes, forward labeled traffic across the MPLS backbone to adjacent nodes.

**Label Switched Paths**

LSPs are created in the network through MPLS. They can be created statically, by RSVP traffic engineering (TE), or by LDP. LSPs created by LDP perform hop-by-hop path setup instead of an end-to-end path.

**LDP Control Plane**

The control plane enables label switched routers (LSRs) to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information.

**Related Topics**

- Configuring LDP Discovery Parameters, on page 18
- Configuring LDP Discovery Over a Link, on page 20
- Configuring LDP Link: Example, on page 48
- Configuring LDP Discovery for Active Targeted Hellos, on page 22
- Configuring LDP Discovery for Passive Targeted Hellos, on page 24
- Configuring LDP Discovery for Targeted Hellos: Example, on page 48

**Exchanging Label Bindings**

LDP creates LSPs to perform the hop-by-hop path setup so that MPLS packets can be transferred between the nodes on the MPLS network.
This figure illustrates the process of label binding exchange for setting up LSPs.

**Figure 1: Setting Up Label Switched Paths**

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

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

**Related Topics**

- Setting Up LDP Neighbors, on page 28
- Configuring LDP Neighbors: Example, on page 49

**LDP Forwarding**

Once label bindings are learned, the LDP control plane is ready to setup the MPLS forwarding plane as shown in the following figure.
Once label bindings are learned, the LDP control plane is ready to setup the MPLS forwarding plane as shown in this figure.

Figure 2: Forwarding Setup

1. Because R3 is next hop for 10.0.0.0 as notified by the FIB, R1 selects label binding from R3 and installs forwarding entry (Layer 1, Layer 3).
2. Because R3 is next hop for 10.0.0.0 (as notified by FIB), R2 selects label binding from R3 and installs forwarding entry (Layer 2, Layer 3).
3. Because R4 is next hop for 10.0.0.0 (as notified by FIB), R3 selects label binding from R4 and installs forwarding entry (Layer 3, Layer 4).
4. Because next hop for 10.0.0.0 (as notified by FIB) is beyond R4, R4 uses NO-LABEL as the outbound and installs the forwarding entry (Layer 4); the outbound packet is forwarded IP-only.
5. Incoming IP traffic on ingress LSR R1 gets label-imposed and is forwarded as an MPLS packet with label L3.
6. Incoming IP traffic on ingress LSR R2 gets label-imposed and is forwarded as an MPLS packet with label L3.
7. R3 receives an MPLS packet with label L3, looks up in the MPLS label forwarding table and switches this packet as an MPLS packet with label L4.
8. R4 receives an MPLS packet with label L4, looks up in the MPLS label forwarding table and finds that it should be Unlabeled, pops the top label, and passes it to the IP forwarding plane.
9. IP forwarding takes over and forwards the packet onward.

Related Topics

Setting Up LDP Forwarding, on page 30
Configuring LDP Forwarding: Example, on page 50
LDP (Label Distribution Protocol) graceful restart provides a control plane mechanism to ensure high availability and allows detection and recovery from failure conditions while preserving Nonstop Forwarding (NSF) services. Graceful restart is a way to recover from signaling and control plane failures without impacting forwarding.

Without LDP graceful restart, when an established session fails, the corresponding forwarding states are cleaned immediately from the restarting and peer nodes. In this case LDP forwarding restarts from the beginning, causing a potential loss of data and connectivity.

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

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

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

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

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

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

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

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

For more information about LDP commands, see *MPLS Label Distribution Protocol Commands* module of the *Cisco IOS XR MPLS Command Reference for the Cisco CRS Router*.

**Related Topics**
- Setting Up LDP NSF Using Graceful Restart, on page 31
- Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 50

**Control Plane Failure**

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

**Figure 3: Control Plane Failure**

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

**Phases in Graceful Restart**

The graceful restart mechanism is divided into different phases:
Control communication failure detection

Control communication failure is detected when the system detects either:

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

Forwarding state maintenance during failure

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

Control state recovery

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

Related Topics

Setting Up LDP NSF Using Graceful Restart, on page 31
Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 50
Recovery with Graceful-Restart

This figure illustrates the process of failure recovery using graceful restart.

**Figure 4: Recovering with Graceful Restart**

1. The router R4 LSR control plane restarts.
2. With the control plane restart, LIB is gone but forwarding states installed by R4’s LDP control plane are not immediately deleted but are marked as stale.
3. Any in-transit packets from R3 to R4 (still labeled with L4) arrive at R4.
4. The MPLS forwarding plane at R4 performs a successful lookup for the local label L4 as forwarding is still intact. The packet is forwarded accordingly.
5. The router R3 LDP peer detects the failure of the control plane and channel and deletes the label bindings from R4. The peer, however, does not delete the corresponding forwarding states but marks them as stale.
6. At this point there are no forwarding disruptions.
7. The peer also starts the neighbor reconnect timer using the reconnect time value.
8. The established LSPs going toward the router R4 are still intact, and there are no broken LSPs.

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

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

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

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

Related Topics
- Setting Up LDP NSF Using Graceful Restart, on page 31
- Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 50

Label Advertisement Control (Outbound Filtering)

By default, LDP advertises labels for all the prefixes to all its neighbors. When this is not desirable (for scalability and security reasons), you can configure LDP to perform outbound filtering for local label advertisement for one or more prefixes to one more peers. This feature is known as *LDP outbound label filtering*, or *local label advertisement control*.

Related Topics
- Configuring Label Advertisement Control (Outbound Filtering), on page 26
- Configuring Label Advertisement (Outbound Filtering): Example, on page 49

Label Acceptance Control (Inbound Filtering)

By default, LDP accepts labels (as remote bindings) for all prefixes from all peers. LDP operates in liberal label retention mode, which instructs LDP to keep remote bindings from all peers for a given prefix. For security reasons, or to conserve memory, you can override this behavior by configuring label binding acceptance for set of prefixes from a given peer.

The ability to filter remote bindings for a defined set of prefixes is also referred to as *LDP inbound label filtering*.

**Note**

Inbound filtering can also be implemented using an outbound filtering policy; however, you may not be able to implement this system if an LDP peer resides under a different administration domain. When both inbound and outbound filtering options are available, we recommend that you use outbound label filtering.

Related Topics
- Configuring Label Acceptance Control (Inbound Filtering), on page 33
- Configuring Label Acceptance (Inbound Filtering): Example, on page 50
Local Label Allocation Control

By default, LDP allocates local labels for all prefixes that are not Border Gateway Protocol (BGP) prefixes. This is acceptable when LDP is used for applications other than Layer 3 virtual private networks (L3VPN) core transport. When LDP is used to set up transport LSPs for L3VPN traffic in the core, it is not efficient or even necessary to allocate and advertise local labels for, potentially, thousands of IGP prefixes. In such a case, LDP is typically required to allocate and advertise local label for loopback /32 addresses for PE routers. This is accomplished using LDP local label allocation control, where an access list can be used to limit allocation of local labels to a set of prefixes. Limiting local label allocation provides several benefits, including reduced memory usage requirements, fewer local forwarding updates, and fewer network and peer updates.

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

Related Topics

- Configuring Local Label Allocation Control, on page 35
- Configuring Local Label Allocation Control: Example, on page 51

Session Protection

When a link comes up, IP converges earlier and much faster than MPLS LDP and may result in MPLS traffic loss until MPLS convergence. If a link flaps, the LDP session will also flap due to loss of link discovery. LDP session protection minimizes traffic loss, provides faster convergence, and protects existing LDP (link) sessions by means of "parallel" source of targeted discovery hello. An LDP session is kept alive and neighbor label bindings are maintained when links are down. Upon reestablishment of primary link adjacencies, MPLS convergence is expedited as LDP need not relearn the neighbor label bindings.

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

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

**Figure 5: Session Protection**

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

---

**Note**

Related Topics

- Configuring Session Protection, on page 36
- Configuring LDP Session Protection: Example, on page 51

**IGP Synchronization**

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

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

In the event of an LDP graceful restart session disconnect, a session is treated as converged as long as the graceful restart neighbor is timed out. Additionally, upon local LDP restart, a checkpointed recovered LDP graceful restart session is used and treated as converged and is given an opportunity to connect and resynchronize.

Under certain circumstances, it might be required to delay declaration of resynchronization to a configurable interval. LDP provides a configuration option to delay declaring synchronization up for up to 60 seconds. LDP communicates this information to IGP upon linkup or session down events.
The configuration for LDP IGP synchronization resides in respective IGPs (OSPF and IS-IS) and there is no LDP-specific configuration for enabling of this feature. However, there is a specific LDP configuration for IGP sync delay timer.

### Related Topics
- Configuring LDP IGP Synchronization: OSPF, on page 36
- Configuring LDP IGP Synchronization—OSPF: Example, on page 51
- Configuring LDP IGP Synchronization: ISIS, on page 37
- Configuring LDP IGP Synchronization—ISIS: Example, on page 52
- Configuring LDP IGP Synchronization Delay Interval, on page 39

### IGP Auto-configuration

To enable LDP on a large number of interfaces, IGP auto-configuration lets you automatically configure LDP on all interfaces associated with a specified IGP interface; for example, when LDP is used for transport in the core network. However, there needs to be one IGP set up to enable LDP auto-configuration.

Typically, LDP assigns and advertises labels for IGP routes and must often be enabled on all active interfaces by an IGP. Without IGP auto-configuration, you must define the set of interfaces under LDP, a procedure that is time-intensive and error-prone.

---

**Note**  
LDP auto-configuration is supported for IPv4 unicast family in the default VRF. The IGP is responsible for verifying and applying the configuration.

You can also disable auto-configuration on a per-interface basis. This permits LDP to enable all IGP interfaces except those that are explicitly disabled and prevents LDP from enabling an interface when LDP auto-configuration is configured under IGP.

### Related Topics
- Enabling LDP Auto-Configuration for a Specified OSPF Instance, on page 40
- Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance, on page 42
- Disabling LDP Auto-Configuration, on page 43
- Configuring LDP Auto-Configuration: Example, on page 52

### IGP Synchronization Process Restart Delay

In the LDP IGP synchronization process, failures and restarts bear a heavy stress on the network. Multiple IGP synchronization notifications from LDP to IGP, and potential generation of multiple SPF and LSAs are known to effect the CPU load considerably. This results in considerable traffic loss when the LDP process fails.

The LDP IGP Synchronization Process Restart Delay is a feature that enables a process-level delay for synchronization events when the LDP fails or restarts. This delay defers the sending of sync-up events to the IGP until most or all the LDP sessions converge and also allows the LDP to stabilize. This allows the LDP
process failure to be less stressful, since IGPs receive all the sync-up events in one bulk. This means that IGP is required to run the SPF and LSAs only one time with an overall view of the sync-up events.

By default the IGP Synchronization Process Restart Delay is disabled and can be enabled by running the configuration command **mpls ldp igp sync delay on-proc-restart**.

### Related Topics
- Configuring LDP IGP Synchronization Process Restart Delay, on page 39

#### LDP Nonstop Routing

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

A disruption in service may include any of these events:

- Route processor (RP) or distributed route processor (DRP) failover
- LDP process restart
- In-service system upgrade (ISSU)
- Minimum disruption restart (MDR)

Unlike graceful restart functionality, LDP NSR does not require protocol extensions and does not force software upgrades on other routers in the network, nor does LDP NSR require peer routers to support NSR.

Process failures of active TCP or LDP results in session loss and, as a result, NSR cannot be provided unless RP switchover is configured as a recovery action. For more information about how to configure switchover as a recovery action for NSR, see Configuring Transports module in *Cisco IOS XR IP Addresses and Services Configuration Guide for the Cisco CRS Router*.

### Related Topics
- Configuring LDP Nonstop Routing, on page 44

#### IP LDP Fast Reroute Loop Free Alternate

The IP Fast Reroute is a mechanism that enables a router to rapidly switch traffic, after an adjacent link failure, node failure, or both, towards a pre-programmed loop-free alternative (LFA) path. This LFA path is used to switch traffic until the router installs a new primary next hop again, as computed for the changed network topology.

The goal of LFA FRR is to reduce failure reaction time to 50 milliseconds by using a pre-computed alternate next hop, in the event that the currently selected primary next hop fails, so that the alternate can be rapidly used when the failure is detected.
This feature targets to address the fast convergence ability by detecting, computing, updating or enabling prefix independent pre-computed alternate loop-free paths at the time of failure.

IGP pre-computes a backup path per IGP prefix. IGP selects one and only one backup path per primary path. RIB installs the best path and download path protection information to FIB by providing correct annotation for protected and protecting paths. FIB pre-installs the backup path in dataplane. Upon the link or node failure, the routing protocol detects the failure, all the backup paths of the impacted prefixes are enabled in a prefix-independent manner.

Prerequisites

The Label Distribution Protocol (LDP) can use the loop-free alternates as long as these prerequisites are met:

The Label Distribution Protocol (LDP) can use the loop-free alternates as long as these prerequisites are met:

1. The Label Distribution Protocol (LDP) must be configured on the routers involved in the LFA computation.
2. The routing protocol must support LDP for loop-free alternates.
3. The interface must be configured as an LDP-enabled interface.
4. The IP Fast Reroute (FRR) feature must be enabled.

There are two approaches in computing LFAs:

- **Link-based (per-link)**—In link-based LFAs, all prefixes reachable through the primary (protected) link share the same backup information. This means that the whole set of prefixes, sharing the same primary, also share the repair or fast reroute (FRR) ability. The per-link approach protects only the next hop address. The per-link approach is suboptimal and not the best for capacity planning. This is because all traffic is redirected to the next hop instead of being spread over multiple paths, which may lead to potential congestion on link to the next hop. The per-link approach does not provide support for node protection.

- **Prefix-based (per-prefix)**—Prefix-based LFAs allow computing backup information per prefix. It protects the destination address. The per-prefix approach is the preferred approach due to its greater applicability, and the greater protection and better bandwidth utilization that it offers.

| Note | The repair or backup information computed for a given prefix using prefix-based LFA may be different from the computed by link-based LFA. |

The per-prefix LFA approach is preferred for LDP IP Fast Reroute LFA for these reasons:

- Better node failure resistance
- Better capacity planning and coverage

Features Not Supported

These interfaces and features are not supported for the IP LDP Fast Reroute Loop Free Alternate feature:

- BVI interface (IRB) is not supported either as primary or backup path.
- GRE tunnel is not supported either as primary or backup path.
- In a multi-topology scenario, the route in topology T can only use LFA within topology T. Hence, the availability of a backup path depends on the topology.

For more information about configuring the IP Fast Reroute Loop-free alternate, see Implementing IS-IS on Cisco IOS XR Software module of the *Cisco IOS XR Routing Configuration Guide for the Cisco CRS Router*. 

Implementing MPLS Label Distribution Protocol

IP LDP Fast Reroute Loop Free Alternate
Downstream on Demand

This Downstream on demand feature adds support for downstream-on-demand mode, where the label is not advertised to a peer, unless the peer explicitly requests it. At the same time, since the peer does not automatically advertise labels, the label request is sent whenever the next-hop points out to a peer that no remote label has been assigned.

To enable downstream-on-demand mode, this configuration must be applied at mpls ldp configuration mode:

```
mpls ldp downstream-on-demand with ACL
```

The ACL contains a list of peer IDs that are configured for downstream-on-demand mode. When the ACL is changed or configured, the list of established neighbors is traversed. If a session's downstream-on-demand configuration has changed, the session is reset in order that the new down-stream-on-demand mode can be configured. The reason for resetting the session is to ensure that the labels are properly advertised between the peers. When a new session is established, the ACL is verified to determine whether the session should negotiate for downstream-on-demand mode. If the ACL does not exist or is empty, downstream-on-demand mode is not configured for any neighbor.

For it to be enabled, the Downstream on demand feature has to be configured on both peers of the session. If only one peer in the session has downstream-on-demand feature configured, then the session does not use downstream-on-demand mode.

If, after, a label request is sent, and no remote label is received from the peer, the router will periodically resend the label request. After the peer advertises a label after receiving the label request, it will automatically readvertise the label if any label attribute changes subsequently.

Related Topics
- Configuring LDP Downstream on Demand mode, on page 45

How to Implement MPLS LDP

A typical MPLS LDP deployment requires coordination among several global neighbor routers. Various configuration tasks are required to implement MPLS LDP:

Configuring LDP Discovery Parameters

Perform this task to configure LDP discovery parameters (which may be crucial for LDP operations).

Note
The LDP discovery mechanism is used to discover or locate neighbor nodes.
## SUMMARY STEPS

1. `configure`
2. `mpls ldp`
3. `[vrf vrf-name] router-id ip-address lsr-id`
4. `discovery { hello | targeted-hello } holdtime seconds`
5. `discovery { hello | targeted-hello } interval seconds`
6. `commit`
7. `(Optional) show mpls ldp [vrf vrf-name] parameters`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>mpls ldp</th>
<th>(Optional) Specifies a non-default VRF. Specifies the router ID of the local node.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td>In Cisco IOS XR software, the router ID is specified as an interface IP address. By default, LDP uses the global router ID (configured by the global router ID process).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>[vrf vrf-name] router-id ip-address lsr-id</th>
<th>(Optional) Specifies a non-default VRF. Specifies the router ID of the local node.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td>In Cisco IOS XR software, the router ID is specified as an interface IP address. By default, LDP uses the global router ID (configured by the global router ID process).</td>
</tr>
</tbody>
</table>

| Step 4 | discovery { hello | targeted-hello } holdtime seconds | Specifies the time that a discovered neighbor is kept without receipt of any subsequent hello messages. The default value for the seconds argument is 15 seconds for link hello and 90 seconds for targeted hello messages. |
|--------|-----------------------------------|-----------------------------------------------------------|
| Example:|                                   | In Cisco IOS XR software, the router ID is specified as an interface IP address. By default, LDP uses the global router ID (configured by the global router ID process). |

| Step 5 | discovery { hello | targeted-hello } interval seconds | Selects the period of time between the transmission of consecutive hello messages. The default value for the seconds argument is 5 seconds for link hello messages and 10 seconds for targeted hello messages. |
|--------|-----------------------------------|-----------------------------------------------------------|
| Example:|                                   | In Cisco IOS XR software, the router ID is specified as an interface IP address. By default, LDP uses the global router ID (configured by the global router ID process). |

<p>| Step 6 | commit | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7&lt;br&gt;show mpls ldp [vrf vrf-name] parameters</td>
<td>(Optional) Displays all the current MPLS LDP parameters. Displays the LDP parameters for the specified VRF.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router
# show mpls ldp parameters

RP/0/RP0/CPU0:router
# show mpls ldp vrf red parameters
```

Related Topics

LDP Control Plane, on page 5

Configuring LDP Discovery Over a Link

Perform this task to configure LDP discovery over a link.

**Note**

There is no need to enable LDP globally.

**Before You Begin**

A stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. [vrf vrf-name] router-id ip-address lsr-id
4. interface type interface-path-id
5. commit
6. (Optional) show mpls ldp discovery
7. (Optional) show mpls ldp vrf vrf-name discovery
8. (Optional) show mpls ldp vrf all discovery summary
9. (Optional) show mpls ldp vrf all discovery brief
10. (Optional) show mpls ldp vrf all ipv4 discovery summary
11. (Optional) show mpls ldp discovery summary all
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>mpls ldp</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>[vrf vrf-name] router-id ip-address lsr-id</td>
<td>(Optional) Specifies a non-default VRF. Specifies the router ID of the local node.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1</td>
<td>In Cisco IOS XR software, the router ID is specified as an interface name or IP address. By default, LDP uses the global router ID (configured by the global router ID process).</td>
</tr>
<tr>
<td>4</td>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode for the LDP protocol. Interface type must be Tunnel-TE.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001 RP/0/RP0/CPU0:router(config-ldp-if)#</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>show mpls ldp discovery</td>
<td>(Optional) Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer’s LDP identifier, and holdtime values.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp discovery</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>show mpls ldp vrf vrf-name discovery</td>
<td>(Optional) Displays the status of the LDP discovery process for the specified VRF.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>show mpls ldp vrf all discovery summary</td>
<td>(Optional) Displays the summarized status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring LDP Discovery for Active Targeted Hellos

Perform this task to configure LDP discovery for active targeted hellos.

**Note**
The active side for targeted hellos initiates the unicast hello toward a specific destination.

**Before You Begin**
These prerequisites are required to configure LDP discovery for active targeted hellos:

- Stable router ID is required at either end of the targeted session. If you do not assign a router ID to the routers, the system will default to the global router ID. Please note that default router IDs are subject to change and may cause an unstable discovery.

- One or more MPLS Traffic Engineering tunnels are established between non-directly connected LSRs.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong></td>
<td></td>
</tr>
<tr>
<td>show mpls ldp vrf all discovery brief</td>
<td>(Optional) Displays the brief status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td></td>
</tr>
<tr>
<td>show mpls ldp vrf all ipv4 discovery summary</td>
<td>(Optional) Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td></td>
</tr>
<tr>
<td>show mpls ldp discovery summary all</td>
<td>(Optional) Displays the aggregate summary across all the LDP discovery processes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls ldp discovery summary all</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- LDP Control Plane, on page 5
- Configuring LDP Link: Example, on page 48
SUMMARY STEPS

1. configure
2. mpls ldp
3. [vrf vrf-name] router-id ip-address lsr-id
4. interface type interface-path-id
5. commit
6. (Optional) show mpls ldp discovery
7. (Optional) show mpls ldp vrf vrf-name discovery
8. (Optional) show mpls ldp vrf all discovery summary
9. (Optional) show mpls ldp vrf all discovery brief
10. (Optional) show mpls ldp vrf all ipv4 discovery summary
11. (Optional) show mpls ldp discovery summary all

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>mpls ldp</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
</tr>
<tr>
<td>3</td>
<td>[vrf vrf-name] router-id ip-address lsr-id</td>
<td>(Optional) Specifies a non-default VRF. Specifies the router ID of the local node.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1</td>
</tr>
<tr>
<td>4</td>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode for the LDP protocol.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001</td>
</tr>
<tr>
<td>5</td>
<td>commit</td>
<td>(Optional) Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer’s LDP identifier, and holdtime values.</td>
</tr>
<tr>
<td>6</td>
<td>show mpls ldp discovery</td>
<td>(Optional) Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer’s LDP identifier, and holdtime values.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# show mpls ldp discovery</td>
</tr>
</tbody>
</table>
## Configuring LDP Discovery for Passive Targeted Hellos

Perform this task to configure LDP discovery for passive targeted hellos.

A passive side for targeted hello is the destination router (tunnel tail), which passively waits for an incoming hello message. Because targeted hellos are unicast, the passive side waits for an incoming hello message to respond with hello toward its discovered neighbor.

### Command or Action

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command: <code>show mpls ldp vrf vrf-name discovery</code></th>
<th>Purpose: (Optional) Displays the status of the LDP discovery process for the specified VRF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command: <code>show mpls ldp vrf all discovery summary</code></th>
<th>Purpose: (Optional) Displays the summarized status of the LDP discovery process for all VRFs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command: <code>show mpls ldp vrf all discovery brief</code></th>
<th>Purpose: (Optional) Displays the brief status of the LDP discovery process for all VRFs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command: <code>show mpls ldp vrf all ipv4 discovery summary</code></th>
<th>Purpose: (Optional) Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th>Command: <code>show mpls ldp discovery summary all</code></th>
<th>Purpose: (Optional) Displays the aggregate summary across all the LDP discovery processes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show mpls ldp discovery summary all</code></td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- LDP Control Plane, on page 5
- Configuring LDP Discovery for Targeted Hellos: Example, on page 48
**Before You Begin**

Stable router ID is required at either end of the link to ensure that the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. [vrf vrf-name] router-id ip-address lsr-id
4. discovery targeted-hello accept
5. commit
6. (Optional) show mpls ldp discovery
7. (Optional) show mpls ldp vrf vrf-name discovery
8. (Optional) show mpls ldp vrf all discovery summary
9. (Optional) show mpls ldp vrf all discovery brief
10. (Optional) show mpls ldp vrf all ipv4 discovery summary
11. (Optional) show mpls ldp discovery summary all

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>mpls ldp</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>[vrf vrf-name] router-id ip-address lsr-id</td>
<td>(Optional) Specifies a non-default VRF. Specifies the router ID of the local node.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>discovery targeted-hello accept</td>
<td>Directs the system to accept targeted hello messages from any source and activates passive mode on the LSR for targeted hello acceptance.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-ldp)# discovery targeted-hello accept</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
## Implementing MPLS Label Distribution Protocol

### Configuring Label Advertisement Control (Outbound Filtering)

Perform this task to configure label advertisement (outbound filtering).

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show mpls ldp discovery</td>
<td>(Optional) Displays the status of the LDP discovery process. This command, without an interface filter, generates a list of interfaces over which the LDP discovery process is running. The output information contains the state of the link (xmt/rcv hellos), local LDP identifier, the discovered peer’s LDP identifier, and holdtime values.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp discovery</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show mpls ldp vrf vrf-name discovery</td>
<td>(Optional) Displays the status of the LDP discovery process for the specified VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf [vrf-name] discovery</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show mpls ldp vrf all discovery summary</td>
<td>(Optional) Displays the summarized status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show mpls ldp vrf all discovery brief</td>
<td>(Optional) Displays the brief status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show mpls ldp vrf all ipv4 discovery summary</td>
<td>(Optional) Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show mpls ldp discovery summary all</td>
<td>(Optional) Displays the aggregate summary across all the LDP discovery processes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls ldp discovery summary all</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics

- LDP Control Plane, on page 5
- Configuring LDP Discovery for Targeted Hellos: Example, on page 48
By default, a label switched router (LSR) advertises all incoming label prefixes to each neighboring router. You can control the exchange of label binding information using the `mpls ldp label advertise` command. Using the optional keywords, you can advertise selective prefixes to all neighbors, advertise selective prefixes to defined neighbors, or disable label advertisement to all peers for all prefixes.

---

**Note**

Prefixes and peers advertised selectively are defined in the access list.

---

**Before You Begin**

Before configuring label advertisement, enable LDP and configure an access list.

---

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. label advertise { disable | for prefix-acl [ to peer-acl ] | interface type interface-path-id }
4. commit

---

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls ldp</td>
<td>Configures label advertisement by specifying one of the following options: disable</td>
</tr>
<tr>
<td>Step 3</td>
<td>label advertise { disable</td>
<td>for prefix-acl [ to peer-acl ]</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
Setting Up LDP Neighbors

Perform this task to set up LDP neighbors.

Before You Begin
Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

SUMMARY STEPS

1. configure
2. mpls ldp
3. interface type interface-path-id
4. discovery transport-address [ ip-address | interface ]
5. exit
6. holdtime seconds
7. neighbor ip-address password [ encryption ] password
8. backoff initial maximum
9. commit
10. (Optional) show mpls ldp neighbor

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls ldp</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode for the LDP protocol.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-ldp)# interface POS 0/1/0/0</td>
<td></td>
</tr>
</tbody>
</table>
| Step 4 | discovery transport-address [ ip-address | interface ] | Provides an alternative transport address for a TCP connection.  
  - Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID.  
  - Transport address configuration is applied for a given LDP-enabled interface.  
  - If the interface version of the command is used, the configured IP address of the interface is passed to its neighbors as the transport address.  
  
  **Example:**  
  - or  
  - RP/0/RP0/CPU0:router(config-ldp-if-af)# discovery transport-address interface  

| Step 5 | exit | Exits the current configuration mode.  
  
  **Example:**  
  - RP/0/RP0/CPU0:router(config-ldp-if)# exit  

| Step 6 | holdtime seconds | Changes the time for which an LDP session is maintained in the absence of LDP messages from the peer.  
  - Outgoing keepalive interval is adjusted accordingly (to make three keepalives in a given holdtime) with a change in session holdtime value.  
  - Session holdtime is also exchanged when the session is established.  
  - In this example holdtime is set to 30 seconds, which causes the peer session to timeout in 30 seconds, as well as transmitting outgoing keepalive messages toward the peer every 10 seconds.  
  
  **Example:**  
  - RP/0/RP0/CPU0:router(config-ldp)# holdtime 30  

| Step 7 | neighbor ip-address password [ encryption ] password | Configures password authentication (using the TCP MD5 option) for a given neighbor.  
  
  **Example:**  
  - RP/0/RP0/CPU0:router(config-ldp)# neighbor 192.168.2.44 password secretpasswd  

| Step 8 | backoff initial maximum | Configures the parameters for the LDP backoff mechanism. The LDP backoff mechanism prevents two incompatibly configured LSRs from engaging in an unthrottled sequence of session setup failures. If a session setup attempt fails due to such incompatibility, each LSR delays its next attempt (backs off), increasing the delay exponentially with each successive failure until the maximum backoff delay is reached.  
  
  **Example:**  
  - RP/0/RP0/CPU0:router(config-ldp)# backoff 10 20  

| Step 9 | commit |  
  
  **Note:**  
  - Implementing MPLS Label Distribution Protocol  
  - Setting Up LDP Neighbors
### Setting Up LDP Forwarding

Perform this task to set up LDP forwarding.

By default, the LDP control plane implements the penultimate hop popping (PHOP) mechanism. The PHOP mechanism requires that label switched routers use the implicit-null label as a local label for the given Forwarding Equivalence Class (FEC) for which LSR is the penultimate hop. Although PHOP has certain advantages, it may be required to extend LSP up to the ultimate hop under certain circumstances (for example, to propagate MPL QoS). This is done using a special local label (explicit-null) advertised to the peers after which the peers use this label when forwarding traffic toward the ultimate hop (egress LSR).

### Before You Begin

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.

### SUMMARY STEPS

1. configure
2. mpls ldp
3. explicit-null
4. commit
5. (Optional) show mpls ldp forwarding
6. (Optional) show mpls forwarding
7. (Optional) ping ip-address

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td></td>
</tr>
</tbody>
</table>
### Implementing MPLS Label Distribution Protocol

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td><code>mpls ldp</code></td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config)# mpls ldp</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>explicit-null</code></td>
<td>Causes a router to advertise an explicit null label in situations where it normally advertises an implicit null label (for example, to enable an ultimate-hop disposition instead of PHOP).</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config-ldp-af)# explicit-null</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>show mpls ldp forwarding</code></td>
<td>(Optional) Displays the MPLS LDP view of installed forwarding states (rewrites).</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show mpls ldp forwarding</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>show mpls forwarding</code></td>
<td>(Optional) Displays a global view of all MPLS installed forwarding states (rewrites) by various applications (LDP, TE, and static).</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# show mpls forwarding</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td><code>ping ip-address</code></td>
<td>(Optional) Checks for connectivity to a particular IP address (going through MPLS LSP as shown in the <code>show mpls forwarding</code> command).</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router# ping 192.168.2.55</code></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**
- LDP Forwarding, on page 6
- Configuring LDP Forwarding: Example, on page 50

**Setting Up LDP NSF Using Graceful Restart**

Perform this task to set up NSF using LDP graceful restart.

LDP graceful restart is a way to enable NSF for LDP. The correct way to set up NSF using LDP graceful restart is to bring up LDP neighbors (link or targeted) with additional configuration related to graceful restart.

**Before You Begin**

Stable router ID is required at either end of the link to ensure the link discovery (and session setup) is successful. If you do not assign a router ID to the routers, the system will default to the global router ID. Default router IDs are subject to change and may cause an unstable discovery.
SUMMARY STEPS

1. configure
2. mpls ldp
3. interface  type interface-path-id
4. exit
5. graceful-restart
6. graceful-restart forwarding-state-holdtime  seconds
7. graceful-restart reconnect-timeout  seconds
8. commit
9. (Optional) show mpls ldp parameters
10. (Optional) show mpls ldp neighbor
11. (Optional) show mpls ldp graceful-restart

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls ldp</td>
<td>Enters interface configuration mode for the LDP protocol.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
</tr>
<tr>
<td>Step 3 interface</td>
<td>Enters MPLS LDP configuration mode for the LDP protocol.</td>
</tr>
<tr>
<td>type interface-path-id</td>
<td>RP/0/RP0/CPU0:router(config-ldp)# interface</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp-if)#</td>
</tr>
<tr>
<td>Step 4 exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp-if)# exit</td>
</tr>
<tr>
<td>Step 5 graceful-restart</td>
<td>Enables the LDP graceful restart feature.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp)# graceful-restart</td>
</tr>
<tr>
<td>Step 6 graceful-restart forwarding-state-holdtime</td>
<td>Specifies the length of time that forwarding can keep LDP-installed forwarding states and rewrites, and specifies when the LDP control plane restarts.</td>
</tr>
<tr>
<td>seconds</td>
<td>RP/0/RP0/CPU0:router(config-ldp)#</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp)#</td>
</tr>
<tr>
<td></td>
<td>After restart of the control plane, when the forwarding state holdtime expires, any previously installed LDP forwarding</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| graceful-restart forwarding-state-holdtime 180 | state or rewrite that is not yet refreshed is deleted from the forwarding.  
- Recovery time sent after restart is computed as the current remaining value of the forwarding state hold timer. |

**Step 7**  
**graceful-restart reconnect-timeout** seconds  

**Example:**  
RP/0/RP0/CPU0:router(config-ldp)# graceful-restart reconnect-timeout 169  

Specifies the length of time a neighbor waits before restarting the node to reconnect before declaring an earlier graceful restart session as down. This command is used to start a timer on the peer (upon a neighbor restart). This timer is referred to as Neighbor Liveness timer.

**Step 8**  
**commit**  

**Step 9**  
**show mpls ldp parameters**  

**Example:**  
RP/0/RP0/CPU0:router(config-ldp)# show mpls ldp parameters  

(Optional) Displays all the current MPLS LDP parameters.

**Step 10**  
**show mpls ldp neighbor**  

**Example:**  
RP/0/RP0/CPU0:router# show mpls ldp neighbor  

(Optional) Displays the status of the LDP session with its neighbors. This command can be run with various filters as well as with the brief option.

**Step 11**  
**show mpls ldp graceful-restart**  

**Example:**  
RP/0/RP0/CPU0:router# show mpls ldp graceful-restart  

(Optional) Displays the status of the LDP graceful restart feature. The output of this command not only shows states of different graceful restart timers, but also a list of graceful restart neighbors, their state, and reconnect count.

**Related Topics**  
- LDP Graceful Restart, on page 8  
- Phases in Graceful Restart, on page 9  
- Recovery with Graceful-Restart, on page 11  
- Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 50

**Configuring Label Acceptance Control (Inbound Filtering)**

Perform this task to configure LDP inbound label filtering.
By default, there is no inbound label filtering performed by LDP and thus an LSR accepts (and retains) all remote label bindings from all peers.

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. label accept for *prefix-acl* from *ip-address*
4. [vrf *vrf-name*] address-family { ipv4 }
5. label remote accept from *ldp-id* for *prefix-acl*
6. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls ldp</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>label accept for <em>prefix-acl</em> from <em>ip-address</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Configures inbound label acceptance for prefixes specified by prefix-acl from neighbor (as specified by its IP address).</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp)# label accept for pfx_acl_1 from 192.168.1.1 RP/0/RP0/CPU0:router(config-ldp)# label accept for pfx_acl_2 from 192.168.2.2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>[vrf <em>vrf-name</em>] address-family { ipv4 }</td>
</tr>
<tr>
<td>Example:</td>
<td>(Optional) Specifies a non-default VRF. Enables the LDP IPv4 or IPv6 address family.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4 RP/0/RP0/CPU0:router(config-ldp)# address-family ipv6</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>label remote accept from <em>ldp-id</em> for <em>prefix-acl</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Configures inbound label acceptance control for prefixes specified by prefix-acl from neighbor (as specified by its LDP ID).</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp-af)# label remote accept from 192.168.1.1:0 for pfx_acl_1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>
Related Topics

- Label Acceptance Control (Inbound Filtering), on page 12
- Configuring Label Acceptance (Inbound Filtering): Example, on page 50

Configuring Local Label Allocation Control

Perform this task to configure label allocation control.

Note

By default, local label allocation control is disabled and all non-BGP prefixes are assigned local labels.

SUMMARY STEPS

1. configure
2. mpls ldp
3. label allocate for prefix-acl
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls ldp</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>Step 3 label allocate for prefix-acl</td>
<td>Configures label allocation control for prefixes as specified by prefix-acl.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp)# label allocate for pfx_acl_1</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

- Local Label Allocation Control, on page 13
- Configuring Local Label Allocation Control: Example, on page 51
Configuring Session Protection

Perform this task to configure LDP session protection. By default, there is no protection is done for link sessions by means of targeted hellos.

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. session protection [ for peer-acl ] [ duration seconds ]
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls ldp</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Step 3 session protection [ for peer-acl ] [ duration seconds ]</td>
<td>Configures LDP session protection for peers specified by peer-acl with a maximum duration, in seconds.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

Session Protection, on page 13
Configuring LDP Session Protection: Example, on page 51

**Configuring LDP IGP Synchronization: OSPF**

Perform this task to configure LDP IGP Synchronization under OSPF.

*Note* By default, there is no synchronization between LDP and IGPs.
SUMMARY STEPS

1. configure
2. router ospf process-name
3. Use one of the following commands:
   • mpls ldp sync
   • area area-id mpls ldp sync
   • area area-id interface name mpls ldp sync
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 router ospf process-name</td>
<td>Identifies the OSPF routing process and enters OSPF configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# router ospf 100</td>
<td></td>
</tr>
<tr>
<td>Step 3 Use one of the following commands:</td>
<td>Enables LDP IGP synchronization on an interface.</td>
</tr>
<tr>
<td>• mpls ldp sync</td>
<td></td>
</tr>
<tr>
<td>• area area-id mpls ldp sync</td>
<td></td>
</tr>
<tr>
<td>• area area-id interface name mpls ldp sync</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ospf)# mpls ldp sync</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

IGP Synchronization, on page 14
Configuring LDP IGP Synchronization—OSPF: Example, on page 51

Configuring LDP IGP Synchronization: ISIS

Perform this task to configure LDP IGP Synchronization under ISIS.
By default, there is no synchronization between LDP and ISIS.

**SUMMARY STEPS**

1. configure
2. router isis instance-id
3. interface type interface-path-id
4. address-family {ipv4 } unicast
5. mpls ldp sync
6. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
</tbody>
</table>
**Example:**  
RP/0/RP0/CPU0:router(config)# router isis 100  
RP/0/RP0/CPU0:router(config-isis)# |
| Step 3 | interface type interface-path-id | Configures the IS-IS protocol on an interface and enters ISIS interface configuration mode.  
**Example:**  
RP/0/RP0/CPU0:router(config-isis)# interface POS 0/2/0/0  
RP/0/RP0/CPU0:router(config-isis-if)# |
| Step 4 | address-family {ipv4 } unicast | Enters address family configuration mode for configuring IS-IS routing for a standard IP version 4 (IPv4) address prefix.  
**Example:**  
RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast  
RP/0/RP0/CPU0:router(config-isis-if-af)# |
| Step 5 | mpls ldp sync | Enables LDP IGP synchronization.  
**Example:**  
RP/0/RP0/CPU0:router(config-isis-if-af)# mpls ldp sync |
| Step 6 | commit | |
Configuring LDP IGP Synchronization Delay Interval

Perform this task to configure the LDP IGP synchronization delay interval.
By default, LDP does not delay declaring sync up as soon as convergence conditions are met.

SUMMARY STEPS

1. configure
2. mpls ldp
3. igp sync delay  delay-time
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls ldp</td>
<td>Configures LDP IGP synchronization delay in seconds.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>Step 3 igp sync delay delay-time</td>
<td>Configures LDP IGP synchronization delay in seconds.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ldp)# igp sync delay 30</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

IGP Synchronization, on page 14
Configuring LDP IGP Synchronization—ISIS: Example, on page 52

Configuring LDP IGP Synchronization Process Restart Delay

Perform this task to enable process restart delay when an LDP fails or restarts.
By default, the LDP IGP Synchronization Process Restart Delay feature is disabled.

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. Use one of the following commands:
   • igp sync delay *seconds*
   • igp sync delay on-proc-restart *delay-time*
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls ldp</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
</tr>
<tr>
<td>Step 3</td>
<td>Configures LDP IGP delay in seconds.</td>
</tr>
<tr>
<td>Use one of the following commands:</td>
<td></td>
</tr>
<tr>
<td>• igp sync delay <em>seconds</em></td>
<td></td>
</tr>
<tr>
<td>• igp sync delay on-proc-restart <em>delay-time</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp)# igp sync delay 30</td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

IGP Synchronization Process Restart Delay, on page 15

**Enabling LDP Auto-Configuration for a Specified OSPF Instance**

Perform this task to enable IGP auto-configuration globally for a specified OSPF process name.
You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

**Note**
This feature is supported for IPv4 unicast family in default VRF only.

**SUMMARY STEPS**

1. configure
2. router ospf  process-name
3. mpls ldp auto-config
4. area  area-id
5. interface  type interface-path-id
6. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf  process-name</td>
<td>Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# router ospf 190</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> mpls ldp auto-config</td>
<td>Enables LDP auto-configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ospf)# mpls ldp auto-config</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> area  area-id</td>
<td>Configures an OSPF area and identifier.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ospf)# area 8</td>
<td>Either a decimal value or an IP address.</td>
</tr>
<tr>
<td><strong>Step 5</strong> interface  type interface-path-id</td>
<td>Enables LDP auto-configuration on the specified interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0</td>
<td>LDP configurable limit for maximum number of interfaces does not apply to IGP auto-configuration interfaces.</td>
</tr>
<tr>
<td><strong>Step 6</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>
Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance

Perform this task to enable IGP auto-configuration in a defined area with a specified OSPF process name. You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

**Note**
This feature is supported for IPv4 unicast family in default VRF only.

**SUMMARY STEPS**

1. **configure**
2. **router ospf** *process-name*
3. **area** *area-id*
4. **mpls ldp auto-config**
5. **interface** *type interface-path-id*
6. **commit**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.</td>
</tr>
<tr>
<td><strong>Step 2</strong> router ospf <em>process-name</em></td>
<td>Configures an OSPF area and identifier. <em>area-id</em> Either a decimal value or an IP address.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# router ospf 100 RP/0/RP0/CPU0:router(config-ospf)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> area <em>area-id</em></td>
<td>Enables LDP auto-configuration.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ospf)# area 8 RP/0/RP0/CPU0:router(config-ospf-ar)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> mpls ldp auto-config</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-ospf-ar)# mpls ldp auto-config</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

Enables LDP auto-configuration on the specified interface. The LDP configurable limit for maximum number of interfaces does not apply to IGP auto-config interfaces.

### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Interface type interface-path-id</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ospf-ar-if)</td>
<td></td>
</tr>
</tbody>
</table>

### Step 6

**commit**

---

**Related Topics**

IGP Auto-configuration, on page 15  
Configuring LDP Auto-Configuration: Example, on page 52  
Disabling LDP Auto-Configuration, on page 43

---

### Disabling LDP Auto-Configuration

Perform this task to disable IGP auto-configuration.

You can disable auto-configuration on a per-interface basis. This lets LDP enable all IGP interfaces except those that are explicitly disabled.

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. interface type interface-path-id
4. igp auto-config disable
5. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls ldp</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp)#</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring LDP Nonstop Routing

Perform this task to configure LDP NSR.

**Note**

By default, NSR is globally-enabled on all LDP sessions except AToM.

#### SUMMARY STEPS

1. configure
2. mpls ldp
3. nsr
4. commit
5. (Optional) show mpls ldp nsr statistics
6. (Optional) show mpls ldp nsr summary
7. (Optional) show mpls ldp nsr pending

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
</tbody>
</table>
Purpose

Command or Action | Purpose
--- | ---
**Step 2** | 
**mpls ldp** | Enters the MPLS LDP configuration mode.
*Example:*  
RP/0/RP0/CPU0:router(config)# mpls ldp

**Step 3** | 
**nsr** | Enables LDP nonstop routing.
*Example:*  
RP/0/RP0/CPU0:router(config-ldp)# nsr

**Step 4** | 
**commit** | 

**Step 5** | 
**show mpls ldp nsr statistics** | (Optional)  
Displays MPLS LDP NSR statistics.
*Example:*  
RP/0/RP0/CPU0:router# show mpls ldp nsr statistics

**Step 6** | 
**show mpls ldp nsr summary** | (Optional)  
Displays MPLS LDP NSR summarized information.
*Example:*  
RP/0/RP0/CPU0:router# show mpls ldp nsr summary

**Step 7** | 
**show mpls ldp nsr pending** | (Optional)  
Displays MPLS LDP NSR pending information.
*Example:*  
RP/0/RP0/CPU0:router# show mpls ldp nsr pending

**Related Topics**

LDP Nonstop Routing, on page 16

**Configuring LDP Downstream on Demand mode**

**SUMMARY STEPS**

1. configure
2. mpls ldp
3. [vrf vrf-name session] downstream-on-demand
4. commit
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls ldp</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>[vrf vrf-name session] downstream-on-demand</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-ldp)# vrf red session downstream-on-demand with ABC</td>
</tr>
</tbody>
</table>

Related Topics

Downstream on Demand, on page 18

Redistributing MPLS LDP Routes into BGP

Perform this task to redistribute Border Gateway Protocol (BGP) autonomous system into an MPLS LDP.

SUMMARY STEPS

1. configure
2. mpls ldp
3. redistribute bgp
4. end or commit
5. show run mpls ldp

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>mpls ldp</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls ldp</td>
</tr>
</tbody>
</table>

**Step 3**
redistribute bgp

**Example:**
RP/0/RP0/CPU0:router(config-ldp)# redistribute bgp advertise-to acl_1

**Purpose**
Allows the redistribution of BGP routes into an MPLS LDP processes.

**Note**
Autonomous system numbers (ASNs) are globally unique identifiers used to identify autonomous systems (ASs) and enable ASs to exchange exterior routing information between neighboring ASs. A unique ASN is allocated to each AS for use in BGP routing. ASNs are encoded as 2-byte numbers and 4-byte numbers in BGP.

**Step 4**
end or commit

**Step 5**
show run mpls ldp

**Step 4**
- When you issue the end command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - 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.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
  - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

**Purpose**
Displays information about the redistributed route information.

### Configuration Examples for Implementing MPLS LDP

These configuration examples are provided to implement LDP:
Configuring LDP with Graceful Restart: Example

The example shows how to enable LDP with graceful restart on the POS interface 0/2/0/0.

```
mpls ldp
  graceful-restart
  interface pos0/2/0/0
```

Configuring LDP Discovery: Example

The example shows how to configure LDP discovery parameters.

```
mpls ldp
  router-id 192.168.70.1
  discovery hello holdtime 15
  discovery hello interval 5
```

```
show mpls ldp parameters
show mpls ldp discovery
```

Configuring LDP Link: Example

The example shows how to configure LDP link parameters.

```
mpls ldp
  interface pos 0/1/0/0
```

```
show mpls ldp discovery
```

Related Topics

- Configuring LDP Discovery Over a Link, on page 20
- LDP Control Plane, on page 5

Configuring LDP Discovery for Targeted Hellos: Example

The examples show how to configure LDP Discovery to accept targeted hello messages.

**Active (tunnel head)**

```
mpls ldp
  router-id 192.168.70.1
  interface tunnel-te 12001
```

```
!```
Passive (tunnel tail)

mpls ldp
 router-id 192.168.70.2
discovery targeted-hello accept
!

Related Topics
 Configuring LDP Discovery for Active Targeted Hellos, on page 22
 Configuring LDP Discovery for Passive Targeted Hellos, on page 24
 LDP Control Plane, on page 5

Configuring Label Advertisement (Outbound Filtering): Example

The example shows how to configure LDP label advertisement control.

mpls ldp
 label
  advertise
    disable
    for pfx_acl_1 to peer_acl_1
    for pfx_acl_2 to peer_acl_2
    for pfx_acl_3
    interface POS 0/1/0/0
    interface POS 0/2/0/0
  !
  !
  ipv4 access-list pfx_acl_1
    10 permit ip host 1.0.0.0 any
  !
  ipv4 access-list pfx_acl_2
    10 permit ip host 2.0.0.0 any
  !
  ipv4 access-list peer_acl_1
    10 permit ip host 1.1.1.1 any
    20 permit ip host 1.1.1.2 any
  !
  ipv4 access-list peer_acl_2
    10 permit ip host 2.2.2.2 any
  !
  show mpls ldp binding

Related Topics
 Configuring Label Advertisement Control (Outbound Filtering), on page 26
 Label Advertisement Control (Outbound Filtering), on page 12

Configuring LDP Neighbors: Example

The example shows how to disable label advertisement.

mpls ldp
  router-id 192.168.70.1
  neighbor 1.1.1.1 password encrypted 110A1016141E
  neighbor 2.2.2.2 implicit-withdraw
Configuring LDP Forwarding: Example

The example shows how to configure LDP forwarding.

```
mpls ldp
  address-family ipv4
  label local advertise explicit-null
!
show mpls ldp forwarding
show mpls forwarding
```

Related Topics

- Setting Up LDP Neighbors, on page 28

Configuring LDP Nonstop Forwarding with Graceful Restart: Example

The example shows how to configure LDP nonstop forwarding with graceful restart.

```
mpls ldp
  log graceful-restart
!
  graceful-restart forwarding state-holdtime 180
  graceful-restart reconnect-timeout 15
  interface pos0/1/0/0
!
show mpls ldp graceful-restart
show mpls ldp neighbor gr
show mpls ldp forwarding
show mpls forwarding
```

Related Topics

- Setting Up LDP NSF Using Graceful Restart, on page 31
- LDP Graceful Restart, on page 8
- Phases in Graceful Restart, on page 9
- Recovery with Graceful-Restart, on page 11

Configuring Label Acceptance (Inbound Filtering): Example

The example shows how to configure inbound label filtering.

```
mpls ldp
```
Implementing MPLS Label Distribution Protocol

Configuring Local Label Allocation Control: Example

The example shows how to configure local label allocation control.

```
label
accept
for pfx_acl_2 from 192.168.2.2
!
!
```

```
mpls ldp
  address-family ipv4
  label remote accept from 192.168.1.1:0 for pfx_acl_2
  !
  !
```

Related Topics

- Configuring Label Acceptance Control (Inbound Filtering), on page 33
- Label Acceptance Control (Inbound Filtering), on page 12

Configuring LDP Session Protection: Example

The example shows how to configure session protection.

```
mpls ldp
  session protection for peer_acl_1 duration 60
  !
```

Related Topics

- Configuring Session Protection, on page 36
- Session Protection, on page 13

Configuring LDP IGP Synchronization—OSPF: Example

The example shows how to configure LDP IGP synchronization for OSPF.

```
router ospf 100
```
mpls ldp sync
!
mpls ldp
  igp sync delay 30
!

Related Topics
Configuring LDP IGP Synchronization: OSPF, on page 36
IGP Synchronization, on page 14

Configuring LDP IGP Synchronization—ISIS: Example

The example shows how to configure LDP IGP synchronization.

```
router isis 100
  interface POS 0/2/0/0
  address-family ipv4 unicast
  mpls ldp sync
  !
  !
  mpls ldp
  igp sync delay 30
```

Related Topics
Configuring LDP IGP Synchronization: ISIS, on page 37
IGP Synchronization, on page 14

Configuring LDP Auto-Configuration: Example

The example shows how to configure the IGP auto-configuration feature globally for a specific OSPF interface ID.

```
router ospf 100
  mpls ldp auto-config
  area 0
    interface pos 1/1/1/1
```

The example shows how to configure the IGP auto-configuration feature on a given area for a given OSPF interface ID.

```
router ospf 100
  area 0
    mpls ldp auto-config
    interface pos 1/1/1/1
```

Related Topics
Enabling LDP Auto-Configuration for a Specified OSPF Instance, on page 40
Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance, on page 42
Disabling LDP Auto-Configuration, on page 43
IGP Auto-configuration, on page 15

**Configure IP LDP Fast Reroute Loop Free Alternate: Examples**

This example shows how to configure LFA FRR with default tie-break configuration:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
 metric-style wide

interface GigabitEthernet0/6/0/13
 point-to-point
 address-family ipv4 unicast
 fast-reroute per-prefix
 # primary path GigabitEthernet0/6/0/13 will exclude the interface
 # GigabitEthernet0/6/0/33 in LFA backup path computation.
 fast-reroute per-prefix exclude interface GigabitEthernet0/6/0/33
!
interface GigabitEthernet0/6/0/23
 point-to-point
 address-family ipv4 unicast
!
interface GigabitEthernet0/6/0/24
 point-to-point
 address-family ipv4 unicast
!
interface GigabitEthernet0/6/0/33
 point-to-point
 address-family ipv4 unicast
!
```

This example shows how to configure TE tunnel as LFA backup:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
 metric-style wide

interface GigabitEthernet0/6/0/13
 point-to-point
 address-family ipv4 unicast
 fast-reroute per-prefix
 # primary path GigabitEthernet0/6/0/13 will exclude the interface
 # GigabitEthernet0/6/0/33 in LFA backup path computation. TE tunnel 1001
 # is using the link GigabitEthernet0/6/0/33.
 fast-reroute per-prefix exclude interface GigabitEthernet0/6/0/33
 fast-reroute per-prefix lfa-candidate interface tunnel-te1001
!
interface GigabitEthernet0/6/0/33
 point-to-point
 address-family ipv4 unicast
!
```

This example shows how to configure LFA FRR with configurable tie-break configuration:

```
router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
 metric-style wide
 fast-reroute per-prefix tiebreaker ?
 downstream Prefer backup path via downstream node
  lc-disjoint Prefer line card disjoint backup path
  lowest-backup-metric  Prefer backup path with lowest total metric
```
node-protecting  Prefer node protecting backup path
primary-path  Prefer backup path from ECMP set
secondary-path  Prefer non-ECMP backup path

fast-reroute per-prefix tiebreaker lc-disjoint index ?
<1-255>  Index
fast-reroute per-prefix tiebreaker lc-disjoint index 10

Sample configuration:

router isis TEST
net 49.0001.0000.0000.0001.00
address-family ipv4 unicast
metric-style wide
fast-reroute per-prefix tiebreaker downstream index 60
fast-reroute per-prefix tiebreaker lc-disjoint index 10
fast-reroute per-prefix tiebreaker lowest-backup-metric index 40
fast-reroute per-prefix tiebreaker node-protecting index 30
fast-reroute per-prefix tiebreaker primary-path index 20
fast-reroute per-prefix tiebreaker secondary-path index 50

interface GigabitEthernet0/6/0/13
point-to-point
address-family ipv4 unicast
fast-reroute per-prefix

interface GigabitEthernet0/1/0/13
point-to-point
address-family ipv4 unicast
fast-reroute per-prefix

interface GigabitEthernet0/3/0/0.1
point-to-point
address-family ipv4 unicast

interface GigabitEthernet0/3/0/0.2
point-to-point
address-family ipv4 unicast

Related Topics

IP LDP Fast Reroute Loop Free Alternate, on page 16

Verify IP LDP Fast Reroute Loop Free Alternate: Example

The following examples show how to verify the IP LDP FRR LFA feature on the router. The following example shows how to verify ISIS FRR output:

RP/0/RP0/CPU0:router#show isis fast-reroute summary

IS-IS 1 IPv4 Unicast FRR summary

<table>
<thead>
<tr>
<th>Critical Priority</th>
<th>High Priority</th>
<th>Medium Priority</th>
<th>Low Priority</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefixes reachable in L1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All paths protected</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1008</td>
</tr>
<tr>
<td>Some paths protected</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unprotected</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Protection coverage</td>
<td>0.00%</td>
<td>0.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Prefixes reachable in L2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All paths protected</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Some paths protected</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unprotected</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Protection coverage</td>
<td>0.00%</td>
<td>0.00%</td>
<td>100.00%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Cisco IOS XR MPLS Configuration Guide for the Cisco CRS Router, Release 4.3.x
The following example shows how to verify the IGP route 211.1.1.1/24 in ISIS Fast Reroute output:

```
RP/0/RP0/CPU0:router#show isis fast-reroute 211.1.1.1/24
L1 211.1.1.1/24 [40/115]
  via 12.0.0.2, GigabitEthernet0/6/0/13, NORTH
  FRR backup via 14.0.2.2, GigabitEthernet0/6/0/0.3, SOUTH

RP/0/RP0/CPU0:router#show isis fast-reroute 211.1.1.1/24 detail
L1 211.1.1.1/24 [40/115] low priority
  via 12.0.0.2, GigabitEthernet0/6/0/13, NORTH
  FRR backup via 14.0.2.2, GigabitEthernet0/6/0/0.3, SOUTH
  P: No, TM: 130, LC: No, NP: Yes, D: Yes
  src sr1.00-00, 173.1.1.2
  L2 adv [40] native, propagated
```

The following example shows how to verify the IGP route 211.1.1.1/24 in RIB output:

```
RP/0/RP0/CPU0:router#show route 211.1.1.1/24
Routing entry for 211.1.1.0/24
  Known via "isis 1", distance 115, metric 40, type level-1
  Installed Nov 27 10:22:20.311 for 1d08h
  Routing Descriptor Blocks
    12.0.0.2, from 173.1.1.2, via GigabitEthernet0/6/0/13, Protected
    Route metric is 40
    14.0.2.2, from 173.1.1.2, via GigabitEthernet0/6/0/0.3, Backup
    Route metric is 0
  No advertising protos.
```

The following example shows how to verify the IGP route 211.1.1.1/24 in FIB output:

```
RP/0/RP0/CPU0:router#show cef 211.1.1.1/24
211.1.1.0/24, version 0, internal 0x40040001 (ptr 0x9d9e1a68) [1], 0x0
(0x9ce0ec40), 0x4500 (0x9e2c69e4)
  Updated Nov 27 10:22:29.825
  remote adjacency to GigabitEthernet0/6/0/13
  Prefix Len 24, traffic index 0, precedence routine (0)
  gateway array (0x9cc622f0) reference count 1158, flags 0x28000d00, source lsd [2],
    [387 type 5 flags 0x101001 (0x9df32398) ext 0x0 (0x0)]
    LW-LDI[type=5, refc=3, ptr=0x9ce0ec40, sh-ldi=0x9df32398]
    via 12.0.0.2, GigabitEthernet0/6/0/13, 0 dependencies, weight 0, class 0,
  protected [flags 0x400]
    path-idx 0, bkup-idx 1 [0x9e5b71b4 0x0]
    hop 12.0.0.2
    local label 16080 labels imposed {16082}
    via 14.0.2.2, GigabitEthernet0/6/0/0.3, 3 dependencies, weight 0, class 0,
  backup [flags 0x300]
    path-idx 1
    hop 14.0.2.2
    remote adjacency
    local label 16080 labels imposed {16079}

RP/0/RP0/CPU0:router#show cef 211.1.1.1/24 detail
211.1.1.0/24, version 0, internal 0x40040001 (ptr 0x9d9e1a68) [1], 0x0
(0x9ce0ec40), 0x4500 (0x9e2c69e4)
  Updated Nov 27 10:22:29.825
  remote adjacency to GigabitEthernet0/6/0/13
  Prefix Len 24, traffic index 0, precedence routine (0)
  gateway array (0x9cc622f0) reference count 1158, flags 0x28000d00, source lsd [2],
    [387 type 5 flags 0x101001 (0x9df32398) ext 0x0 (0x0)]
    LW-LDI[type=5, refc=3, ptr=0x9ce0ec40, sh-ldi=0x9df32398]
    via 12.0.0.2, GigabitEthernet0/6/0/13, 0 dependencies, weight 0, class 0,
  protected [flags 0x400]
    path-idx 0, bkup-idx 1 [0x9e5b71b4 0x0]
    hop 12.0.0.2
    local label 16080 labels imposed {16082}
    via 14.0.2.2, GigabitEthernet0/6/0/0.3, 3 dependencies, weight 0, class 0,
  backup [flags 0x300]
```
The following examples show how to verify the IGP route 211.1.1.1/24 in MPLS LDP output:

RP/0/RP0/CPU0:router#show mpls ldp forwarding 211.1.1.1/24

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Label</th>
<th>Label</th>
<th>Outgoing</th>
<th>Next Hop</th>
<th>GR</th>
<th>Stale</th>
</tr>
</thead>
<tbody>
<tr>
<td>211.1.1.0/24</td>
<td>16080</td>
<td>16082</td>
<td>Gi0/6/0/13</td>
<td>12.0.0.2</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16079</td>
<td>Gi0/6/0/0.3</td>
<td>14.0.2.2 (!)</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

RP/0/RP0/CPU0:router#show mpls ldp forwarding 211.1.1.1/24 detail

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Label</th>
<th>Label</th>
<th>Outgoing</th>
<th>Next Hop</th>
<th>GR</th>
<th>Stale</th>
</tr>
</thead>
<tbody>
<tr>
<td>211.1.1.0/24</td>
<td>16080</td>
<td>16082</td>
<td>Gi0/6/0/13</td>
<td>12.0.0.2</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16079</td>
<td>Gi0/6/0/0.3</td>
<td>14.0.2.2 (!)</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

Routing update: Nov 27 10:22:19.560 (1d08h ago)
Forwarding update: Nov 27 10:22:29.060 (1d08h ago)

Related Topics

IP LDP Fast Reroute Loop Free Alternate, on page 16

Additional References

For additional information related to Implementing MPLS Label Distribution Protocol, refer to the following references:

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
</table>
## Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

## MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

## RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Not all supported RFCs are listed.</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 3031</td>
<td>Multiprotocol Label Switching Architecture</td>
<td></td>
</tr>
<tr>
<td>RFC 3036</td>
<td>LDP Specification</td>
<td></td>
</tr>
<tr>
<td>RFC 3037</td>
<td>LDP Applicability</td>
<td></td>
</tr>
<tr>
<td>RFC 3478</td>
<td>Graceful Restart Mechanism for Label Distribution Protocol</td>
<td></td>
</tr>
<tr>
<td>RFC 3815</td>
<td>Definitions of Managed Objects for MPLS LDP</td>
<td></td>
</tr>
<tr>
<td>RFC 5036</td>
<td>Label Distribution and Management Downstream on Demand Label Advertisement</td>
<td></td>
</tr>
<tr>
<td>RFC 5286</td>
<td>Basic Specification for IP Fast Reroute: Loop-Free Alternates</td>
<td></td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
Implementing RSVP for MPLS-TE and MPLS O-UNI

The Multiprotocol Label Switching (MPLS) is a standards-based solution, driven by the Internet Engineering Task Force (IETF), devised to convert the Internet and IP backbones from best-effort networks into business-class transport media.

Resource Reservation Protocol (RSVP) is a signaling protocol that enables systems to request resource reservations from the network. RSVP processes protocol messages from other systems, processes resource requests from local clients, and generates protocol messages. As a result, resources are reserved for data flows on behalf of local and remote clients. RSVP creates, maintains, and deletes these resource reservations. RSVP provides a secure method to control quality-of-service (QoS) access to a network.

MPLS Traffic Engineering (MPLS-TE) and MPLS Optical User Network Interface (MPLS O-UNI) use RSVP to signal label switched paths (LSPs).

Feature History for Implementing RSVP for MPLS-TE and MPLS O-UNI

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 3.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.2</td>
<td>Support was added for ACL-based prefix filtering.</td>
</tr>
<tr>
<td>Release 3.4.1</td>
<td>Support was added for RSVP authentication.</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>The RSVP MIB feature was added.</td>
</tr>
</tbody>
</table>

- Prerequisites for Implementing RSVP for MPLS-TE and MPLS O-UNI, page 60
- Information About Implementing RSVP for MPLS-TE and MPLS O-UNI, page 60
- Information About Implementing RSVP Authentication, page 65
- How to Implement RSVP, page 70
Prerequisites for Implementing RSVP for MPLS-TE and MPLS O-UNI

These prerequisites are required to implement RSVP for MPLS-TE and MPLS O-UNI:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Either a composite mini-image plus an MPLS package, or a full image, must be installed.

Information About Implementing RSVP for MPLS-TE and MPLS O-UNI

To implement MPLS RSVP, you must understand these concepts:

Related Topics

- How to Implement RSVP Authentication, on page 79

Overview of RSVP for MPLS-TE and MPLS O-UNI

RSVP is a network control protocol that enables Internet applications to signal LSPs for MPLS-TE, and LSPs for O-UNI. The RSVP implementation is compliant with the IETF RFC 2205, RFC 3209, and OIF2000.125.7.

When configuring an O-UNI LSP, the RSVP session is bidirectional. The exchange of data between a pair of machines actually constitutes a single RSVP session. The RSVP session is established using an Out-Of-Band (OOB) IP Control Channel (IPCC) with the neighbor. The RSVP messages are transported over an interface other than the data interface.

RSVP supports extensions according to OIF2000.125.7 requirements, including:

- Generalized Label Request
- Generalized UNI Attribute
- UNI Session
- New Error Spec sub-codes

RSVP is automatically enabled on interfaces on which MPLS-TE is configured. For MPLS-TE LSPs with nonzero bandwidth, the RSVP bandwidth has to be configured on the interfaces. There is no need to configure
RSVP, if all MPLS-TE LSPs have zero bandwidth. For O-UNI, there is no need for any RSVP configuration.

RSVP Refresh Reduction, defined in RFC 2961, includes support for reliable messages and summary refresh messages. Reliable messages are retransmitted rapidly if the message is lost. Because each summary refresh message contains information to refresh multiple states, this greatly reduces the amount of messaging needed to refresh states. For refresh reduction to be used between two routers, it must be enabled on both routers. Refresh Reduction is enabled by default.

Message rate limiting for RSVP allows you to set a maximum threshold on the rate at which RSVP messages are sent on an interface. Message rate limiting is disabled by default.

The process that implements RSVP is restartable. A software upgrade, process placement or process failure of RSVP or any of its collaborators, has been designed to ensure Nonstop Forwarding (NSF) of the data plane.

RSVP supports graceful restart, which is compliant with RFC 3473. It follows the procedures that apply when the node reestablishes communication with the neighbor’s control plane within a configured restart time.

It is important to note that RSVP is not a routing protocol. RSVP works in conjunction with routing protocols and installs the equivalent of dynamic access lists along the routes that routing protocols calculate. Because of this, implementing RSVP in an existing network does not require migration to a new routing protocol.

### Related Topics
- Configuring RSVP Packet Dropping, on page 74
- Set DSCP for RSVP Packets: Example, on page 92
- Verifying RSVP Configuration, on page 75

## LSP Setup

LSP setup is initiated when the LSP head node sends path messages to the tail node (see the RSVP Operation figure).

This figure illustrates an LSP setup for non-O-UNI applications. In the case of an O-UNI application, the RSVP signaling messages are exchanged on a control channel, and the corresponding data channel to be used is acquired from the LMP Manager module based on the control channel. Also the O-UNI LSP’s are by default bidirectional while the MPLS-TE LSP’s are uni-directional.

### Figure 6: RSVP Operation

![RSVP Operation Diagram](image)

The Path messages reserve resources along the path to each node, creating Path soft states on each node. When the tail node receives a path message, it sends a reservation (RESV) message with a label back to the previous node. When the reservation message arrives at the previous node, it causes the reserved resources to be locked and forwarding entries are programmed with the MPLS label sent from the tail-end node. A new MPLS label is allocated and sent to the next node upstream.
When the reservation message reaches the head node, the label is programmed and the MPLS data starts to flow along the path.

**High Availability**

RSVP is designed to ensure nonstop forwarding under the following constraints:

- Ability to tolerate the failure of one or more MPLS/O-UNI processes.
- Ability to tolerate the failure of one RP of a 1:1 redundant pair.
- Hitless software upgrade.

The RSVP high availability (HA) design follows the constraints of the underlying architecture where processes can fail without affecting the operation of other processes. A process failure of RSVP or any of its collaborators does not cause any traffic loss or cause established LSPs to go down. When RSVP restarts, it recovers its signaling states from its neighbors. No special configuration or manual intervention are required. You may configure RSVP graceful restart, which offers a standard mechanism to recover RSVP state information from neighbors after a failure.

**Graceful Restart**

RSVP graceful restart provides a control plane mechanism to ensure high availability (HA), which allows detection and recovery from failure conditions while preserving nonstop forwarding services on the systems running Cisco IOS XR software.

RSVP graceful restart provides a mechanism that minimizes the negative effects on MPLS traffic caused by these types of faults:

- Disruption of communication channels between two nodes when the communication channels are separate from the data channels. This is called *control channel failure*.
- Control plane of a node fails but the node preserves its data forwarding states. This is called *node failure*.

The procedure for RSVP graceful restart is described in the "Fault Handling" section of RFC 3473, *Generalized MPLS Signaling, RSVP-TE Extensions*. One of the main advantages of using RSVP graceful restart is recovery of the control plane while preserving nonstop forwarding and existing labels.

**Graceful Restart: Standard and Interface-Based**

When you configure RSVP graceful restart, Cisco IOS XR software sends and expects node-id address based Hello messages (that is, Hello Request and Hello Ack messages). The RSVP graceful restart Hello session is not established if the neighbor router does not respond with a node-id based Hello Ack message.

You can also configure graceful restart to respond (send Hello Ack messages) to interface-address based Hello messages sent from a neighbor router in order to establish a graceful restart Hello session on the neighbor router. If the neighbor router does not respond with node-id based Hello Ack message, however, the RSVP graceful restart Hello session is not established.

Cisco IOS XR software provides two commands to configure graceful restart:

- `signalling hello graceful-restart`
• signalling hello graceful-restart interface-based

**Note**

By default, graceful restart is disabled. To enable interface-based graceful restart, you must first enable standard graceful restart. You cannot enable interface-based graceful restart independently.

---

**Related Topics**

- Enabling Graceful Restart, on page 72
- Enable Graceful Restart: Example, on page 91
- Enable Interface-Based Graceful Restart: Example, on page 91

---

**Graceful Restart: Figure**

This figure illustrates how RSVP graceful restart handles a node failure condition.

**Figure 7: Node Failure with RSVP**

RSVP graceful restart requires the use of RSVP hello messages. Hello messages are used between RSVP neighbors. Each neighbor can autonomously issue a hello message containing a hello request object. A receiver that supports the hello extension replies with a hello message containing a hello acknowledgment (ACK) object. This means that a hello message contains either a hello Request or a hello ACK object. These two objects have the same format.

The restart cap object indicates a node's restart capabilities. It is carried in hello messages if the sending node supports state recovery. The restart cap object has the following two fields:

**Restart Time**

Time after a loss in Hello messages within which RSVP hello session can be reestablished. It is possible for a user to manually configure the Restart Time.
Recovery Time

Time that the sender waits for the recipient to re-synchronize states after the re-establishment of hello messages. This value is computed and advertised based on number of states that existed before the fault occurred.

For graceful restart, the hello messages are sent with an IP Time to Live (TTL) of 64. This is because the destination of the hello messages can be multiple hops away. If graceful restart is enabled, hello messages (containing the restart cap object) are sent to an RSVP neighbor when RSVP states are shared with that neighbor.

Restart cap objects are sent to an RSVP neighbor when RSVP states are shared with that neighbor. If the neighbor replies with hello messages containing the restart cap object, the neighbor is considered to be graceful restart capable. If the neighbor does not reply with hello messages or replies with hello messages that do not contain the restart cap object, RSVP backs off sending hellos to that neighbor. If graceful restart is disabled, no hello messages (Requests or ACKs) are sent. If a hello Request message is received from an unknown neighbor, no hello ACK is sent back.

ACL-based Prefix Filtering

RSVP provides for the configuration of extended access lists (ACLs) to forward, drop, or perform normal processing on RSVP router-alert (RA) packets. Prefix filtering is designed for use at core access routers in order that RA packets (identified by a source/destination address) can be seamlessly forwarded across the core from one access point to another (or, conversely to be dropped at this node). RSVP applies prefix filtering rules only to RA packets because RA packets contain source and destination addresses of the RSVP flow.

Note

RA packets forwarded due to prefix filtering must not be sent as RSVP bundle messages, because bundle messages are hop-by-hop and do not contain RA. Forwarding a Bundle message does not work, because the node receiving the messages is expected to apply prefix filtering rules only to RA packets.

For each incoming RSVP RA packet, RSVP inspects the IP header and attempts to match the source/destination IP addresses with a prefix configured in an extended ACL. The results are as follows:

- If an ACL does not exist, the packet is processed like a normal RSVP packet.
- If the ACL match yields an explicit permit (and if the packet is not locally destined), the packet is forwarded. The IP TTL is decremented on all forwarded packets.
- If the ACL match yields an explicit deny, the packet is dropped.

If there is no explicit permit or explicit deny, the ACL infrastructure returns an implicit (default) deny. RSVP can be configured to drop the packet. By default, RSVP processes the packet if the ACL match yields an implicit (default) deny.

Related Topics

Configuring ACLs for Prefix Filtering, on page 73
Configure ACL-based Prefix Filtering: Example, on page 92
RSVP MIB

RFC 2206, RSVP Management Information Base Using SMiv2 defines all the SNMP MIB objects that are relevant to RSVP. By implementing the RSVP MIB, you can perform these functions:

- Specifies two traps (NetFlow and LostFlow) which are triggered when a new flow is created or deleted.
- Lets you use SNMP to access objects belonging to RSVP.

Related Topics

Enabling RSVP Traps, on page 78
Enable RSVP Traps: Example, on page 93

Information About Implementing RSVP Authentication

Before implementing RSVP authentication, you must configure a keychain first. The name of the keychain must be the same as the one used in the keychain configuration. For more information about configuring keychains, see Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router.

Note
RSVP authentication supports only keyed-hash message authentication code (HMAC) type algorithms.

To implement RSVP authentication on Cisco IOS XR software, you must understand the following concepts:

RSVP Authentication Functions

You can carry out these tasks with RSVP authentication:

- Set up a secure relationship with a neighbor by using secret keys that are known only to you and the neighbor.
- Configure RSVP authentication in global, interface, or neighbor configuration modes.
- Authenticate incoming messages by checking if there is a valid security relationship that is associated based on key identifier, incoming interface, sender address, and destination address.
- Add an integrity object with message digest to the outgoing message.
- Use sequence numbers in an integrity object to detect replay attacks.

RSVP Authentication Design

Network administrators need the ability to establish a security domain to control the set of systems that initiates RSVP requests.

The RSVP authentication feature permits neighbors in an RSVP network to use a secure hash to sign all RSVP signaling messages digitally, thus allowing the receiver of an RSVP message to verify the sender of the message without relying solely on the sender's IP address.
The signature is accomplished on a per-RSVP-hop basis with an RSVP integrity object in the RSVP message as defined in RFC 2747. This method provides protection against forgery or message modification. However, the receiver must know the security key used by the sender to validate the digital signature in the received RSVP message.

Network administrators manually configure a common key for each RSVP neighbor on the shared network.

The following reasons explain how to choose between global, interface, or neighbor configuration modes:

- **Global configuration mode** is optimal when a router belongs to a single security domain (for example, part of a set of provider core routers). A single common key set is expected to be used to authenticate all RSVP messages.

- **Interface, or neighbor configuration mode**, is optimal when a router belongs to more than one security domain. For example, a provider router is adjacent to the provider edge (PE), or a PE is adjacent to an edge device. Different keys can be used but not shared.

Global configuration mode configures the defaults for interface and neighbor interface modes. These modes, unless explicitly configured, inherit the parameters from global configuration mode, as follows:

- Window-size is set to 1.
- Lifetime is set to 1800.
- The **key-source key-chain** command is set to none or disabled.

**Related Topics**

- Configuring a Lifetime for an Interface for RSVP Authentication, on page 83
- RSVP Authentication by Using All the Modes: Example, on page 95

**Global, Interface, and Neighbor Authentication Modes**

You can configure global defaults for all authentication parameters including key, window size, and lifetime. These defaults are inherited when you configure authentication for each neighbor or interface. However, you can also configure these parameters individually on a neighbor or interface basis, in which case the global values (configured or default) are no longer inherited.

**Note**

RSVP uses the following rules when choosing which authentication parameter to use when that parameter is configured at multiple levels (interface, neighbor, or global). RSVP goes from the most specific to least specific; that is, neighbor, interface, and global.

Global keys simplify the configuration and eliminate the chances of a key mismatch when receiving messages from multiple neighbors and multiple interfaces. However, global keys do not provide the best security.

Interface keys are used to secure specific interfaces between two RSVP neighbors. Because many of the RSVP messages are IP routed, there are many scenarios in which using interface keys are not recommended. If all keys on the interfaces are not the same, there is a risk of a key mismatch for the following reasons:

- When the RSVP graceful restart is enabled, RSVP hello messages are sent with a source IP address of the local router ID and a destination IP address of the neighbor router ID. Because multiple routes can exist between the two neighbors, the RSVP hello message can traverse to different interfaces.
When the RSVP fast reroute (FRR) is active, the RSVP Path and Resv messages can traverse multiple interfaces.

When Generalized Multiprotocol Label Switching (GMPLS) optical tunnels are configured, RSVP messages are exchanged with router IDs as the source and destination IP addresses. Since multiple control channels can exist between the two neighbors, the RSVP messages can traverse different interfaces.

Neighbor-based keys are particularly useful in a network in which some neighbors support RSVP authentication procedures and others do not. When the neighbor-based keys are configured for a particular neighbor, you are advised to configure all the neighbor’s addresses and router IDs for RSVP authentication.

**Related Topics**

- Configuring a Lifetime for RSVP Authentication in Global Configuration Mode, on page 80
- RSVP Authentication Global Configuration Mode: Example, on page 93
- Specifying the RSVP Authentication Keychain in Interface Mode, on page 82
- RSVP Authentication by Using All the Modes: Example, on page 95

## Security Association

A security association (SA) is defined as a collection of information that is required to maintain secure communications with a peer to counter replay attacks, spoofing, and packet corruption.

This table lists the main parameters that define a security association.

**Table 3: Security Association Main Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src</td>
<td>IP address of the sender.</td>
</tr>
<tr>
<td>dst</td>
<td>IP address of the final destination.</td>
</tr>
<tr>
<td>interface</td>
<td>Interface of the SA.</td>
</tr>
<tr>
<td>direction</td>
<td>Send or receive type of the SA.</td>
</tr>
<tr>
<td>Lifetime</td>
<td>Expiration timer value that is used to collect unused security association data.</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>Last sequence number that was either sent or accepted (dependent of the direction type).</td>
</tr>
<tr>
<td>key-source</td>
<td>Source of keys for the configurable parameter.</td>
</tr>
<tr>
<td>keyID</td>
<td>Key number (returned from the key-source) that was last used.</td>
</tr>
<tr>
<td>digest</td>
<td>Algorithm last used (returned from the key-source).</td>
</tr>
</tbody>
</table>
### Security Association

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window Size</td>
<td>Specifies the tolerance for the configurable parameter. The parameter is applicable when the direction parameter is the receive type.</td>
</tr>
<tr>
<td>Window</td>
<td>Specifies the last <code>window size</code> value sequence number that is received or accepted. The parameter is applicable when the direction parameter is the receive type.</td>
</tr>
</tbody>
</table>

An SA is created dynamically when sending and receiving messages that require authentication. The neighbor, source, and destination addresses are obtained either from the IP header or from an RSVP object, such as a HOP object, and whether the message is incoming or outgoing.

When the SA is created, an expiration timer is created. When the SA authenticates a message, it is marked as recently used. The lifetime timer periodically checks if the SA is being used. If so, the flag is cleared and is cleaned up for the next period unless it is marked again.

This table shows how to locate the source and destination address keys for an SA that is based on the message type.

### Table 4: Source and Destination Address Locations for Different Message Types

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Source Address Location</th>
<th>Destination Address Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>HOP object</td>
<td>SESSION object</td>
</tr>
<tr>
<td>PathTear</td>
<td>HOP object</td>
<td>SESSION object</td>
</tr>
<tr>
<td>PathError</td>
<td>HOP object</td>
<td>IP header</td>
</tr>
<tr>
<td>Resv</td>
<td>HOP object</td>
<td>IP header</td>
</tr>
<tr>
<td>ResvTear</td>
<td>HOP object</td>
<td>IP header</td>
</tr>
<tr>
<td>ResvError</td>
<td>HOP object</td>
<td>IP header</td>
</tr>
<tr>
<td>ResvConfirm</td>
<td>IP header</td>
<td>CONFIRM object</td>
</tr>
<tr>
<td>Ack</td>
<td>IP header</td>
<td>IP header</td>
</tr>
<tr>
<td>Srefresh</td>
<td>IP header</td>
<td>IP header</td>
</tr>
<tr>
<td>Hello</td>
<td>IP header</td>
<td>IP header</td>
</tr>
<tr>
<td>Bundle</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Key-source Key-chain

The key-source key-chain is used to specify which keys to use.

You configure a list of keys with specific IDs and have different lifetimes so that keys are changed at predetermined intervals automatically, without any disruption of service. Rollover enhances network security by minimizing the problems that could result if an untrusted source obtained, deduced, or guessed the current key.

RSVP handles rollover by using the following key ID types:

- On TX, use the youngest eligible key ID.
- On RX, use the key ID that is received in an integrity object.

For more information about implementing keychain management, see Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router.

Guidelines for Window-Size and Out-of-Sequence Messages

These guidelines are required for window-size and out-of-sequence messages:

- Default window-size is set to 1. If a single message is received out-of-sequence, RSVP rejects it and displays a message.
- When RSVP messages are sent in burst mode (for example, tunnel optimization), some messages can become out-of-sequence for a short amount of time.
- Window size can be increased by using the `window-size` command. When the window size is increased, replay attacks can be detected with duplicate sequence numbers.
Caveats for Out-of-Sequence

These caveats are listed for out-of-sequence:

• When RSVP messages traverse multiple interface types with different maximum transmission unit (MTU) values, some messages can become out-of-sequence if they are fragmented.

• Packets with some IP options may be reordered.

• Change in QoS configurations may lead to a transient reorder of packets.

• QoS policies can cause a reorder of packets in a steady state.

Because all out-of-sequence messages are dropped, the sender must retransmit them. Because RSVP state timeouts are generally long, out-of-sequence messages during a transient state do not lead to a state timeout.

How to Implement RSVP

RSVP requires coordination among several routers, establishing exchange of RSVP messages to set up LSPs. Depending on the client application, RSVP requires some basic configuration, as described in these topics:

Configuring Traffic Engineering Tunnel Bandwidth

To configure traffic engineering tunnel bandwidth, you must first set up TE tunnels and configure the reserved bandwidth per interface (there is no need to configure bandwidth for the data channel or the control channel).

Cisco IOS XR software supports two MPLS DS-TE modes: Prestandard and IETF.

Note

For prestandard DS-TE you do not need to configure bandwidth for the data channel or the control channel. There is no other specific RSVP configuration required for this application. When no RSVP bandwidth is specified for a particular interface, you can specify zero bandwidth in the LSP setup if it is configured under RSVP interface configuration mode or MPLS-TE configuration mode.

Related Topics

- Configuring a Prestandard DS-TE Tunnel, on page 165
- Configuring an IETF DS-TE Tunnel Using RDM, on page 167
- Configuring an IETF DS-TE Tunnel Using MAM, on page 169

Confirming DiffServ-TE Bandwidth

Perform this task to confirm DiffServ-TE bandwidth.

In RSVP global and subpools, reservable bandwidths are configured per interface to accommodate TE tunnels on the node. Available bandwidth from all configured bandwidth pools is advertised using IGP. RSVP signals the TE tunnel with appropriate bandwidth pool requirements.
SUMMARY STEPS

1. configure
2. rsvp
3. interface type interface-path-id
4. bandwidth total-bandwidth max-flow sub-pool sub-pool-bw
5. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 rsvp</td>
<td>Enters RSVP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# rsvp</td>
<td></td>
</tr>
<tr>
<td>Step 3 interface type interface-path-id</td>
<td>Enters interface configuration mode for the RSVP protocol.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp)# interface pos 0/2/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4 bandwidth total-bandwidth max-flow sub-pool sub-pool-bw</td>
<td>Sets the reservable bandwidth, the maximum RSVP bandwidth available for a flow and the sub-pool bandwidth on this interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 1000 100 sub-pool 150</td>
<td></td>
</tr>
<tr>
<td>Step 5 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Differentiated Services Traffic Engineering, on page 115
Bandwidth Configuration (MAM): Example, on page 89
Bandwidth Configuration (RDM): Example, on page 90

Configuring MPLS O-UNI Bandwidth

For this application you do not need to configure bandwidth for the data channel or the control channel. There is no other specific RSVP configuration needed for this application.
Enabling Graceful Restart

Perform this task to enable graceful restart for implementations using both node-id and interface-based hellos. RSVP graceful restart provides a control plane mechanism to ensure high availability, which allows detection and recovery from failure conditions while preserving nonstop forwarding services.

SUMMARY STEPS

1. configure
2. rsvp
3. signalling graceful-restart
4. signalling graceful-restart interface-based
5. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 rsvp</td>
<td>Enters the RSVP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# rsvp</td>
<td></td>
</tr>
<tr>
<td>Step 3 signalling graceful-restart</td>
<td>Enables the graceful restart process on the node.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp)# signalling graceful-restart</td>
<td></td>
</tr>
<tr>
<td>Step 4 signalling graceful-restart interface-based</td>
<td>Enables interface-based graceful restart process on the node.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp)# signalling graceful-restart interface-based</td>
<td></td>
</tr>
<tr>
<td>Step 5 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

- Graceful Restart: Standard and Interface-Based, on page 62
- Enable Graceful Restart: Example, on page 91
- Enable Interface-Based Graceful Restart: Example, on page 91
Configuring ACL-based Prefix Filtering

Two procedures are provided to show how RSVP Prefix Filtering is associated:

• Configuring ACLs for Prefix Filtering, on page 73
• Configuring RSVP Packet Dropping, on page 74

Configuring ACLs for Prefix Filtering

Perform this task to configure an extended access list ACL that identifies the source and destination prefixes used for packet filtering.

Note
The extended ACL needs to be configured separately using extended ACL configuration commands.

SUMMARY STEPS

1. configure
2. rsvp
3. signalling prefix-filtering access-list
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>rsvp</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# rsvp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>signalling prefix-filtering access-list</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-rsvp)# signalling prefix-filtering access-list banks</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

Related Topics
ACL-based Prefix Filtering, on page 64
Configuring RSVP Packet Dropping

Perform this task to configure RSVP to drop RA packets when the ACL match returns an implicit (default) deny.

The default behavior performs normal RSVP processing on RA packets when the ACL match returns an implicit (default) deny.

SUMMARY STEPS

1. configure
2. rsvp
3. signalling prefix-filtering default-deny-action
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 rsvp</td>
<td>Enters the RSVP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# rsvp</td>
<td></td>
</tr>
<tr>
<td>Step 3 signalling prefix-filtering default-deny-action</td>
<td>Drops RA messages.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp)# signalling prefix-filtering default-deny-action</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Overview of RSVP for MPLS-TE and MPLS O-UNI, on page 60
Set DSCP for RSVP Packets: Example, on page 92
Verifying RSVP Configuration

This figure illustrates the topology.

Figure 8: Sample Topology

Perform the following steps to verify RSVP configuration.

SUMMARY STEPS

1. `show rsvp session`
2. `show rsvp counters messages summary`
3. `show rsvp counters events`
4. `show rsvp interface type interface-path-id [detail]`
5. `show rsvp graceful-restart`
6. `show rsvp graceful-restart [neighbors ip-address | detail]`
7. `show rsvp interface`
8. `show rsvp neighbor`

DETAILED STEPS

**Step 1**

`show rsvp session`

Verifies that all routers on the path of the LSP are configured with at least one Path State Block (PSB) and one Reservation State Block (RSB) per session.

Example:

```
RP/0/RP0/CPU0:router# show rsvp session
Type Destination Add DPort Proto/ExtTunID PSBs RSBs Reqs
---- --------------- ----- --------------- ----- ----- ----- LSP4
172.16.70.70 6 10.51.51.51 1 1 0
```

In the example, the output represents an LSP from ingress (head) router 10.51.51.51 to egress (tail) router 172.16.70.70. The tunnel ID (also called the destination port) is 6.

Example:

If no states can be found for a session that should be up, verify the application (for example, MPLS-TE and O-UNI) to see if everything is in order. If a session has one PSB but no RSB, this indicates that either the Path message is not making it to the egress (tail) router or the reservation message is not making it back to the router R1 in question.
Go to the downstream router R2 and display the session information:

**Example:**

If R2 has no PSB, either the path message is not making it to the router or the path message is being rejected (for example, due to lack of resources). If R2 has a PSB but no RSB, go to the next downstream router R3 to investigate. If R2 has a PSB and an RSB, this means the reservation is not making it from R2 to R1 or is being rejected.

**Step 2** show rsvp counters messages summary

Verifies whether the RSVP message is being transmitted and received.

**Example:**

```
RP/0/RP0/CPU0:router# show rsvp counters messages summary
All RSVP Interfaces Recv Xmit Recv Xmit Path 0 25 Resv 30 0 PathError 0 0 ResvError 0 1 PathTear 0 30 ResvTear 12 0 ResvConfirm 0 0 Ack 24 37 Bundle 0 Hello 0 5099 SRefresh 8974 9012 OutOfOrder 0 Retransmit 20 Rate Limited 0
```

**Step 3** show rsvp counters events

Verifies how many RSVP states have expired. Because RSVP uses a soft-state mechanism, some failures will lead to RSVP states to expire due to lack of refresh from the neighbor.

**Example:**

```
RP/0/RP0/CPU0:router# show rsvp counters events
mgmtEthernet0/0/0/0 tunnel6 Expired Path states 0 Expired Path states 0 Expired Resv states 0 Expired Resv states 0 NACKs received 0 NACKs received 0 POS0/3/0/0 POS0/3/0/1 Expired Path states 0 Expired Path states 0 Expired Resv states 0 Expired Resv states 1 NACKs received 0 NACKs received 1
```

**Step 4** show rsvp interface type interface-path-id [detail]

Verifies that refresh reduction is working on a particular interface.

**Example:**

```
RP/0/RP0/CPU0:router# show rsvp interface pos0/3/0/3 detail
```

**Step 5** show rsvp graceful-restart
Verifies that graceful restart is enabled locally.

**Example:**

```
RP/0/RP0/CPU0:router# show rsvp graceful-restart
Graceful restart: enabled Number of global neighbors: 1 Local MPLS router id: 10.51.51.51 Restart time: 60 seconds Recovery time: 0 seconds Recovery timer: Not running Hello interval: 5000 milliseconds Maximum Hello miss-count: 3
```

**Step 6**

```
show rsvp graceful-restart neighbors ip-address | detail
```

Verifies that graceful restart is enabled on the neighbor(s). These examples show that neighbor 192.168.60.60 is not responding to hello messages.

**Example:**

```
RP/0/RP0/CPU0:router# show rsvp graceful-restart neighbors 192.168.60.60
Neighbor App State Recovery Reason Since LostCnt ------------------------ ------- ------ -------- ------------
192.168.60.60 MPLS INIT DONE N/A 12/06/2003 0
```

```
RP/0/RP0/CPU0:router# show rsvp graceful-restart neighbors detail
Neighbor: 192.168.60.60 Source: 10.51.51.51 (MPLS) Hello instance for application MPLS Hello State: INIT (for 3d23h)
Number of times communications with neighbor lost: 0 Reason: N/A Recovery State: DONE Number of Interface neighbors: 1 address: 10.64.64.65 Restart time: 0 seconds Recovery time: 0 seconds Restart timer: Not running Recovery timer: Not running Hello interval: 5000 milliseconds Maximum allowed missed Hello messages: 3
```

**Step 7**

```
show rsvp interface
```

Verifies the available RSVP bandwidth.

**Example:**

```
RP/0/RP0/CPU0:router# show rsvp interface
Interface MaxBW MaxFlow Allocated MaxSub ----------------
------ -------- --------------- -------- Et0/0/0/0 0 0 0 ( 0%) 0 PO0/3/0/0 1000M 1000M 0 ( % ) 0 PO0/3/0/1 1000M 1000M 0 ( % ) 0 PO0/3/0/2 1000M 1000M 0 ( % ) 0 PO0/3/0/3 1000M 1000M 1K ( % ) 0
```

**Step 8**

```
show rsvp neighbor
detail
```

Verifies the RSVP neighbors.

**Example:**

```
RP/0/RP0/CPU0:router# show rsvp neighbor detail
Global Neighbor: 40.40.40.40 Interface Neighbor: 1.1.1.1 Interface: POS0/0/0/0 Remote epoch: 0xXXXXXXX Out of order messages: 0 Retransmitted messages: 0 Interface Neighbor: 2.2.2.2 Interface: POS0/1/0/0 Refresh Reduction: "Enabled" or "Disabled". Remote epoch: 0xXXXXXXX Out of order messages: 0 Retransmitted messages: 0
```
Enabling RSVP Traps

With the exception of the RSVP MIB traps, no action is required to activate the MIBs. This MIB feature is automatically enabled when RSVP is turned on; however, RSVP traps must be enabled. Perform this task to enable all RSVP MIB traps, NewFlow traps, and LostFlow traps.

SUMMARY STEPS

1. configure
2. snmp-server traps rsvp lost-flow
3. snmp-server traps rsvp new-flow
4. snmp-server traps rsvp all
5. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>snmp-server traps rsvp lost-flow</td>
<td>Sends RSVP notifications to enable RSVP LostFlow traps.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# snmp-server traps rsvp lost-flow</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>snmp-server traps rsvp new-flow</td>
<td>Sends RSVP notifications to enable RSVP NewFlow traps.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# snmp-server traps rsvp new-flow</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>snmp-server traps rsvp all</td>
<td>Sends RSVP notifications to enable all RSVP MIB traps.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# snmp-server traps rsvp all</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
How to Implement RSVP Authentication

There are three types of RSVP authentication modes—global, interface, and neighbor. These topics describe how to implement RSVP authentication for each mode:

Configuring Global Configuration Mode RSVP Authentication

These tasks describe how to configure RSVP authentication in global configuration mode:

Enabling RSVP Authentication Using the Keychain in Global Configuration Mode

Perform this task to enable RSVP authentication for cryptographic authentication by specifying the keychain in global configuration mode.

Note

You must configure a keychain before completing this task (see Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router).

### SUMMARY STEPS

1. configure
2. rsvp authentication
3. key-source key-chain key-chain-name
4. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>rsvp authentication</td>
<td>Enters RSVP authentication configuration mode.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router(config)# rsvp authentication
RP/0/RP0/CPU0:router(config-rsvp-auth)#
```
### Configuring Global Configuration Mode RSVP Authentication

#### COMMANDS AND ACTIONS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3</td>
<td>key-source key-chain key-chain-name</td>
<td>Specifies the source of the key information to authenticate RSVP signaling messages.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-rsvp-auth)#</td>
<td>key-source key-chain mpls-keys</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-rsvp-auth)#</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

#### RELATED TOPICS

- Key-source Key-chain, on page 69
- RSVP Authentication Global Configuration Mode: Example, on page 93

### Configuring a Lifetime for RSVP Authentication in Global Configuration Mode

Perform this task to configure a lifetime value for RSVP authentication in global configuration mode.

#### SUMMARY STEPS

1. configure
2. rsvp authentication
3. life-time seconds
4. commit

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>rsvp authentication</td>
<td>Enters RSVP authentication configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)#</td>
<td>rsvp authentication</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-rsvp-auth)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>life-time seconds</td>
<td>Controls how long RSVP maintains security associations with other trusted RSVP neighbors.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-rsvp-auth)#</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Global Configuration Mode RSVP Authentication

**Related Topics**
- Global, Interface, and Neighbor Authentication Modes, on page 66
- RSVP Authentication Global Configuration Mode: Example, on page 93

**Configuring the Window Size for RSVP Authentication in Global Configuration Mode**

Perform this task to configure the window size for RSVP authentication in global configuration mode.

**SUMMARY STEPS**

1. configure
2. rsvp authentication
3. window-size \( N \)
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>rsvp authentication</td>
</tr>
</tbody>
</table>
| **Example:** | \`
RP/0/RP0/CPU0:router(config)# rsvp authentication
RP/0/RP0/CPU0:router(config-rsvp-auth)#
\` |
| **Step 3** | window-size \( N \) |
| **Example:** | \`
RP/0/RP0/CPU0:router(config-rsvp-auth)#
\` |

Specifies the maximum number of RSVP authenticated messages that can be received out-of-sequence.

\( N \)

Size of the window to restrict out-of-sequence messages. The range is from 1 to 64. The default value is 1, in which case all out-of-sequence messages are dropped.

### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>life-time 2000</strong></td>
<td>seconds</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Length of time (in seconds) that RSVP maintains idle security associations with other trusted RSVP neighbors. Range is from 30 to 86400. The default value is 1800.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>
### Configuring an Interface for RSVP Authentication

These tasks describe how to configure an interface for RSVP authentication:

#### Specifying the RSVP Authentication Keychain in Interface Mode

Perform this task to specify RSVP authentication keychain in interface mode.

You must configure a keychain first (see *Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router*).

### SUMMARY STEPS

1. `configure`
2. `rsvp interface type interface-path-id`
3. `authentication`
4. `key-source key-chain key-chain-name`
5. `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>rsvp interface type interface-path-id</code></td>
<td>Enters RSVP interface configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0:router(config)# rsvp interface POS 0/2/1/0
RP/0/RP0/CPU0:router(config-rsvp-if)#
```
### Command or Action | Purpose |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters RSVP authentication configuration mode.</td>
</tr>
<tr>
<td><strong>authentication</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-rsvp-if)# authentication RP/0/RP0/CPU0:router(config-rsvp-if-auth)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the source of the key information to authenticate RSVP signaling messages.</td>
</tr>
<tr>
<td><strong>key-source key-chain</strong></td>
<td></td>
</tr>
<tr>
<td><strong>key-chain-name</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config-rsvp-if-auth)# key-source key-chain mpls-keys</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><strong>commit</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- Global, Interface, and Neighbor Authentication Modes, on page 66
- RSVP Authentication by Using All the Modes: Example, on page 95

### Configuring a Lifetime for an Interface for RSVP Authentication

Perform this task to configure a lifetime for the security association for an interface.

**SUMMARY STEPS**

1. configure
2. rsvp interface type interface-path-id
3. authentication
4. life-time seconds
5. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><strong>configure</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters RSVP interface configuration mode.</td>
</tr>
<tr>
<td><strong>rsvp interface type interface-path-id</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# rsvp interface POS 0/2/1/0</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring an Interface for RSVP Authentication

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3 authentication</td>
<td>Enters RSVP authentication configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-rsvp-if)# authentication&lt;br&gt;RP/0/RP0/CPU0:router(config-rsvp-if-auth)#</td>
<td></td>
</tr>
<tr>
<td>Step 4 life-time <code>seconds</code></td>
<td>Controls how long RSVP maintains security associations with other trusted RSVP neighbors.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RP0/CPU0:router(config-rsvp-if-auth)# life-time 2000</td>
<td><code>seconds</code> Length of time (in seconds) that RSVP maintains idle security associations with other trusted RSVP neighbors. Range is from 30 to 86400. The default value is 1800.</td>
</tr>
<tr>
<td>Step 5 commit</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- RSVP Authentication Design, on page 65
- RSVP Authentication by Using All the Modes: Example, on page 95

### Configuring the Window Size for an Interface for RSVP Authentication

Perform this task to configure the window size for an interface for RSVP authentication to check the validity of the sequence number received.

**SUMMARY STEPS**

1. configure
2. rsvp interface type interface-path-d
3. authentication
4. window-size `N`
5. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>rsvp interface type interface-path-d</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config)# rsvp interface</strong></td>
</tr>
<tr>
<td></td>
<td><strong>POS 0/2/1/0</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RP/0/RP0/CPU0:router(config-rsvp-if)#</strong></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>authentication</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config-rsvp-if)# authentication</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RP/0/RP0/CPU0:router(config-rsvp-if-auth)#</strong></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>window-size N</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>RP/0/RP0/CPU0:router(config-rsvp-if-auth)# window-size 33</strong></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>commit</strong></td>
</tr>
</tbody>
</table>

**Related Topics**

- Guidelines for Window-Size and Out-of-Sequence Messages, on page 69
- RSVP Authentication by Using All the Modes: Example, on page 95
- RSVP Authentication for an Interface: Example, on page 94

**Configuring RSVP Neighbor Authentication**

These tasks describe how to configure the RSVP neighbor authentication:

- Specifying the Keychain for RSVP Neighbor Authentication, on page 85
- Configuring a Lifetime for RSVP Neighbor Authentication, on page 86
- Configuring the Window Size for RSVP Neighbor Authentication, on page 87

**Specifying the Keychain for RSVP Neighbor Authentication**

Perform this task to specify the keychain RSVP neighbor authentication.
You must configure a keychain first (see Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router).

**SUMMARY STEPS**

1. configure
2. rsvp neighbor IP-address authentication
3. key-source key-chain key-chain-name
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>rsvp neighbor IP-address authentication</td>
<td>Enters neighbor authentication configuration mode. Use the rsvp neighbor command to activate RSVP cryptographic authentication for a neighbor.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# rsvp neighbor 1.1.1.1 authentication</td>
<td>IP address of the neighbor. A single IP address for a specific neighbor; usually one of the neighbor's physical or logical (loopback) interfaces.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>authentication</td>
<td>Configures the RSVP authentication parameters.</td>
</tr>
<tr>
<td>Step 3</td>
<td>key-source key-chain key-chain-name</td>
<td>Specifies the source of the key information to authenticate RSVP signaling messages.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)# key-source key-chain mpls-keys</td>
<td>key-chain-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Name of the keychain. The maximum number of characters is 32.</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Key-source Key-chain, on page 69
- Security Association, on page 67
- RSVP Neighbor Authentication: Example, on page 94

**Configuring a Lifetime for RSVP Neighbor Authentication**

Perform this task to configure a lifetime for security association for RSVP neighbor authentication mode.
SUMMARY STEPS

1. configure
2. rsvp neighbor IP-address authentication
3. life-time seconds
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters RSVP neighbor authentication configuration mode. Use the <code>rsvp neighbor</code> command to specify a neighbor under RSVP.</td>
</tr>
</tbody>
</table>

**Step 2** rsvp neighbor IP-address authentication

Example:

```
RP/0/RP0/CPU0:router(config)# rsvp neighbor 1.1.1.1 authentication
RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)#
```

Purpose:
- **IP address**
  - IP address of the neighbor. A single IP address for a specific neighbor; usually one of the neighbor's physical or logical (loopback) interfaces.
- **authentication**
  - Configures the RSVP authentication parameters.

**Step 3** life-time seconds

Example:

```
RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)# life-time 2000
```

Purpose:
- **seconds**
  - Length of time (in seconds) that RSVP maintains idle security associations with other trusted RSVP neighbors. The argument specifies the length of time.
  - Range is from 30 to 86400. The default value is 1800.

**Step 4** commit

Related Topics

- Security Association, on page 67
- RSVP Authentication Global Configuration Mode: Example, on page 93

Configuring the Window Size for RSVP Neighbor Authentication

Perform this task to configure the RSVP neighbor authentication window size to check the validity of the sequence number received.
### SUMMARY STEPS

1. configure
2. `rsvp neighbor IP address authentication`
3. `window-size N`
4. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters RSVP neighbor authentication configuration mode. Use the <code>rsvp neighbor</code> command to specify a neighbor under RSVP.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>rsvp neighbor IP address authentication</code></td>
<td><strong>IP address</strong> IP address of the neighbor. A single IP address for a specific neighbor; usually one of the neighbor's physical or logical (loopback) interfaces. <strong>authentication</strong> Configures the RSVP authentication parameters.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>window-size N</code></td>
<td>Specifies the maximum number of RSVP authenticated messages that is received out-of-sequence. <strong>N</strong> Size of the window to restrict out-of-sequence messages. The range is from 1 to 64. The default value is 1, in which case all out-of-sequence messages are dropped.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- Guidelines for Window-Size and Out-of-Sequence Messages, on page 69
- RSVP Authentication by Using All the Modes: Example, on page 95
- RSVP Authentication for an Interface: Example, on page 94

### Verifying the Details of the RSVP Authentication

To display the security associations that RSVP has established with other RSVP neighbors, use the `show rsvp authentication` command.
Eliminating Security Associations for RSVP Authentication

To eliminate RSVP authentication SA’s, use the `clear rsvp authentication` command. To eliminate RSVP counters for each SA, use the `clear rsvp counters authentication` command.

Configuration Examples for RSVP

Sample RSVP configurations are provided for some of the supported RSVP features.

- **Bandwidth Configuration (Prestandard):** Example, on page 89
- **Bandwidth Configuration (MAM):** Example, on page 89
- **Bandwidth Configuration (RDM):** Example, on page 90
- **Refresh Reduction and Reliable Messaging Configuration:** Examples, on page 90
- **Configure Graceful Restart:** Examples, on page 91
- **Configure ACL-based Prefix Filtering:** Example, on page 92
- **Set DSCP for RSVP Packets:** Example, on page 92
- **Enable RSVP Traps:** Example, on page 93

Bandwidth Configuration (Prestandard): Example

The example shows the configuration of bandwidth on an interface using prestandard DS-TE mode. The example configures an interface for a reservable bandwidth of 7500, specifies the maximum bandwidth for one flow to be 1000 and adds a sub-pool bandwidth of 2000.

```
rsvp interface pos 0/3/0/0
bandwidth 7500 1000 sub-pool 2000
```

Bandwidth Configuration (MAM): Example

The example shows the configuration of bandwidth on an interface using MAM. The example shows how to limit the total of all RSVP reservations on POS interface 0/3/0/0 to 7500 kbps, and allows each single flow to reserve no more than 1000 kbps.

```
rsvp interface pos 0/3/0/0
bandwidth mam 7500 1000
```

Related Topics

- **Confirming DiffServ-TE Bandwidth,** on page 70
- **Differentiated Services Traffic Engineering,** on page 115
Bandwidth Configuration (RDM): Example

The example shows the configuration of bandwidth on an interface using RDM. The example shows how to limit the total of all RSVP reservations on POS interface 0/3/0/0 to 7500 kbps, and allows each single flow to reserve no more than 1000 kbps.

```
rsvp interface pos 0/3/0/0
bandwidth rdm 7500 1000
```

Related Topics

- Confirming DiffServ-TE Bandwidth, on page 70
- Differentiated Services Traffic Engineering, on page 115

Refresh Reduction and Reliable Messaging Configuration: Examples

Refresh reduction feature as defined by RFC 2961 is supported and enabled by default. The examples illustrate the configuration for the refresh reduction feature. Refresh reduction is used with a neighbor only if the neighbor supports it also.

Refresh Interval and the Number of Refresh Messages Configuration: Example

The example shows how to configure the refresh interval to 30 seconds on POS 0/3/0/0 and how to change the number of refresh messages the node can miss before cleaning up the state from the default value of 4 to 6.

```
rsvp interface pos 0/3/0/0
signalling refresh interval 30
signalling refresh missed 6
```

Retransmit Time Used in Reliable Messaging Configuration: Example

The example shows how to set the retransmit timer to 2 seconds. To prevent unnecessary retransmits, the retransmit time value configured on the interface must be greater than the ACK hold time on its peer.

```
rsvp interface pos 0/4/0/1
signalling refresh reduction reliable retransmit-time 2000
```

Acknowledgement Times Configuration: Example

The example shows how to change the acknowledge hold time from the default value of 400 ms, to delay or speed up sending of ACKs, and the maximum acknowledgment message size from default size of 4096 bytes. The example shows how to change the acknowledge hold time from the default value of 400 ms and how to delay or speed up sending of ACKs. The maximum acknowledgment message default size is from 4096 bytes.

```
rsvp interface pos 0/4/0/1
signalling refresh reduction reliable ack-hold-time 1000
rsvp interface pos 0/4/0/1
signalling refresh reduction reliable ack-max-size 1000
```
Ensure retransmit time on the peers’ interface is at least twice the amount of the ACK hold time to prevent unnecessary retransmissions.

**Summary Refresh Message Size Configuration: Example**

The example shows how to set the summary refresh message maximum size to 1500 bytes.

```bash
rsvp interface pos 0/4/0/1
signalling refresh reduction summary max-size 1500
```

**Disable Refresh Reduction: Example**

If the peer node does not support refresh reduction, or for any other reason you want to disable refresh reduction on an interface, the example shows how to disable refresh reduction on that interface.

```bash
rsvp interface pos 0/4/0/1
signalling refresh reduction disable
```

**Configure Graceful Restart: Examples**

RSVP graceful restart is configured globally or per interface (as are refresh-related parameters). These examples show how to enable graceful restart, set the restart time, and change the hello message interval.

**Enable Graceful Restart: Example**

The example shows how to enable the RSVP graceful restart by default. If disabled, enable it with the following command.

```bash
rsvp signalling graceful-restart
```

**Related Topics**

- **Enabling Graceful Restart, on page 72**
- **Graceful Restart: Standard and Interface-Based, on page 62**

**Enable Interface-Based Graceful Restart: Example**

The example shows how to enable the RSVP graceful restart feature on an interface.

```bash
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config-rsvp)#interface bundle-ether 17
RP/0/RP0/CPU0:router(config-rsvp-if)#signalling hello graceful-restart ?
interface-based Configure Interface-based Hello
RP/0/RP0/CPU0:router(config-rsvp-if)#signalling hello graceful-restart interface-based
RP/0/RP0/CPU0:router(config-rsvp-if)#
```

**Related Topics**

- **Enabling Graceful Restart, on page 72**
Graceful Restart: Standard and Interface-Based, on page 62

Change the Restart-Time: Example

The example shows how to change the restart time that is advertised in hello messages sent to neighbor nodes.

```
rsvp signalling graceful-restart restart-time 200
```

Change the Hello Interval: Example

The example shows how to change the interval at which RSVP graceful restart hello messages are sent per neighbor, and change the number of hellos missed before the neighbor is declared down.

```
rsvp signalling hello graceful-restart refresh interval 4000
rsvp signalling hello graceful-restart refresh misses 4
```

Configure ACL-based Prefix Filtering: Example

The example shows when RSVP receives a Router Alert (RA) packet from source address 1.1.1.1 and 1.1.1.1 is not a local address. The packet is forwarded with IP TTL decremented. Packets destined to 2.2.2.2 are dropped. All other RA packets are processed as normal RSVP packets.

```
show run ipv4 access-list
ipv4 access-list rsvpacl
10 permit ip host 1.1.1.1 any
20 deny ip any host 2.2.2.2
!
show run rsvp
rsvp
signalling prefix-filtering access-list rsvpacl
!
```

Related Topics

- Configuring ACLs for Prefix Filtering, on page 73
- ACL-based Prefix Filtering, on page 64

Set DSCP for RSVP Packets: Example

The configuration example sets the Differentiated Services Code Point (DSCP) field in the IP header of RSVP packets.

```
rsvp interface pos0/2/0/1
signalling dscp 20
```

Related Topics

- Configuring RSVP Packet Dropping, on page 74
- Overview of RSVP for MPLS-TE and MPLS O-UNI, on page 60
Enable RSVP Traps: Example

The example enables the router to send all RSVP traps:

```
configure
  snmp-server traps rsvp all
```

The example enables the router to send RSVP LostFlow traps:

```
configure
  snmp-server traps rsvp lost-flow
```

The example enables the router to send RSVP NewFlow traps:

```
configure
  snmp-server traps rsvp new-flow
```

Related Topics

- Enabling RSVP Traps, on page 78
- RSVP MIB, on page 65

Configuration Examples for RSVP Authentication

These configuration examples are used for RSVP authentication:

- RSVP Authentication Global Configuration Mode: Example, on page 93
- RSVP Authentication for an Interface: Example, on page 94
- RSVP Neighbor Authentication: Example, on page 94
- RSVP Authentication by Using All the Modes: Example, on page 95

RSVP Authentication Global Configuration Mode: Example

The configuration example enables authentication of all RSVP messages and increases the default lifetime of the SAs.

```
rsdp
  authentication
    key-source key-chain default_keys
    life-time 3600
  !
```

Note

The specified keychain (default_keys) must exist and contain valid keys, or signaling will fail.

Related Topics

- Enabling RSVP Authentication Using the Keychain in Global Configuration Mode, on page 79
- Key-source Key-chain, on page 69
RSVP Authentication for an Interface: Example

The configuration example enables authentication of all RSVP messages that are being sent or received on one interface only, and sets the window-size of the SAs.

```plaintext
rsvp
interface GigabitEthernet0/6/0/0
  authentication
  window-size 64
!
!
```

Note: Because the key-source keychain configuration is not specified, the global authentication mode keychain is used and inherited. The global keychain must exist and contain valid keys or signaling fails.

Related Topics

- Configuring the Window Size for RSVP Authentication in Global Configuration Mode, on page 81
- Configuring the Window Size for an Interface for RSVP Authentication, on page 84
- Configuring the Window Size for RSVP Neighbor Authentication, on page 87
- Guidelines for Window-Size and Out-of-Sequence Messages, on page 69

RSVP Neighbor Authentication: Example

The configuration example enables authentication of all RSVP messages that are being sent to and received from only a particular IP address.

```plaintext
rsvp
neighbor 10.0.0.1
  authentication
    key-source key-chain nbr_keys
    !
    !
```

Related Topics

- Specifying the Keychain for RSVP Neighbor Authentication, on page 85
- Key-source Key-chain, on page 69
- Security Association, on page 67
RSVP Authentication by Using All the Modes: Example

The configuration example shows how to perform the following functions:

- Authenticates all RSVP messages.
- Authenticates the RSVP messages to or from 10.0.0.1 by setting the keychain for the **key-source key-chain** command to nbr_keys, SA lifetime is set to 3600, and the default window-size is set to 1.
- Authenticates the RSVP messages not to or from 10.0.0.1 by setting the keychain for the **key-source key-chain** command to default_keys, SA lifetime is set to 3600, and the window-size is set 64 when using GigabitEthernet0/6/0/0; otherwise, the default value of 1 is used.

```plaintext
rsvp
interface GigabitEthernet0/6/0/0
    authentication
        window-size 64
! neighbor 10.0.0.1
    authentication
        key-source key-chain nbr_keys
! authentication
    key-source key-chain default_keys
    life-time 3600
!`

**Note**

If a keychain does not exist or contain valid keys, this is considered a configuration error because signaling fails. However, this can be intended to prevent signaling. For example, when using the above configuration, if the nbr_keys does not contain valid keys, all signaling with 10.0.0.1 fails.

**Related Topics**

- Configuring the Window Size for RSVP Authentication in Global Configuration Mode, on page 81
- Configuring the Window Size for an Interface for RSVP Authentication, on page 84
- Configuring the Window Size for RSVP Neighbor Authentication, on page 87
- Guidelines for Window-Size and Out-of-Sequence Messages, on page 69
- Specifying the RSVP Authentication Keychain in Interface Mode, on page 82
- Global, Interface, and Neighbor Authentication Modes, on page 66
- Configuring a Lifetime for an Interface for RSVP Authentication, on page 83
- RSVP Authentication Design, on page 65

**Additional References**

For additional information related to implementing GMPLS UNI, refer to the following references:
### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMPLS UNI commands</td>
<td>GMPLS UNI Commands module in <em>Cisco IOS XR MPLS Command Reference for the Cisco CRS Router</em></td>
</tr>
<tr>
<td>MPLS Traffic Engineering commands</td>
<td>MPLS Traffic Engineering commands module in <em>Cisco IOS XR MPLS Command Reference for the Cisco CRS Router</em></td>
</tr>
<tr>
<td>RSVP commands</td>
<td>RSVP commands module in <em>Cisco IOS XR MPLS Command Reference for the Cisco CRS Router</em></td>
</tr>
<tr>
<td>Getting started material</td>
<td><em>Cisco IOS XR Getting Started Guide for the Cisco CRS Router</em></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Configuring AAA Services module in <em>Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router</em></td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>OIF UNI 1.0</td>
<td><em>User Network Interface (UNI) 1.0 Signaling Specification</em></td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 3471</td>
<td><em>Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description</em></td>
</tr>
<tr>
<td>RFCs</td>
<td>Title</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>RFC 4872</td>
<td>RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery</td>
</tr>
<tr>
<td>RFC 6205</td>
<td>Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers</td>
</tr>
</tbody>
</table>

**Technical Assistance**

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
Implementing MPLS Forwarding

All Multiprotocol Label Switching (MPLS) features require a core set of MPLS label management and forwarding services; the MPLS Forwarding Infrastructure (MFI) supplies these services.

Feature History for Implementing MPLS-TE

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 3.0</td>
<td>No modification.</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>The MPLS IP Time-to-Live Propagation feature was added.</td>
</tr>
</tbody>
</table>

Prerequisites for Implementing Cisco MPLS Forwarding

These prerequisites are required to implement MPLS Forwarding:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Router that runs Cisco IOS XR software.
- Installed composite mini-image and the MPLS package, or a full composite image.
Restrictions for Implementing Cisco MPLS Forwarding

- Label switching on a Cisco router requires that Cisco Express Forwarding (CEF) be enabled.
- CEF is mandatory for Cisco IOS XR software and it does not need to be enabled explicitly.

Information About Implementing MPLS Forwarding

To implement MPLS Forwarding, you should understand these concepts:

MPLS Forwarding Overview

MPLS combines the performance and capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing. MPLS enables service providers to meet the challenges of growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure. The MPLS architecture is flexible and can be employed in any combination of Layer 2 technologies. MPLS support is offered for all Layer 3 protocols, and scaling is possible well beyond that typically offered in today’s networks.

Based on routing information that is stored in the VRF IP routing table and VRF CEF table, packets are forwarded to their destination using MPLS.

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet it pops the label and uses it to direct the packet to the correct CE router. Label forwarding across the provider backbone, is based on either dynamic label switching or traffic engineered paths. A customer data packet carries two levels of labels when traversing the backbone:

- Top label directs the packet to the correct PE router
- Second label indicates how that PE router should forward the packet to the CE router

Related Topics

Configuring the Size of the Local Label, on page 103

Label Switching Functions

In conventional Layer 3 forwarding mechanisms, as a packet traverses the network, each router extracts all the information relevant to forwarding the packet from the Layer 3 header. This information is then used as an index for a routing table lookup to determine the next hop for the packet.

In the most common case, the only relevant field in the header is the destination address field, but in some cases, other header fields might also be relevant. As a result, the header analysis must be done independently at each router through which the packet passes. In addition, a complicated table lookup must also be done at each router.
In label switching, the analysis of the Layer 3 header is done only once. The Layer 3 header is then mapped into a fixed-length, unstructured value called a *label*.

Many different headers can map to the same label, as long as those headers always result in the same choice of next hop. In effect, a label represents a forwarding equivalence class—that is, a set of packets which, however different they may be, are indistinguishable by the forwarding function.

The initial choice of a label need not be based exclusively on the contents of the Layer 3 packet header; for example, forwarding decisions at subsequent hops can also be based on routing policy.

Once a label is assigned, a short label header is added at the front of the Layer 3 packet. This header is carried across the network as part of the packet. At subsequent hops through each MPLS router in the network, labels are swapped and forwarding decisions are made by means of MPLS forwarding table lookup for the label carried in the packet header. Hence, the packet header does not need to be reevaluated during packet transit through the network. Because the label is of fixed length and unstructured, the MPLS forwarding table lookup process is both straightforward and fast.

### Distribution of Label Bindings

Each label switching router (LSR) in the network makes an independent, local decision as to which label value to use to represent a forwarding equivalence class. This association is known as a label binding.

**Note**
The distribution of label bindings cannot be done statically for the Layer 2 VPN pseudowire.

Each LSR informs its neighbors of the label bindings it has made. This awareness of label bindings by neighboring routers is facilitated by these protocols:

- **Label Distribution Protocol (LDP)**
  - Supports MPLS forwarding along normally routed paths.

- **Resource Reservation Protocol (RSVP)**
  - Supports MPLS traffic engineering.

- **Border Gateway Protocol (BGP)**
  - Supports MPLS virtual private networks (VPNs).

When a labeled packet is sent from LSR A to the neighboring LSR B, the label value carried by the IP packet is the label value that LSR B assigned to represent the forwarding equivalence class of the packet. Thus, the label value changes as the IP packet traverses the network.

### MFI Control-Plane Services

The MFI control-plane provides services to MPLS applications, such as Label Distribution Protocol (LDP) and Traffic Engineering (TE), that include enabling and disabling MPLS on an interface, local label allocation, MPLS rewrite setup (including backup links), management of MPLS label tables, and the interaction with other forwarding paths (IP Version 4 [IPv4] for example) to set up imposition and disposition.
MFI Data-Plane Services

The MFI data-plane provides a software implementation of MPLS forwarding in all of these forms:

- Imposition
- Disposition
- Label swapping

Time-to-Live Propagation in Hierarchical MPLS

Cisco IOS XR software provides the flexibility to enable or disable the time-to-live (TTL) propagation for locally generated packets that are independent of packets forwarded from a customer edge (CE) device.

The IP header contains a field of 8 bits that signifies the time that a packet still has before its life ends and is dropped. When an IP packet is sent, its TTL is usually 255 and is then decremented by 1 at each hop. When the TTL field is decremented down to zero, the datagram is discarded. In such a case, the router that dropped the IP packet for which the TTL reached 0 sends an Internet Control Message Protocol (ICMP) message type 11 and code 0 (time exceeded) to the originator of the IP packet.

Related Topics

Configuring the Time-to-Live Propagation in Hierarchical MPLS, on page 102

MPLS Maximum Transmission Unit

MPLS maximum transmission unit (MTU) indicates that the maximum size of the IP packet can still be sent on a data link, without fragmenting the packet. In addition, data links in MPLS networks have a specific MTU, but for labeled packets. All IPv4 packets have one or more labels. This does imply that the labeled packets are slightly bigger than the IP packets, because for every label, four bytes are added to the packet. So, if $n$ is the number of labels, $n \times 4$ bytes are added to the size of the packet when the packet is labeled. The MPLS MTU parameter pertains to labeled packets.

How to Implement MPLS Forwarding

These topics explain how to configure a router for MPLS forwarding.

Configuring the Time-to-Live Propagation in Hierarchical MPLS

Perform this task to enable or disable the time-to-live (TTL) propagation for locally generated packets that are independent of packets forwarded from a customer edge (CE) device.
SUMMARY STEPS

1. configure
2. mpls ip-ttl-propagate disable [ forwarded | local ]
3. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Stops the propagation of IP TTL to and from the MPLS header. The example shows how to disable IP TTL propagation for forwarded MPLS packets.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls ip-ttl-propagate disable [ forwarded</td>
<td>local ]</td>
</tr>
<tr>
<td></td>
<td>Prevents the <code>traceroute</code> command from showing the hops for the forwarded packets.</td>
</tr>
<tr>
<td></td>
<td>local</td>
</tr>
<tr>
<td></td>
<td>Prevents the <code>traceroute</code> command from showing the hops only for local packets.</td>
</tr>
<tr>
<td><strong>Step 3</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

*Time-to-Live Propagation in Hierarchical MPLS*, on page 102

**Configuring the Size of the Local Label**

Perform this task to configure the dynamic range of local labels that are available on packet interfaces.

SUMMARY STEPS

1. configure
2. mpls label range table `table-id {minimum maximum}`
3. commit
4. show mpls label range
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Configure the size of the local label space. The example shows how to configure the size of the local label space using a minimum of 16200 and a maximum of 120000.</td>
</tr>
<tr>
<td>Step 2 mpls label range table table-id {minimum maximum}</td>
<td>Displays the range of local labels available for use on packet interfaces.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls label range 16200 120000</td>
<td></td>
</tr>
<tr>
<td>Step 3 commit</td>
<td></td>
</tr>
<tr>
<td>Step 4 show mpls label range</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# show mpls label range</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
MPLS Forwarding Overview, on page 100

Additional References
For additional information related to implementing MPLS Forwarding, refer to the following references:

Related Documents

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>
### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 3031</td>
<td><em>Multiprotocol Label Switching Architecture</em></td>
</tr>
<tr>
<td>RFC 3443</td>
<td><em>Time to Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks</em></td>
</tr>
<tr>
<td>RFC 4105</td>
<td><em>Requirements for Inter-Area MPLS Traffic Engineering</em></td>
</tr>
</tbody>
</table>
Multiprotocol Label Switching (MPLS) is a standards-based solution driven by the Internet Engineering Task Force (IETF) that was devised to convert the Internet and IP backbones from best-effort networks into business-class transport mediums.

MPLS, with its label switching capabilities, eliminates the need for an IP route look-up and creates a virtual circuit (VC) switching function, allowing enterprises the same performance on their IP-based network services as with those delivered over traditional networks such as Frame Relay or Asynchronous Transfer Mode (ATM).

MPLS traffic engineering (MPLS-TE) software enables an MPLS backbone to replicate and expand upon the TE capabilities of Layer 2 ATM and Frame Relay networks. MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what now can be achieved only by overlaying a Layer 3 network on a Layer 2 network.

The LMP and GMPLS-NNI features are not supported on PRP hardware.

### Feature History for Implementing MPLS-TE

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 2.0</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 3.3.0</td>
<td>Support was added for Generalized MPLS.</td>
</tr>
<tr>
<td>Release 3.4.0</td>
<td>Support was added for Flexible Name-based Tunnel Constraints, Interarea MPLS-TE, MPLS-TE Forwarding Adjacency, GMPLS Protection and Restoration, and GMPLS Path Protection.</td>
</tr>
<tr>
<td>Release 3.4.1</td>
<td>Support was added for MPLS-TE and fast reroute link bundling.</td>
</tr>
<tr>
<td>Release 3.5.0</td>
<td>Support was added for Unequal Load Balancing, IS-IS IP Fast Reroute Loop-free Alternative routing functionality, and Path Computation Element (PCE).</td>
</tr>
<tr>
<td>Release</td>
<td>Modification</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Release 3.7.0</td>
<td>Support was added for the following features:</td>
</tr>
<tr>
<td></td>
<td>• Ignore Intermediate System-to-Intermediate System (IS-IS) overload bit setting in MPLS-TE.</td>
</tr>
<tr>
<td></td>
<td>• MPLS-TE/Fast Reroute (FRR) over Virtual Local Area Network (VLAN) interfaces.</td>
</tr>
<tr>
<td>Release 3.8.0</td>
<td>Support was added for the following features:</td>
</tr>
<tr>
<td></td>
<td>• MPLS-TE Automatic Bandwidth.</td>
</tr>
<tr>
<td></td>
<td>• SRLG (Shared Risk Link Groups).</td>
</tr>
<tr>
<td></td>
<td>• Policy Based Tunnel Selection (PBTS) IPv6 that includes the Interior Gateway Protocol (IGP) default path.</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>The Point-to-Multipoint TE feature was added.</td>
</tr>
<tr>
<td>Release 4.0.0</td>
<td>Support was added for the following features:</td>
</tr>
<tr>
<td></td>
<td>• AutoTunnel Backup</td>
</tr>
<tr>
<td></td>
<td>• SRLG (Shared Risk Link Groups)</td>
</tr>
<tr>
<td>Release 4.1.0</td>
<td>Support was added for the following features:</td>
</tr>
<tr>
<td></td>
<td>• Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE</td>
</tr>
<tr>
<td>Release 4.1.1</td>
<td>The Auto-Tunnel Mesh feature was added.</td>
</tr>
<tr>
<td>Release 4.2.0</td>
<td>Support was added for the following features:</td>
</tr>
<tr>
<td></td>
<td>• Soft-Preemption</td>
</tr>
<tr>
<td></td>
<td>• Path Option Attributes</td>
</tr>
<tr>
<td>Release 4.2.1</td>
<td>The Auto-Tunnel Attribute-set feature was added for auto-backup tunnels.</td>
</tr>
<tr>
<td>Release 4.2.3</td>
<td>Support was added for the following features:</td>
</tr>
<tr>
<td></td>
<td>• End-to-End TE Path Protection Enhancements — Explicit Path Protection and Co-existence of Path Protection with Fast Reroute</td>
</tr>
<tr>
<td></td>
<td>• P2MP-TE Inter-area Enhancements</td>
</tr>
<tr>
<td>Release 6.1.1</td>
<td>Named Tunnel feature was added.</td>
</tr>
</tbody>
</table>
Prerequisites for Implementing Cisco MPLS Traffic Engineering

These prerequisites are required to implement MPLS TE:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Router that runs Cisco IOS XR software.
- Installed composite mini-image and the MPLS package, or a full composite image.
- IGP activated.
- To configure Point-to-Multipoint (P2MP)-TE, a base set of RSVP and TE configuration parameters on ingress, midpoint, and egress nodes in the MPLS network is required. In addition, Point-to-Point (P2P) parameters are required.
- Enable LDP globally by using the mpls ldp command to allocate local labels even in RSVP (MPLS TE) only core. You do not have to specify any interface if the core is LDP free.

Information About Implementing MPLS Traffic Engineering

To implement MPLS-TE, you should understand these concepts:

Overview of MPLS Traffic Engineering

MPLS-TE software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. MPLS is an integration of Layer 2 and Layer 3 technologies. By making traditional Layer 2 features available to Layer 3, MPLS enables traffic engineering. Thus, you can offer in a one-tier network what now can be achieved only by overlaying a Layer 3 network on a Layer 2 network.

MPLS-TE is essential for service provider and Internet service provider (ISP) backbones. Such backbones must support a high use of transmission capacity, and the networks must be very resilient so that they can withstand link or node failures. MPLS-TE provides an integrated approach to traffic engineering. With MPLS, traffic engineering capabilities are integrated into Layer 3, which optimizes the routing of IP traffic, given the constraints imposed by backbone capacity and topology.
Benefits of MPLS Traffic Engineering

MPLS-TE enables ISPs to route network traffic to offer the best service to their users in terms of throughput and delay. By making the service provider more efficient, traffic engineering reduces the cost of the network.

Currently, some ISPs base their services on an overlay model. In the overlay model, transmission facilities are managed by Layer 2 switching. The routers see only a fully meshed virtual topology, making most destinations appear one hop away. If you use the explicit Layer 2 transit layer, you can precisely control how traffic uses available bandwidth. However, the overlay model has numerous disadvantages. MPLS-TE achieves the TE benefits of the overlay model without running a separate network and without a non-scalable, full mesh of router interconnects.

How MPLS-TE Works

MPLS-TE automatically establishes and maintains label switched paths (LSPs) across the backbone by using RSVP. The path that an LSP uses is determined by the LSP resource requirements and network resources, such as bandwidth. Available resources are flooded by means of extensions to a link-state-based Interior Gateway Protocol (IGP).

MPLS-TE tunnels are calculated at the LSP headend router, based on a fit between the required and available resources (constraint-based routing). The IGP automatically routes the traffic to these LSPs.

Typically, a packet crossing the MPLS-TE backbone travels on a single LSP that connects the ingress point to the egress point. MPLS-TE is built on these mechanisms:

Tunnel interfaces

From a Layer 2 standpoint, an MPLS tunnel interface represents the headend of an LSP. It is configured with a set of resource requirements, such as bandwidth and media requirements, and priority. From a Layer 3 standpoint, an LSP tunnel interface is the headend of a unidirectional virtual link to the tunnel destination.

MPLS-TE path calculation module

This calculation module operates at the LSP headend. The module determines a path to use for an LSP. The path calculation uses a link-state database containing flooded topology and resource information.

RSVP with TE extensions

RSVP operates at each LSP hop and is used to signal and maintain LSPs based on the calculated path.

MPLS-TE link management module

This module operates at each LSP hop, performs link call admission on the RSVP signaling messages, and performs bookkeeping on topology and resource information to be flooded.

Link-state IGP (Intermediate System-to-Intermediate System [IS-IS] or Open Shortest Path First [OSPF]—each with traffic engineering extensions)

These IGPs are used to globally flood topology and resource information from the link management module.
Enhancements to the shortest path first (SPF) calculation used by the link-state IGP (IS-IS or OSPF)

The IGP automatically routes traffic to the appropriate LSP tunnel, based on tunnel destination. Static routes can also be used to direct traffic to LSP tunnels.

Label switching forwarding

This forwarding mechanism provides routers with a Layer 2-like ability to direct traffic across multiple hops of the LSP established by RSVP signaling.

One approach to engineering a backbone is to define a mesh of tunnels from every ingress device to every egress device. The MPLS-TE path calculation and signaling 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 distributed using load sharing among the tunnels.

Related Topics

Building MPLS-TE Topology, on page 150
Creating an MPLS-TE Tunnel, on page 153
Build MPLS-TE Topology and Tunnels: Example, on page 258

MPLS Traffic Engineering

Multiprotocol Label Switching (MPLS) is an Internet Engineering Task Force (IETF)-specified framework that provides efficient designation, routing, forwarding, and switching of traffic flows through the network. TE is the process of adjusting bandwidth allocations to ensure that enough bandwidth is available for high-priority traffic.

In MPLS TE, the upstream router creates a network tunnel for a particular traffic stream and sets the bandwidth available for that tunnel.

Backup AutoTunnels

The MPLS Traffic Engineering AutoTunnel Backup feature enables a router to dynamically build backup tunnels on the interfaces that are configured with MPLS TE tunnels. This feature enables a router to dynamically build backup tunnels when they are needed. This prevents you from having to build MPLS TE tunnels statically.

The MPLS Traffic Engineering (TE)—AutoTunnel Backup feature has these benefits:

• Backup tunnels are built automatically, eliminating the need for users to preconfigure each backup tunnel and then assign the backup tunnel to the protected interface.

• Protection is expanded—FRR does not protect IP traffic that is not using the TE tunnel or Label Distribution Protocol (LDP) labels that are not using the TE tunnel.

This feature protects against these failures:

• P2P Tunnel NHOP protection—Protects against link failure for the associated P2P protected tunnel
• **P2P Tunnel NNHOP protection**—Protects against node failure for the associated P2P protected tunnel

• **P2MP Tunnel NHOP protection**—Protects against link failure for the associated P2MP protected tunnel

**Related Topics**

Enabling an AutoTunnel Backup, on page 160

Removing an AutoTunnel Backup, on page 161

Establishing MPLS Backup AutoTunnels to Protect Fast Reroutable TE LSPs, on page 162

Establishing Next-Hop Tunnels with Link Protection, on page 163

Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 270

**AutoTunnel Attribute-set**

This feature supports auto-tunnels configuration using attribute templates, known as attribute-set. The TE attribute-set template that specifies a set of TE tunnel attributes, is locally configured at the head-end of auto-tunnels. The control plane triggers the automatic provisioning of a corresponding TE tunnel, whose characteristics are specified in the respective attribute-set.

Currently, auto-tunnel backups are created with the default values of all tunnel attributes. To support configurable attributes for auto-tunnel backup, it is required to configure attribute-set and assign it to the backup tunnels. The attribute-set consists of a set of tunnel attributes such as priority, affinity, signaled bandwidth, logging, policy-class, record-route and so on.

The following rules (consistent across all auto-tunnels) apply while configuring the attribute-set:

• If no attribute-set template is defined, the auto-tunnels is created using default attribute values.

• If an attribute-set is defined and the attribute-set template is already configured, the auto-tunnel is created using the attributes specified in the associated attribute-set.

• If an attribute-set is assigned, but it is not defined or configured, auto-tunnel is not created.

• Any number of attribute-sets can be configured with same attribute settings.

• Empty tunnel attribute implies all parameters have default values.

• When specific attribute is not specified in the attribute-set, a default value for that attribute is used.

**Link Protection**

The backup tunnels that bypass only a single link of the LSP path provide link protection. They protect LSPs, if a link along their path fails, by rerouting the LSP traffic to the next hop, thereby bypassing the failed link. These are referred to as NHOP backup tunnels because they terminate at the LSP’s next hop beyond the point of failure.
This figure illustrates link protection.

**Figure 9: Link Protection**

Node Protection

The backup tunnels that bypass next-hop nodes along LSP paths are called NNHOP backup tunnels because they terminate at the node following the next-hop node of the LSPs, thereby bypassing the next-hop node. They protect LSPs by enabling the node upstream of a link or node failure to reroute the LSPs and their traffic around a node failure to the next-hop node. NNHOP backup tunnels also provide protection from link failures because they bypass the failed link and the node.

This figure illustrates node protection.

**Figure 10: Node Protection**

Backup AutoTunnel Assignment

At the head or mid points of a tunnel, the backup assignment finds an appropriate backup to protect a given primary tunnel for FRR protection.

The backup assignment logic is performed differently based on the type of backup configured on the output interface used by the primary tunnel. Configured backup types are:
• Static Backup
• AutoTunnel Backup
• No Backup (In this case no backup assignment is performed and the tunnels is unprotected.)

**Note**

Static backup and Backup AutoTunnel cannot exist together on the same interface or link.

**Note**

Node protection is always preferred over link protection in the Backup AutoTunnel assignment.

In order that the Backup AutoTunnel feature operates successfully, the following configuration must be applied at global configuration level:

```
ipv4 unnumbered mpls traffic-eng Loopback 0
```

**Note**

The Loopback 0 is used as router ID.

**Explicit Paths**

Explicit paths are used to create backup autotunnels as follows:

**For NHOP Backup Autotunnels:**

• NHOP excludes the protected link's local IP address.
• NHOP excludes the protected link’s remote IP address.
• The explicit-path name is _autob_nhop_tunnelxxx, where xxx matches the dynamically created backup tunnel ID.

**For NNHOP Backup Autotunnels:**

• NNHOP excludes the protected link’s local IP address.
• NNHOP excludes the protected link’s remote IP address (link address on next hop).
• NNHOP excludes the NHOP router ID of the protected primary tunnel next hop.
• The explicit-path name is _autob_nnhop_tunnelxxx, where xxx matches the dynamically created backup tunnel ID.

**Periodic Backup Promotion**

The periodic backup promotion attempts to find and assign a better backup for primary tunnels that are already protected.
With AutoTunnel Backup, the only scenario where two backups can protect the same primary tunnel is when both an NHOP and NNHOP AutoTunnel Backups get created. The backup assignment takes place as soon as the NHOP and NNHOP backup tunnels come up. So, there is no need to wait for the periodic promotion.

Although there is no exception for AutoTunnel Backups, periodic backup promotion has no impact on primary tunnels protected by AutoTunnel Backup.

One exception is when a manual promotion is triggered by the user using the `mpls traffic-eng fast-reroute timers promotion` command, where backup assignment or promotion is triggered on all FRR protected primary tunnels—even unprotected ones. This may trigger the immediate creation of some AutoTunnel Backup, if the command is entered within the time window when a required AutoTunnel Backup has not been yet created.

You can configure the periodic promotion timer using the global configuration `mpls traffic-eng fast-reroute timers promotion` sec command. The range is 0 to 604800 seconds.

**Note**
A value of 0 for the periodic promotion timer disables the periodic promotion.

## Protocol-Based CLI
Cisco IOS XR software provides a protocol-based command line interface. The CLI provides commands that can be used with the multiple IGP protocols supported by MPLS-TE.

## Differentiated Services Traffic Engineering
MPLS Differentiated Services (Diff-Serv) Aware Traffic Engineering (DS-TE) is an extension of the regular MPLS-TE feature. Regular traffic engineering does not provide bandwidth guarantees to different traffic classes. A single bandwidth constraint is used in regular TE that is shared by all traffic. To support various classes of service (CoS), users can configure multiple bandwidth constraints. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

MPLS DS-TE provides the ability to configure multiple bandwidth constraints on an MPLS-enabled interface. Available bandwidths from all configured bandwidth constraints are advertised using IGP. TE tunnel is configured with bandwidth value and class-type requirements. Path calculation and admission control take the bandwidth and class-type into consideration. RSVP is used to signal the TE tunnel with bandwidth and class-type requirements.

MPLS DS-TE is deployed with either Russian Doll Model (RDM) or Maximum Allocation Model (MAM) for bandwidth calculations.

Cisco IOS XR software supports two DS-TE modes: Prestandard and IETF.

**Related Topics**
- Confirming DiffServ-TE Bandwidth, on page 70
- Bandwidth Configuration (MAM): Example, on page 89
- Bandwidth Configuration (RDM): Example, on page 90
**Prestandard DS-TE Mode**

Prestandard DS-TE uses the Cisco proprietary mechanisms for RSVP signaling and IGP advertisements. This DS-TE mode does not interoperate with third-party vendor equipment. Note that prestandard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces.

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

TE class map is not used with Prestandard DS-TE mode.

**Related Topics**

Configure a Prestandard DS-TE Tunnel, on page 165
Configure IETF DS-TE Tunnels: Example, on page 259

**IETF DS-TE Mode**

IETF DS-TE mode uses IETF-defined extensions for RSVP and IGP. This mode interoperates with third-party vendor equipment.

IETF mode supports multiple bandwidth constraint models, including RDM and MAM, both with two bandwidth pools. In an IETF DS-TE network, identical bandwidth constraint models must be configured on all nodes.

TE class map is used with IETF DS-TE mode and must be configured the same way on all nodes in the network.

**Bandwidth Constraint Models**

IETF DS-TE mode provides support for the RDM and MAM bandwidth constraints models. Both models support up to two bandwidth pools.

Cisco IOS XR software provides global configuration for the switching between bandwidth constraint models. Both models can be configured on a single interface to preconfigure the bandwidth constraints before swapping to an alternate bandwidth constraint model.

**Note**  
NSF is not guaranteed when you change the bandwidth constraint model or configuration information.

By default, RDM is the default bandwidth constraint model used in both pre-standard and IETF mode.

**Maximum Allocation Bandwidth Constraint Model**

The MAM constraint model has the following characteristics:

- Easy to use and intuitive.
- Isolation across class types.
- Simultaneously achieves isolation, bandwidth efficiency, and protection against QoS degradation.
Russian Doll Bandwidth Constraint Model

The RDM constraint model has these characteristics:

- Allows greater sharing of bandwidth among different class types.
- Ensures bandwidth efficiency simultaneously and protection against QoS degradation of all class types.
- Specifies that it is used in conjunction with preemption to simultaneously achieve isolation across class-types such that each class-type is guaranteed its share of bandwidth, bandwidth efficiency, and protection against QoS degradation of all class types.

Note

We recommend that RDM not be used in DS-TE environments in which the use of preemption is precluded. Although RDM ensures bandwidth efficiency and protection against QoS degradation of class types, it does guarantee isolation across class types.

Related Topics

Configuring an IETF DS-TE Tunnel Using RDM, on page 167

TE Class Mapping

Each of the eight available bandwidth values advertised in the IGP corresponds to a TE class. Because the IGP advertises only eight bandwidth values, there can be a maximum of only eight TE classes supported in an IETF DS-TE network.

TE class mapping must be exactly the same on all routers in a DS-TE domain. It is the responsibility of the operator configure these settings properly as there is no way to automatically check or enforce consistency.

The operator must configure TE tunnel class types and priority levels to form a valid TE class. When the TE class map configuration is changed, tunnels already up are brought down. Tunnels in the down state, can be set up if a valid TE class map is found.

The default TE class and attributes are listed. The default mapping includes four class types.

Table 5: TE Classes and Priority

<table>
<thead>
<tr>
<th>TE Class</th>
<th>Class Type</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Unused</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>Unused</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TE Class</td>
<td>Class Type</td>
<td>Priority</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Unused</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>Unused</td>
<td>—</td>
</tr>
</tbody>
</table>

**Flooding**

Available bandwidth in all configured bandwidth pools is flooded on the network 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 sub-pool available bandwidth and maximum bandwidth to flood the network in these events:

- Periodic timer expires (this does not depend on bandwidth pool type).
- Tunnel origination node has out-of-date information for either available global pool or sub-pool bandwidth, causing tunnel admission failure at the midpoint.
- Consumed bandwidth crosses user-configured thresholds. The same threshold is used for both global pool and sub-pool. If one bandwidth crosses the threshold, both bandwidths are 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 sub-pool), 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 can cause the locked bandwidth allocated to sub-pool tunnels to be reduced (and hence to cross a threshold). A sub-pool TE tunnel setup can similarly cause the locked bandwidth for global pool TE tunnels to cross a threshold. Thus, sub-pool TE and global pool TE tunnels can affect each other when flooding is triggered by thresholds.
Fast Reroute

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

FRR (link or node) is supported over sub-pool 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 are redirected into the protection LSP, regardless of whether they are sub-pool or global pool tunnels.

---

**Note**

The ability to configure FRR on a per-LSP basis makes it possible to provide different levels of fast restoration to tunnels from different bandwidth pools.

You should be aware of these requirements for the backup tunnel path:

- Backup tunnel must not pass through the element it protects.
- Primary tunnel and a backup tunnel should intersect at least at two points (nodes) on the path: point of local repair (PLR) and merge point (MP). PLR is the headend of the backup tunnel, and MP is the tailend of the backup tunnel.

---

**Note**

When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.

---

**Related Topics**

[Protecting MPLS Tunnels with Fast Reroute, on page 157](#)

---

**IS-IS IP Fast Reroute Loop-free Alternative**

For bandwidth protection, there must be sufficient backup bandwidth available to carry primary tunnel traffic. Use the `ipfrr lfa` command to compute loop-free alternates for all links or neighbors in the event of a link or node failure. To enable node protection on broadcast links, IPRR and bidirectional forwarding detection (BFD) must be enabled on the interface under IS-IS.

---

**Note**

MPLS FRR and IPFRR cannot be configured on the same interface at the same time.

---

For information about configuring BFD, see *Cisco IOS XR Interface and Hardware Configuration Guide for the Cisco CRS-1 Router*. 

MPLS-TE and Fast Reroute over Link Bundles

MPLS Traffic Engineering (TE) and Fast Reroute (FRR) are supported over bundle interfaces. MPLS-TE/FRR over virtual local area network (VLAN) interfaces is supported. Bidirectional forwarding detection (BFD) over VLAN is used as an FRR trigger to obtain less than 50 milliseconds of switchover time.

These link bundle types are supported for MPLS-TE/FRR:

- Over POS link bundles.
- Over Ethernet link bundles.
- Over VLANs over Ethernet link bundles.
- Number of links are limited to 100 for MPLS-TE and FRR.
- VLANs go over any Ethernet interface (for example, GigabitEthernet, TenGigE, and FastEthernet, so forth).

FRR is supported over bundle interfaces in the following ways:

- Uses minimum links as a threshold to trigger FRR over a bundle interface.
- Uses the minimum total available bandwidth as a threshold to trigger FRR.

Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE

The Ignore Intermediate System-to-Intermediate System (IS-IS) overload bit avoidance feature allows network administrators to prevent RSVP-TE label switched paths (LSPs) from being disabled, when a router in that path has its Intermediate System-to-Intermediate System (IS-IS) overload bit set. The IS-IS overload bit avoidance feature is activated using this command:

```
mpls traffic-eng path-selection ignore overload
```

The IS-IS overload bit avoidance feature is deactivated using the `no` form of this command:

```
no mpls traffic-eng path-selection ignore overload
```

When the IS-IS overload bit avoidance feature is activated, all nodes, including head nodes, mid nodes, and tail nodes, with the overload bit set, are ignored. This means that they are still available for use with RSVP-TE label switched paths (LSPs). This feature enables you to include an overloaded node in CSPF.

Enhancement Options of IS-IS OLA

You can restrict configuring IS-IS overload bit avoidance with the following enhancement options:

- `path-selection ignore overload head`
  The tunnels stay up if `set-overload-bit` is set by IS-IS on the head router. Ignores overload during CSPF for LSPs originating from an overloaded node. In all other cases (mid, tail, or both), the tunnel stays down.

- `path-selection ignore overload mid`
The tunnels stay up if `set-overload-bit` is set by IS-IS on the mid router. Ignores overload during CSPF for LSPs transiting from an overloaded node. In all other cases (head, tail, or both), the tunnel stays down.

- **path-selection ignore overload tail**
  The tunnels stay up if `set-overload-bit` is set by IS-IS on the tail router. Ignores overload during CSPF for LSPs terminating at an overloaded node. In all other cases (head, mid, or both), the tunnel stays down.

- **path-selection ignore overload**
  The tunnels stay up irrespective of on which router the `set-overload-bit` is set by IS-IS.

  **Note**
  When you do not select any of the options, including head nodes, mid nodes, and tail nodes, you get a behavior that is applicable to all nodes. This behavior is backward compatible in nature.

For more information related to IS-IS overload avoidance related commands, see *Cisco IOS XR MPLS Command Reference for the Cisco CRS Router*.

**Related Topics**
- Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE, on page 173
- Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 260

**DWDM Transponder Integration**

A GMPLS UNI based solution preserves all the advantages of the integration of the DWDM transponder into the router blade. These advantages include:

- improved CAPEX and OPEX models
- component, space and power savings
- improved IP availability through pro-active protection.

**GMPLS Benefits**

GMPLS bridges the IP and photonic layers, thereby making possible interoperable and scalable parallel growth in the IP and photonic dimensions.

This allows for rapid service deployment and operational efficiencies, as well as for increased revenue opportunities. A smooth transition becomes possible from a traditional segregated transport and service overlay model to a more unified peer model.

By streamlining support for multiplexing and switching in a hierarchical fashion, and by utilizing the flexible intelligence of MPLS-TE, optical switching GMPLS becomes very helpful for service providers wanting to manage large volumes of traffic in a cost-efficient manner.
GMPLS Support

GMPLS-TE provides support for:

- Open Shortest Path First (OSPF) for bidirectional TE tunnel
- Frame, lambda, and port (fiber) labels
- Numbered or Unnumbered links
- OSPF extensions—Route computation with optical constraints
- RSVP extensions—Graceful Restart
- Graceful deletion
- LSP hierarchy
- Peer model
- Border model Control plane separation
- Interarea or AS-Verbatim
- BGP4 or MPLS
- Restoration—Dynamic path computation
- Control channel manager
- Link summary
- Protection and restoration

Related Topics

- Configuring Router IDs, on page 174
- Configuring OSPF over IPCC, on page 176

GMPLS Protection and Restoration

GMPLS provides protection against failed channels (or links) between two adjacent nodes (span protection) and end-to-end dedicated protection (path protection). After the route is computed, signaling to establish the backup paths is carried out through RSVP-TE or CR-LDP. For span protection, 1+1 or M:N protection schemes are provided by establishing secondary paths through the network. In addition, you can use signaling messages to switch from the failed primary path to the secondary path.

Note

Only 1:1 end-to-end path protection is supported.

The restoration of a failed path refers to the dynamic establishment of a backup path. This process requires the dynamic allocation of resources and route calculation. The following restoration methods are described:

- Line restoration—Finds an alternate route at an intermediate node.
- Path restoration—Initiates at the source node to route around a failed path within the path for a specific LSP.
Restoration schemes provide more bandwidth usage, because they do not preallocate any resource for an LSP. GMPLS combines MPLS-FRR and other types of protection, such as SONET/SDH and wavelength. In addition to SONET alarms in POS links, protection and restoration is also triggered by bidirectional forwarding detection (BFD).

### 1:1 LSP Protection
When one specific protecting LSP or span protects one specific working LSP or span, 1:1 protection scheme occurs. However, normal traffic is transmitted only over one LSP at a time for working or recovery. 1:1 protection with extra traffic refers to the scheme in which extra traffic is carried over a protecting LSP when the protecting LSP is not being used for the recovery of normal traffic. For example, the protecting LSP is in standby mode. When the protecting LSP is required to recover normal traffic from the failed working LSP, the extra traffic is preempted. Extra traffic is not protected, but it can be restored. Extra traffic is transported using the protected LSP resources.

### Shared Mesh Restoration and M:N Path Protection
Both shared mesh restoration and M:N (1:N is more practical) path protection offers sharing for protection resources for multiple working LSPs. For 1:N protection, a specific protecting LSP is dedicated to the protection of up to N working LSPs and spans. Shared mesh is defined as preplanned LSP rerouting, which reduces the restoration resource requirements by allowing multiple restoration LSPs to be initiated from distinct ingress nodes to share common resources, such as links and nodes.

### End-to-end Recovery
End-to-end recovery refers to an entire LSP from the source for an ingress router endpoint to the destination for an egress router endpoint.

### GMPLS Protection Requirements
The GMPLS protection requirements are specific to the protection scheme that is enabled at the data plane. For example, SONET APS or MPLS-FRR are identified as the data level for GMPLS protection.

### GMPLS Prerequisites
The following prerequisites are required to implement GMPLS on Cisco IOS XR software:
- You must be in a user group associated with a task group that includes the proper task IDs for GMPLS commands.
- Router that runs Cisco IOS XR software.
- Installation of the Cisco IOS XR software mini-image on the router.

### Flexible Name-based Tunnel Constraints
MPLS-TE Flexible Name-based Tunnel Constraints provides a simplified and more flexible means of configuring link attributes and path affinities to compute paths for MPLS-TE tunnels.
In the traditional TE scheme, links are configured with attribute-flags that are flooded with TE link-state parameters using Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF).

MPLS-TE Flexible Name-based Tunnel Constraints lets you assign, or map, up to 32 color names for affinity and attribute-flag attributes instead of 32-bit hexadecimal numbers. After mappings are defined, the attributes can be referred to by the corresponding color name in the command-line interface (CLI). Furthermore, you can define constraints using include, include-strict, exclude, and exclude-all arguments, where each statement can contain up to 10 colors, and define include constraints in both loose and strict sense.

You can configure affinity constraints using attribute flags or the Flexible Name Based Tunnel Constraints scheme; however, when configurations for both schemes exist, only the configuration pertaining to the new scheme is applied.

---

### Note

Related Topics

- Assigning Color Names to Numeric Values, on page 197
- Associating Affinity-Names with TE Links, on page 198
- Associating Affinity Constraints for TE Tunnels, on page 199
- Configure Flexible Name-based Tunnel Constraints: Example, on page 262

---

### MPLS Traffic Engineering Interarea Tunneling

These topics describe the following new extensions of MPLS-TE:

- Interarea Support, on page 124
- Multiarea Support, on page 125
- Loose Hop Expansion, on page 126
- Loose Hop Reoptimization, on page 126
- Fast Reroute Node Protection, on page 126

---

### Interarea Support

The MPLS-TE interarea tunneling feature allows you to establish P2P and P2MP TE tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thereby eliminating the requirement that headend and tailend routers reside in a single area.

Interarea support allows the configuration of a TE LSP that spans multiple areas, where its headend and tailend label switched routers (LSRs) reside in different IGP areas.

Multiarea and Interarea TE are required by the customers running multiple IGP area backbones (primarily for scalability reasons). This lets you limit the amount of flooded information, reduces the SPF duration, and lessens the impact of a link or node failure within an area, particularly with large WAN backbones split in multiple areas.
This figure shows a typical interarea TE network.

*Figure 11: Interarea (OSPF) TE Network Diagram*

### Multiarea Support

Multiarea support allows an area border router (ABR) LSR to support MPLS-TE in more than one IGP area. A TE LSP is still confined to a single area.

Multiarea and Interarea TE are required when you run multiple IGP area backbones. The Multiarea and Interarea TE allows you to:

- Limit the volume of flooded information.
- Reduce the SPF duration.
- Decrease the impact of a link or node failure within an area.

*Figure 12: Interlevel (IS-IS) TE Network*

As shown in the figure, R2, R3, R7, and R4 maintain two databases for routing and TE information. For example, R3 has TE topology information related to R2, flooded through Level-1 IS-IS LSPs plus the TE topology information related to R4, R9, and R7, flooded as Level 2 IS-IS Link State PDUs (LSPs) (plus, its own IS-IS LSP).

**Note**

You can configure multiple areas within an IS-IS Level 1. This is transparent to TE. TE has topology information about the IS-IS level, but not the area ID.
**Loose Hop Expansion**

Loose hop optimization allows the reoptimization of tunnels spanning multiple areas and solves the problem which occurs when an MPLS-TE LSP traverses hops that are not in the LSP’s headend's OSPF area and IS-IS level.

Interarea MPLS-TE allows you to configure an interarea traffic engineering (TE) label switched path (LSP) by specifying a loose source route of ABRs along the path. It is then the responsibility of the ABR (having a complete view of both areas) to find a path obeying the TE LSP constraints within the next area to reach the next hop ABR (as specified on the headend). The same operation is performed by the last ABR connected to the tailend area to reach the tailend LSR.

For P2MP-TE tunnels, ABRs support loose hop ERO expansion to find path to the next ABR until it reaches to the tail-end LSR, without introducing remerge.

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

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

**Loose Hop Reoptimization**

Loose hop reoptimization allows the reoptimization of the tunnels spanning multiple areas and solves the problem which occurs when an MPLS-TE headend does not have visibility into other IGP areas.

Whenever the headend attempts to reoptimize a tunnel, it tries to find a better path to the ABR in the headend area. If a better path is found then the headend initiates the setup of a new LSP. In case a suitable path is not found in the headend area, the headend initiates a querying message. The purpose of this message is to query the ABRs in the areas other than the headend area to check if there exist any better paths in those areas. The purpose of this message is to query the ABRs in the areas other than the headend area, to check if a better path exists. If a better path does not exist, ABR forwards the query to the next router downstream. Alternatively, if better path is found, ABR responds with a special Path Error to the headend to indicate the existence of a better path outside the headend area. Upon receiving the Path Error that indicates the existence of a better path, the headend router initiates the reoptimization.

**ABR Node Protection**

Because one IGP area does not have visibility into another IGP area, it is not possible to assign backup to protect ABR node. To overcome this problem, node ID sub-object is added into the record route object of the primary tunnel so that at a PLR node, backup destination address can be checked against primary tunnel record-route object and assign a backup tunnel.

**Fast Reroute Node Protection**

If a link failure occurs within an area, the upstream router directly connected to the failed link generates an RSVP path error message to the headend. As a response to the message, the headend sends an RSVP path tear message and the corresponding path option is marked as invalid for a specified period and the next path-option (if any) is evaluated.
To retry the ABR immediately, a second path option (identical to the first one) should be configured. Alternatively, the retry period (path-option hold-down, 2 minutes by default) can be tuned to achieve a faster retry.

**Related Topics**
- Protecting MPLS Tunnels with Fast Reroute, on page 157

## MPLS-TE Forwarding Adjacency

The MPLS-TE Forwarding Adjacency feature allows a network administrator to handle traffic engineering, label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network based on the Shortest Path First (SPF) algorithm. A forwarding adjacency can be created between routers regardless of their location in the network.

### MPLS-TE Forwarding Adjacency Benefits

TE tunnel interfaces are advertised in the IGP network just like any other links. Routers can then use these advertisements in their IGPs to compute the SPF even if they are not the head end of any TE tunnels.

**Related Topics**
- Configuring MPLS-TE Forwarding Adjacency, on page 203
- Configure Forwarding Adjacency: Example, on page 264

### MPLS-TE Forwarding Adjacency Restrictions

The MPLS-TE Forwarding Adjacency feature has these restrictions:

- Using the MPLS-TE Forwarding Adjacency increases the size of the IGP database by advertising a TE tunnel as a link.
- The MPLS-TE Forwarding Adjacency is supported by Intermediate System-to-Intermediate System (IS-IS).
- When the MPLS-TE Forwarding Adjacency is enabled on a TE tunnel, the link is advertised in the IGP network as a Type-Length-Value (TLV) 22 without any TE sub-TLV.
- MPLS-TE forwarding adjacency tunnels must be configured bidirectionally.
- Multicast intact is not supported with MPLS-TE Forwarding Adjacency.

### MPLS-TE Forwarding Adjacency Prerequisites

Your network must support the following features before enabling the MPLS-TE Forwarding Adjacency feature:

- MPLS
- IP Cisco Express Forwarding
- Intermediate System-to-Intermediate System (IS-IS)
Unequal Load Balancing

Unequal load balancing permits the routing of unequal proportions of traffic through tunnels to a common destination. Load shares on tunnels to the same destination are determined by TE from the tunnel configuration and passed through the MPLS Label Switching Database (LSD) to the Forwarding Information Base (FIB).

Load share values are renormalized by the FIB using values suitable for use by the forwarding code. The exact traffic ratios observed may not, therefore, exactly mirror the configured traffic ratios. This effect is more pronounced if there are many parallel tunnels to a destination, or if the load shares assigned to those tunnels are very different. The exact renormalization algorithm used is platform-dependent.

There are two ways to configure load balancing:

**Explicit configuration**

Using this method, load shares are explicitly configured on each tunnel.

**Bandwidth configuration**

If a tunnel is not configured with load-sharing parameters, the tunnel bandwidth and load-share values are considered equivalent for load-share calculations between tunnels, and a direct comparison between bandwidth and load-share configuration values is calculated.

Load shares are not dependent on any configuration other than the load share and bandwidth configured on the tunnel and the state of the global configuration switch.

**Related Topics**

- Setting Unequal Load Balancing Parameters, on page 204
- Enabling Unequal Load Balancing, on page 205
- Configure Unequal Load Balancing: Example, on page 265

Path Computation Element

Path Computation Element (PCE) solves the specific issue of inter-domain path computation for MPLS-TE label switched path (LSPs), when the head-end router does not possess full network topology information (for example, when the head-end and tail-end routers of an LSP reside in different IGP areas).

PCE uses area border routers (ABRs) to compute a TE LSP spanning multiple IGP areas as well as computation of Inter-AS TE LSP.

PCE is usually used to define an overall architecture, which is made of several components, as follows:

**Path Computation Element (PCE)**

Represents a software module (which can be a component or application) that enables the router to compute paths applying a set of constraints between any pair of nodes within the router’s TE topology database. PCEs are discovered through IGP.
Path Computation Client (PCC)

Represents a software module running on a router that is capable of sending and receiving path computation requests and responses to and from PCEs. The PCC is typically an LSR (Label Switching Router).

PCC-PCE communication protocol (PCEP)

Specifies that PCEP is a TCP-based protocol defined by the IETF PCE WG, and defines a set of messages and objects used to manage PCEP sessions and to request and send paths for multi-domain TE LSPs. PCEP is used for communication between PCC and PCE (as well as between two PCEs) and employs IGP extensions to dynamically discover PCE.

This figure shows a typical PCE implementation.

*Figure 13: Path Computation Element Network Diagram*

Path computation elements provides support for the following message types and objects:

- Message types: Open, PCReq, PCRep, PCErr, Close
- Objects: OPEN, CLOSE, RP, END-POINT, LSPA, BANDWIDTH, METRIC, and NO-PATH

**Related Topics**

- Configuring a Path Computation Client, on page 206
- Configuring a Path Computation Element Address, on page 207
- Configuring PCE Parameters, on page 208
- Configure PCE: Example, on page 266

**Policy-Based Tunnel Selection**

These topics provide information about policy-based tunnel selection (PBTS):
Policy-Based Tunnel Selection

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

PBTS works by selecting tunnels based on the classification criteria of the incoming packets, which are based on the IP precedence, experimental (EXP), or type of service (ToS) field in the packet. When there are no paths with a default class configured, this traffic is forwarded using the paths with the lowest class value. PBTS supports up to seven (exp 1 - 7) EXP values associated with a single TE tunnel.

This figure illustrates a PBTS implementation.

**Figure 14: Policy-Based Tunnel Selection Implementation**

Related Topics

- Configuring Policy-based Tunnel Selection, on page 211
- Configure Policy-based Tunnel Selection: Example, on page 267

Policy-Based Tunnel Selection Functions

The following PBTS functions are supported:

- IPv4 traffic arrives unlabeled on the VRF interface and the non-VRF interface.
- MPLS traffic is supported on the VRF interface and the non-VRF interface.
- Load balancing across multiple TE tunnels with the same traffic class attribute is supported.
- Selected TE tunnels are used to service the lowest tunnel class as default tunnels.
- LDP over TE tunnel and single-hop TE tunnel are supported.
- Both Interior Gateway Protocol (IGP) and Label Distribution Protocol (LDP) paths are used as the default path for all traffic that belongs to a class that is not configured on the TE tunnels.
- According to the quality-of-service (QoS) policy, tunnel selection is based on the outgoing experimental (EXP) value and the remarked EXP value.
- L2VPN preferred path selection lets traffic be directed to a particular TE tunnel.
IPv6 traffic for both 6VPE and 6PE scenarios are supported.

Related Topics

Configuring Policy-based Tunnel Selection, on page 211
Configure Policy-based Tunnel Selection: Example, on page 267

PBTS Restrictions

When implementing PBTS, the following restrictions are listed:

- When QoS EXP remarking on an interface is enabled, the EXP value is used to determine the egress tunnel interface, not the incoming EXP value.
- Egress-side remarking does not affect PBTS tunnel selection.
- When no default tunnel is available for forwarding, traffic is dropped.

Path Protection

Path protection provides an end-to-end failure recovery mechanism (that is, a full path protection) for MPLS-TE tunnels. A secondary Label Switched Path (LSP) is established, in advance, to provide failure protection for the protected LSP that is carrying a tunnel's TE traffic. When there is a failure on the protected LSP, the source router immediately enables the secondary LSP to temporarily carry the tunnel's traffic. If there is a failure on the secondary LSP, the tunnel no longer has path protection until the failure along the secondary path is cleared. Path protection can be used within a single area (OSPF or IS-IS), external BGP [eBGP], and static routes.

The failure detection mechanisms triggers a switchover to a secondary tunnel by:

- Path error or resv-tear from Resource Reservation Protocol (RSVP) signaling
- Notification from the Bidirectional Forwarding Detection (BFD) protocol that a neighbor is lost
- Notification from the Interior Gateway Protocol (IGP) that the adjacency is down
- Local teardown of the protected tunnel's LSP due to preemption in order to signal higher priority LSPs, a Packet over SONET (POS) alarm, online insertion and removal (OIR), and so on

An alternate recovery mechanism is Fast Reroute (FRR), which protects MPLS-TE LSPs only from link and node failures, by locally repairing the LSPs at the point of failure. Co-existence of FRR and path protection is supported; this means FRR and path-protection can be configured on the same tunnel at the same time.

Although not as fast as link or node protection, presignaling a secondary LSP is faster than configuring a secondary primary path option, or allowing the tunnel's source router to dynamically recalculate a path. The actual recovery time is topology-dependent, and affected by delay factors such as propagation delay or switch fabric latency.

Related Topics

Configure Tunnels for Path Protection: Example, on page 267
Pre-requisites for Path Protection

These are the pre-requisites for enabling path protection:

- Ensure that your network supports MPLS-TE, Cisco Express Forwarding, and Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).
- Enable MPLS.
- Configure TE on the routers.
- Configure a TE tunnel with a dynamic path option by using the `path-option` command with the `dynamic` keyword.

Related Topics

Configure Tunnels for Path Protection: Example, on page 267

Restrictions for Path Protection

- Only Point-to-Point (P2P) tunnels are supported.
- Point-to-Multipoint (P2MP) TE tunnels are not supported.
- A maximum of one standby LSP is supported.
- There can be only one secondary path for each dynamic path option.
- Explicit path option can be configured for the path protected TE with the secondary path option as dynamic.
- A maximum number of path protected tunnel TE heads is 2000.
- A maximum number of TE tunnel heads is equal to 4000.
- When path protection is enabled for a tunnel, and the primary label switched path (LSP) is not assigned a backup tunnel, but the standby LSP is assigned fast-reroute (FRR), the MPLS TE FRR protected value displayed is different from the Cisco express forwarding (CEF) fast-reroute value.
- Inter-area is not supported for path protection.

Related Topics

Configure Tunnels for Path Protection: Example, on page 267

Restrictions for Explicit Path Protection

Explicit paths are used to create backup autotunnels. Explicit path protection provides a recovery mechanism to protect explicit paths for MPLS-TE tunnels. These restrictions are listed to protect an explicit path:

- Only one explicit protecting path is supported per path-option.
- Link or node path diversity is not ensured for explicit protecting paths.
- An explicit protecting path cannot protect a dynamic path option.
- All options such as `verbatim`, `lockdown` are supported for the protecting path as long as it's explicit.
- An explicit path cannot be protected by its own path option level.
- An explicit path can be protected by a path option level that references the same explicit path name or identifier, because it is considered another path-option.
- Enhanced path protection is not supported.

Related Topics
Configure Tunnels for Path Protection: Example, on page 267

Co-existence of Path Protection with Fast Reroute

Path protection and FRR can be configured on the same tunnel at the same time. The co-existence of path protection and FRR on the same tunnel provides these benefits:

- Protection is expanded — having an FRR protected tunnel that is also path-protected ensures that failures of non-protected links on the primary path are handled more efficiently by a quick switch-over to the pre-signaled standby LSP.
- Quick and effective re-optimization — having a pre-computed standby LSP allows the system to minimize re-optimization LSP path calculation and signaling, by simply switching over to the pre-signaled standby LSP. Effectively, path protection switch over replaces the post-FRR LSP down event re-optimization.
- Total time on backup is reduced — handling FRR failure using a path protection switch over reduces total time on backup because the traffic is diverted from the backup to the standby, as soon as the head-end receives the FRR LSP down notification, without having to wait for a re-optimization LSP.

MPLS-TE Automatic Bandwidth

The MPLS-TE automatic bandwidth feature measures the traffic in a tunnel and periodically adjusts the signaled bandwidth for the tunnel.

These topics provide information about MPLS-TE automatic bandwidth:

MPLS-TE Automatic Bandwidth Overview

MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every head-end. MPLS-TE monitors the traffic rate on a tunnel interface. Periodically, MPLS-TE resizes the bandwidth on the tunnel interface to align it closely with the traffic in the tunnel. MPLS-TE automatic bandwidth can perform these functions:

- Monitors periodic polling of the tunnel output rate
- Resizes the tunnel bandwidth by adjusting the highest rate observed during a given period

For every traffic-engineered tunnel that is configured for an automatic bandwidth, the average output rate is sampled, based on various configurable parameters. Then, the tunnel bandwidth is readjusted automatically based upon either the largest average output rate that was noticed during a certain interval, or a configured maximum bandwidth value.

This table lists the automatic bandwidth functions.
### Table 6: Automatic Bandwidth Variables

<table>
<thead>
<tr>
<th>Function</th>
<th>Command</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application frequency</td>
<td><strong>application</strong> command</td>
<td>Configures how often the tunnel bandwidths changed for each tunnel. The application period is the period of A minutes between the bandwidth applications during which the output rate collection is done.</td>
<td>24 hours</td>
</tr>
<tr>
<td>Requested bandwidth</td>
<td><strong>bw-limit</strong> command</td>
<td>Limits the range of bandwidth within the automatic-bandwidth feature that can request a bandwidth.</td>
<td>0 Kbps</td>
</tr>
<tr>
<td>Collection frequency</td>
<td><strong>auto-bw collect</strong> command</td>
<td>Configures how often the tunnel output rate is polled globally for all tunnels.</td>
<td>5 min</td>
</tr>
<tr>
<td>Highest collected bandwidth</td>
<td>—</td>
<td>You cannot configure this value.</td>
<td>—</td>
</tr>
<tr>
<td>Delta</td>
<td>—</td>
<td>You cannot configure this value.</td>
<td>—</td>
</tr>
</tbody>
</table>

The output rate on a tunnel is collected at regular intervals that are configured by using the `application` command in MPLS-TE auto bandwidth interface configuration mode. When the application period timer expires, and when the difference between the measured and the current bandwidth exceeds the adjustment threshold, the tunnel is reoptimized. Then, the bandwidth samples are cleared to record the new largest output rate at the next interval.

When reoptimizing the LSP with the new bandwidth, a new path request is generated. If the new bandwidth is not available, the last good LSP continues to be used. This way, the network experiences no traffic interruptions.

If minimum or maximum bandwidth values are configured for a tunnel, the bandwidth, which the automatic bandwidth signals, stays within these values.

**Note**

When more than 100 tunnels are `auto-bw` enabled, the algorithm will jitter the first application of every tunnel by a maximum of 20% (max 1 hour). The algorithm does this to avoid too many tunnels running auto bandwidth applications at the same time.
If a tunnel is shut down, and is later brought again, the adjusted bandwidth is lost and the tunnel is brought back with the initial configured bandwidth. In addition, the application period is reset when the tunnel is brought back.

**Related Topics**
- Configuring the Collection Frequency, on page 212
- Configuring the Automatic Bandwidth Functions, on page 214
- Configure Automatic Bandwidth: Example, on page 268

**Adjustment Threshold**

*Adjustment Threshold* is defined as a percentage of the current tunnel bandwidth and an absolute (minimum) bandwidth. Both thresholds must be fulfilled for the automatic bandwidth to resignal the tunnel. The tunnel bandwidth is resized only if the difference between the largest sample output rate and the current tunnel bandwidth is larger than the adjustment thresholds.

For example, assume that the automatic bandwidth is enabled on a tunnel in which the highest observed bandwidth $B$ is 30 Mbps. Also, assume that the tunnel was initially configured for 45 Mbps. Therefore, the difference is 15 mbit/s. Now, assuming the default adjustment thresholds of 10% and 10 kbps, the tunnel is signalled with 30 Mbps when the application timer expires. This is because 10% of 45 Mbit/s is 4.5 Mbit/s, which is smaller than 15 Mbit/s. The absolute threshold, which by default is 10 kbps, is also crossed.

**Overflow Detection**

Overflow detection is used if a bandwidth must be resized as soon as an overflow condition is detected, without having to wait for the expiry of an automatic bandwidth application frequency interval.

For overflow detection one configures a limit $N$, a percentage threshold $Y\%$ and optionally, a minimum bandwidth threshold $Z$. The percentage threshold is defined as the percentage of the actual signalled tunnel bandwidth. When the difference between the measured bandwidth and the actual bandwidth are both larger than $Y\%$ and $Z$ threshold, for $N$ consecutive times, then the system triggers an overflow detection.

The bandwidth adjustment by the overflow detection is triggered only by an increase of traffic volume through the tunnel, and not by a decrease in the traffic volume. When you trigger an overflow detection, the automatic bandwidth application interval is reset.

By default, the overflow detection is disabled and needs to be manually configured.

**Underflow Detection**

Underflow detection is used when the bandwidth on a tunnel drops significantly, which is similar to overflow but in reverse.

Underflow detection applies the highest bandwidth value from the samples which triggered the underflow. For example, if you have an underflow limit of three, and the following samples trigger the underflow for 10 kbps, 20 kbps, and 15 kbps, then, 20 kbps is applied.

Unlike overflow, the underflow count is not reset across an application period. For example, with an underflow limit of three, you can have the first two samples taken at the end of an application period and then the underflow gets triggered by the first sample of the next application period.
Restrictions for MPLS-TE Automatic Bandwidth

When the automatic bandwidth cannot update the tunnel bandwidth, the following restrictions are listed:

• Tunnel is in a fast reroute (FRR) backup, active, or path protect active state. This occurs because of the assumption that protection is a temporary state, and there is no need to reserve the bandwidth on a backup tunnel. You should prevent taking away the bandwidth from other primary or backup tunnels.

• Reoptimization fails to occur during a lockdown. In this case, the automatic bandwidth does not update the bandwidth unless the bandwidth application is manually triggered by using the `mpls traffic-eng auto-bw apply` command in EXEC mode.

Related Topics

Forcing the Current Application Period to Expire Immediately, on page 214

Point-to-Multipoint Traffic-Engineering

Point-to-Multipoint Traffic-Engineering Overview

The Point-to-Multipoint (P2MP) Resource Reservation Protocol-Traffic Engineering (RSVP-TE) solution allows service providers to implement IP multicast applications, such as IPTV and real-time video, broadcast over the MPLS label switch network. The RSVP-TE protocol is extended to signal point-to-point (P2P) and P2MP label switched paths (LSPs) across the MPLS and GMPLS networks.

By using RSVP-TE extensions as defined in RFC 4875, multiple subLSPs are signaled for a given TE source. The P2MP tunnel is considered as a set of Source-to-Leaf (S2L) subLSPs that connect the TE source to multiple leaf Provider Edge (PE) nodes.

At the TE source, the ingress point of the P2MP-TE tunnel, IP multicast traffic is encapsulated with a unique MPLS label, which is associated with the P2MP-TE tunnel. The traffic continues to be label-switched in the P2MP tree. If needed, the labeled packet is replicated at branch nodes along the P2MP tree. When the labeled packet reaches the egress leaf (PE) node, the MPLS label is removed and forwarded onto the IP multicast tree across the PE-CE link.

To enable end-to-end IP multicast connectivity, RSVP is used in the MPLS-core for P2MP-TE signaling and PIM is used for PE-CE link signaling.

• All edge routers are running PIM-SSM or Source-Specific Multicast (SSM) to exchange multicast routing information with the directly-connected Customer Edge (CE) routers.

• In the MPLS network, RSVP P2MP-TE replaces PIM as the tree building mechanism, RSVP-TE grafts or prunes a given P2MP tree when the end-points are added or removed in the TE source configuration (explicit user operation).

These are the definitions for Point-to-Multipoint (P2MP) tunnels:

Source

Configures the node in which Label Switched Path (LSP) signaling is initiated.
Mid-point
Specifies the transit node in which LSP signaling is processed (for example, not a source or receiver).

Receiver, Leaf, and Destination
Specifies the node in which LSP signaling ends.

Branch Point
Specifies the node in which packet replication is performed.

Source-to-Leaf (S2L) SubLSP
Specifies the P2MP-TE LSP segment that runs from the source to one leaf.

Point-to-Multipoint Traffic-Engineering Features
• P2MP RSVP-TE (RFC 4875) is supported. RFC 4875 is based on nonaggregate signaling; for example, per S2L signaling. Only P2MP LSP is supported.
• `interface tunnel-mte` command identifies the P2MP interface type.
• P2MP tunnel setup is supported with label replication.
• Fast-Reroute (FRR) protection is supported with sub-50 msec for traffic loss.
• Explicit routing is supported by using under utilized links.
• Reoptimization is supported by calculating a better set of paths to the destination with no traffic loss.

Note
Per-S2L reoptimization is not supported.

• IPv4 and IPv6 payloads are supported.
• IPv4 and IPv6 multicast forwarding are supported on a P2MP tunnel interface through a static IGMP and MLD group configuration.
• Both IP multicast and P2MP Label Switch Multicast (LSM) coexist in the same network; therefore, both use the same forwarding plane (LFIB or MPLS Forwarding Infrastructure [MFI]).
• P2MP label replication supports only Source-Specific Multicast (SSM) traffic. SSM configuration supports the default value, none.
• Static mapping for multicast groups to the P2MP-TE tunnel is required.

Point-to-Multipoint Traffic-Engineering Benefits
• Single point of traffic control ensures that signaling and path engineering parameters (for example, protection and diversity) are configured only at the TE source node.
• Ability to configure explicit paths to enable optimized traffic distribution and prevention of single point of failures in the network.
• Link protection of MPLS-labeled traffic traversing branch paths of the P2MP-TE tree.
Ability to do bandwidth Admission Control (AC) during setup and signaling of P2MP-TE paths in the MPLS network.

Related Topics

- Enabling Multicast Routing on the Router, on page 228
- Configure Point-to-Multipoint for the Source: Example, on page 277
- Configure the Point-to-Multipoint Solution: Example, on page 279
- Disabling Destinations, on page 234
- Disable a Destination: Example, on page 279
- Logging Per Destinations for Point-to-Multipoint, on page 236
- Configure the Point-to-Multipoint Tunnel: Example, on page 278
- Configure the Point-to-Multipoint Solution: Example, on page 279
- Point-to-Multipoint RSVP-TE, on page 138

Point-to-Multipoint RSVP-TE

RSVP-TE signals a P2MP tunnel base that is based on a manual configuration. If all Source-to-Leaf (S2L)s use an explicit path, the P2MP tunnel creates a static tree that follows a predefined path based on a constraint such as a deterministic Label Switched Path (LSP). If the S2L uses a dynamic path, RSVP-TE creates a P2MP tunnel base on the best path in the RSVP-TE topology. RSVP-TE supports bandwidth reservation for constraint-based routing.

When an explicit path option is used, specify both the local and peer IP addresses in the explicit path option, provided the link is a GigabitEthernet or a TenGigE based interface. For point-to-point links like POS or bundle POS, it is sufficient to mention the remote or peer IP address in the explicit path option.

RSVP-TE distributes stream information in which the topology tree does not change often (where the source and receivers are). For example, large scale video distribution between major sites is suitable for a subset of multicast applications. Because multicast traffic is already in the tunnel, the RSVP-TE tree is protected as long as you build a backup path.

Fast-Reroute (FRR) capability is supported for P2MP RSVP-TE by using the unicast link protection. You can choose the type of traffic to go to the backup link.

The P2MP tunnel is applicable for all TE Tunnel destination (IntraArea and InterArea). Inter-AS is not supported.

The P2MP tunnel is signaled by the dynamic and explicit path option in the IGP intra area. Only interArea and interAS, which are used for the P2MP tunnels, are signaled by the verbatim path option.

Related Topics

- Configuring the Static Group for the Point-to-Multipoint Interface, on page 230
- Configure Point-to-Multipoint for the Source: Example, on page 277
- Configure the Point-to-Multipoint Solution: Example, on page 279
- Point-to-Multipoint Fast Reroute, on page 139
Point-to-Multipoint Fast Reroute

MPLS-TE Fast Reroute (FRR) is a mechanism to minimize interruption in traffic delivery to a TE Label Switched Path (LSP) destination as a result of link failures. FRR enables temporarily fast switching of LSP traffic along an alternative backup path around a network failure, until the TE tunnel source signals a new end-to-end LSP.

Both Point-to-Point (P2P) and P2MP-TE support only the Facility FRR method from RFC 4090. P2P LSPs are used to backup P2MP S2L (source 2 Leaf). Only link and bandwidth protection for P2MP S2Ls are supported. Node protection is not supported.

MPLS-TE link protection relies on the fact that labels for all primary LSPs and subLSPs are using the MPLS global label allocation. For example, one single (global) label space is used for all MPLS-TE enabled physical interfaces on a given MPLS LSP.

Related Topics

- Point-to-Multipoint Traffic-Engineering Overview, on page 136
- Point-to-Multipoint RSVP-TE, on page 138

Point-to-Multipoint Label Switch Path

The Point-to-Multipoint Label Switch Path (P2MP LSP) has only a single root, which is the Ingress Label Switch Router (LSR). The P2MP LSP is created based on a receiver that is connected to the Egress LSR. The Egress LSR initiates the creation of the tree (for example, tunnel grafting or pruning is done by performing an individual sub-LSP operation) by creating the Forwarding Equivalency Class (FEC) and Opaque Value.

Note

Grafting and pruning operate on a per destination basis.

The Opaque Value contains the stream information that uniquely identifies the tree to the root. To receive label switched multicast packets, the Egress Provider Edge (PE) indicates to the upstream router (the next hop closest to the root) which label it uses for the multicast source by applying the label mapping message.

The upstream router does not need to have any knowledge of the source; it needs only the received FEC to identify the correct P2MP LSP. If the upstream router does not have any FEC state, it creates it and installs the assigned downstream outgoing label into the label forwarding table. If the upstream router is not the root of the tree, it must forward the label mapping message to the next hop upstream. This process is repeated hop-by-hop until the root is reached.

By using downstream allocation, the router that wants to receive the multicast traffic assigns the label for it. The label request, which is sent to the upstream router, is similar to an unsolicited label mapping (that is, the upstream does not request it). The upstream router that receives that label mapping uses the specific label to send multicast packets downstream to the receiver. The advantage is that the router, which allocates the labels, does not get into a situation where it has the same label for two different multicast sources. This is because it manages its own label space allocation locally.

Path Option for Point-to-Multipoint RSVP-TE

P2MP tunnels are signaled by using the dynamic and explicit path-options in an IGP intra area. InterArea cases for P2MP tunnels are signaled by the verbatim path option.
Path options for P2MP tunnels are individually configured for each sub-LSP. Only one path option per sub-LSP (destination) is allowed. You can choose whether the corresponding sub-LSP is dynamically or explicitly routed. For the explicit option, you can configure the verbatim path option to bypass the topology database lookup and verification for the specified destination.

Both dynamic and explicit path options are supported on a per destination basis by using the `path-option` (P2MP-TE) command. In addition, you can combine both path options.

**Explicit Path Option**

Configures the intermediate hops that are traversed by a sub-LSP going from the TE source to the egress MPLS node. Although an explicit path configuration enables granular control of sub-LSP paths in an MPLS network, multiple explicit paths are configured for specific network topologies with a limited number of (equal cost) links or paths.

**Dynamic Path Option**

Computes the IGP path of a P2MP tree sub-LSP that is based on the OSPF and ISIS algorithm. The TE source is dynamically calculated based on the IGP topology.

---

**Note**

Dynamic path option can only compute fully-diverse standby paths. While, explicit path option supports partially diverse standby paths as well.

---

**Dynamic Path Calculation Requirements**

Dynamic path calculation for each sub-LSP uses the same path parameters as those for the path calculation of regular point-to-point TE tunnels. As part of the sub-LSP path calculation, the link resource (bandwidth) is included, which is flooded throughout the MPLS network through the existing RSVP-TE extensions to OSPF and ISIS. Instead of dynamic calculated paths, explicit paths are also configured for one or more sub-LSPs that are associated with the P2MP-TE tunnel.

- OSPF or ISIS are used for each destination.
- TE topology and tunnel constraints are used to input the path calculation.
- Tunnel constraints such as affinity, bandwidth, and priorities are used for all destinations in a tunnel.
- Path calculation yields an explicit route to each destination.

**Static Path Calculation Requirements**

The static path calculation does not require any new extensions to IGP to advertise link availability.

- Explicit path is required for every destination.
- Offline path calculation is used.
- TE topology database is not needed.
- If the topology changes, reoptimization is not required.

**Related Topics**

Configuring Destinations for the Tunnel Interface, on page 231
MPLS Traffic Engineering Shared Risk Link Groups

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

OSPF and Intermediate System-to-Intermediate System (IS-IS) flood the SRLG value information (including other TE link attributes such as bandwidth availability and affinity) using a sub-type length value (sub-TLV), so that all routers in the network have the SRLG information for each link.

To activate the SRLG feature, configure the SRLG value of each link that has a shared risk with another link. A maximum of 30 SRLGs per interface is allowed. You can configure this feature on multiple interfaces including the bundle interface.

Figure 15: Shared Risk Link Group illustrates the MPLS TE SRLG values configured on the bundle interface.

Related Topics

  Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
  Creating an Explicit Path With Exclude SRLG, on page 219
  Using Explicit Path With Exclude SRLG, on page 220
  Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
  Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
  Configure the MPLS-TE Shared Risk Link Groups: Example, on page 268
Explicit Path

The Explicit Path configuration allows you to configure the explicit path. An IP explicit path is a list of IP addresses, each representing a node or link in the explicit path.

The MPLS Traffic Engineering (TE)—IP Explicit Address Exclusion feature provides a means to exclude a link or node from the path for an Multiprotocol Label Switching (MPLS) TE label-switched path (LSP).

This feature is enabled through the `explicit-path` command that allows you to create an IP explicit path and enter a configuration submode for specifying the path. The feature adds to the submode commands of the `exclude-address` command for specifying addresses to exclude from the path.

The feature also adds to the submode commands of the `exclude-srlg` command that allows you to specify the IP address to get SRLGs to be excluded from the explicit path.

If the excluded address or excluded srlg for an MPLS TE LSP identifies a flooded link, the constraint-based shortest path first (CSPF) routing algorithm does not consider that link when computing paths for the LSP. If the excluded address specifies a flooded MPLS TE router ID, the CSPF routing algorithm does not allow paths for the LSP to traverse the node identified by the router ID.

Related Topics

- Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
- Creating an Explicit Path With Exclude SRLG, on page 219
- Using Explicit Path With Exclude SRLG, on page 220
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 268

Fast ReRoute with SRLG Constraints

Fast ReRoute (FRR) protects MPLS TE Label Switch Paths (LSPs) from link and node failures by locally repairing the LSPs at the point of failure. This protection allows data to continue to flow on LSPs, while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or nodes.

Backup tunnels that bypass only a single link of the LSP’s path provide Link Protection. They protect LSPs by specifying the protected link IP addresses to extract SRLG values that are to be excluded from the explicit path, thereby bypassing the failed link. These are referred to as next-hop (NHOP) backup tunnels because
they terminate at the LSP's next hop beyond the point of failure. Figure 16: NHOP Backup Tunnel with SRLG constraint illustrates an NHOP backup tunnel.

Figure 16: NHOP Backup Tunnel with SRLG constraint

In the topology shown in the above figure, the backup tunnel path computation can be performed in this manner:

- Get all SRLG values from the exclude-SRLG link (SRLG values 5 and 6)
- Mark all the links with the same SRLG value to be excluded from SPF
- Path computation as CSPF R2->R6->R7->R3

FRR provides Node Protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called NNHOP backup tunnels because they terminate at the node following the next-hop node of the LSP paths, thereby bypassing the next-hop node. They protect LSPs when a node along their path fails, by enabling the node upstream to the point of failure to reroute the LSPs and their traffic, around the failed node to the next-next hop. They also protect LSPs by specifying the protected link IP addresses that are to be excluded from the explicit path, and the SRLG values associated with the IP addresses excluded from the explicit path.
NNHOP backup tunnels also provide protection from link failures by bypassing the failed link as well as the node. Figure 17: NNHOP Backup Tunnel with SRLG constraint illustrates an NNHOP backup tunnel.

Figure 17: NNHOP Backup Tunnel with SRLG constraint

In the topology shown in the above figure, the backup tunnel path computation can be performed in this manner:

- Get all SRLG values from the exclude-SRLG link (SRLG values 5 and 6)
- Mark all links with the same SRLG value to be excluded from SPF
- Verify path with SRLG constraint
- Path computation as CSPF R2->R9->R10->R4

Related Topics
Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
Creating an Explicit Path With Exclude SRLG, on page 219
Using Explicit Path With Exclude SRLG, on page 220
Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
Configure the MPLS-TE Shared Risk Link Groups: Example, on page 268

Importance of Protection
This section describes the following:
- Delivery of Packets During a Failure
- Multiple Backup Tunnels Protecting the Same Interface
Related Topics

Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
Creating an Explicit Path With Exclude SRLG, on page 219
Using Explicit Path With Exclude SRLG, on page 220
Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
Configure the MPLS-TE Shared Risk Link Groups: Example, on page 268

Delivery of Packets During a Failure

Backup tunnels that terminate at the NNHOP protect both the downstream link and node. This provides protection for link and node failures.

Related Topics

Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
Creating an Explicit Path With Exclude SRLG, on page 219
Using Explicit Path With Exclude SRLG, on page 220
Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
Configure the MPLS-TE Shared Risk Link Groups: Example, on page 268

Multiple Backup Tunnels Protecting the Same Interface

• Redundancy—If one backup tunnel is down, other backup tunnels protect LSPs.
• Increased backup capacity—If the protected interface is a high-capacity link and no single backup path exists with an equal capacity, multiple backup tunnels can protect that one high-capacity link. The LSPs using this link falls over to different backup tunnels, allowing all of the LSPs to have adequate bandwidth protection during failure (rerouting). If bandwidth protection is not desired, the router spreads LSPs across all available backup tunnels (that is, there is load balancing across backup tunnels).

Related Topics

Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
Creating an Explicit Path With Exclude SRLG, on page 219
Using Explicit Path With Exclude SRLG, on page 220
Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
Configure the MPLS-TE Shared Risk Link Groups: Example, on page 268

SRLG Limitations

There are few limitations to the configured SRLG feature:
• The exclude-address and exclude-srlg options are not allowed in the IP explicit path strict-address network.
• Whenever SRLG values are modified after tunnels are signalled, they are verified dynamically in the next path verification cycle.

Related Topics
- Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
- Creating an Explicit Path With Exclude SRLG, on page 219
- Using Explicit Path With Exclude SRLG, on page 220
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 268

Soft-Preemption
MPLS-TE preemption consists of freeing the resources of an established LSP, and assigning them to a new LSP. The freeing of resources causes a traffic disruption to the LSP that is being preempted. Soft preemption is an extension to the RSVP-TE protocol to minimize and even eliminate such traffic disruption over the preempted LSP.

The soft-preemption feature attempts to preempt the LSPs in a graceful manner to minimize or eliminate traffic loss. However, the link might be over-subscribed for a period of time.

In a network that implements soft preemption, zero traffic loss is achieved in this manner:
• When signaling a new LSP, the ingress router indicates to all the intermediate nodes that the existing LSP is to be softly preempted, in case its resources are needed and is to be reassigned.
• When a given intermediate node needs to soft-preempt the existing LSP, it sends a new or special path error (preemption pending) to the ingress router. The intermediate node does not dismantle the LSP and maintains its state.
• When the ingress router receives the path error (preemption pending) from the intermediate node, it immediately starts a re-optimization that avoids the link that caused the preemption.
• When the re-optimization is complete, the ingress router tears down the soft-preempted LSP.

Related Topics
- Enabling Soft-Preemption on a Node, on page 238
- Enabling Soft-Preemption on a Tunnel, on page 239

Path Option Attributes
The path option attributes are configurable through a template configuration. This template, named attribute-set, is configured globally in the MPLS traffic-engineering mode.

You can apply an attribute-set to a path option on a per-LSP basis. The path option configuration is extended to take a path option attribute name. LSPs computed with a particular path option uses the attributes as specified by the attribute-set under that path option.

These prerequisites are required to implement path option attributes:
• Path option type attribute-set is configured in the MPLS TE mode
- Path option CLI extended to accept an attribute-set name

**Note** The **signalled-bandwidth** and **affinity** attributes are supported under the attribute-set template.

**Related Topics**
- Configuring Attributes within a Path-Option Attribute, on page 240

**Configuration Hierarchy of Path Option Attributes**

You can specify a value for an attribute within a path option attribute-set template. This does not prevent the configuring of the same attribute at a tunnel level. However, it is important to note that only one level is taken into account. So, the configuration at the LSP level is considered more specific than the one at the level of the tunnel, and it is used from this point onwards.

Attributes that are not specified within an attribute-set take their values as usual—configuration at the tunnel level, configuration at the global MPLS level, or default values. Here is an example:

```
attribute-set path-option MYSET
  affinity 0xBEEF mask 0xBEEF

interface tunnel-te 10
  affinity 0xCAFE mask 0xCAFE
  signalled-bandwidth 1000
  path-option 1 dynamic attribute-set name MYSET
  path-option 2 dynamic
```

In this example, the attribute-set named **MYSET** is specifying affinity as 0xBEEF. The signalled bandwidth has not been configured in this **MYSET**. The **tunnel 10**, meanwhile, has affinity 0xCAFE configured. LSPs computed from path-option 1 uses the affinity 0xBEEF/0xBEEF, while LSPs computed from path-option 2 uses the affinity 0xCAFE/0xCAFE. All LSPs computed using any of these path-options use **signalled-bandwidth** as 1000, as this is the only value that is specified only at the tunnel level.

**Note** The attributes configured in a path option attribute-set template takes precedence over the same attribute configured under a tunnel. An attribute configured under a tunnel is used only if the equivalent attribute is **not** specified by the in-use path option attribute-set template.

**Related Topics**
- Configuring Attributes within a Path-Option Attribute, on page 240

**Traffic Engineering Bandwidth and Bandwidth Pools**

MPLS traffic engineering allows constraint-based routing (CBR) of IP traffic. One of the constraints satisfied by CBR is the availability of required bandwidth over a selected path. Regular TE tunnel bandwidth is called the **global pool**. The **subpool bandwidth** is a portion of the global pool. If it is not in use, the subpool bandwidth is not reserved from the global pool. Therefore, subpool tunnels require a priority higher than that of non-subpool tunnels.
You can configure the signalled-bandwidth path option attribute to use either the global pool (default) or the subpool bandwidth. The signalled-bandwidth value for the path option may be any valid value and the pool does not have to be the same as that which is configured on the tunnel.

---

**Note**

When you configure signalled-bandwidth for path options with the `signalled-bandwidth bandwidth [sub-pool | global] kbps` command, use either all subpool bandwidths or all global-pool bandwidth values.

---

**Related Topics**

- Configuring Attributes within a Path-Option Attribute, on page 240

---

### Path Option Switchover

Reoptimization to a particular path option is not possible if the in-use path option and the new path option do not share the same bandwidth class. The path option switchover operation would fail in such a scenario. Use this command at the EXEC configuration mode to switchover to a newer path option:

```
mpls traffic-eng switchover tunnel-xx ID path-option index
```

The switchover to a newer path option is achieved, in these instances:
- when a lower index path option is available
- when any signalling message or topology update causes the primary LSP to go down
- when a local interface fails on the primary LSP or a path error is received on the primary LSP

---

**Note**

Path option switchover between various path options with different bandwidth classes is not allowed.

---

**Related Topics**

- Configuring Attributes within a Path-Option Attribute, on page 240

---

### Path Option and Path Protection

When path-protection is enabled, a standby LSP is established to protect traffic going over the tunnel. The standby LSP may be established using either the same path option as the primary LSP, or a different one.

The standby LSP is computed to be diverse from the primary LSP, so bandwidth class differences does not matter. This is true in all cases of diversity except node-diversity. With node diversity, it is possible for the standby LSP to share up to two links with the primary LSP, the link exiting the head node, and the link entering the tail node.

If you want to switchover from one path option to another path option and these path options have different classes, the path option switchover is rejected. However, the path option switchover can not be blocked in the path-protection feature. When the standby LSP becomes active using another path option of a different class type, the path option switchover cannot be rejected at the head end. It might get rejected by the downstream node.
Node-diversity is only possible under limited conditions. The conditions that must be met are:

- there is no second path that is both node and link diverse
- the current LSP uses a shared-media link at the head egress or tail ingress
- the shared-media link used by the current LSP permits computation of a node-diverse path

In Cisco IOS XR, reoptimization between different class types would actually be rejected by the next hop. This rejection will occur by an admission failure.

**Related Topics**

Configuring Attributes within a Path-Option Attribute, on page 240

---

**Auto-Tunnel Mesh**

The MPLS traffic engineering auto-tunnel mesh (Auto-mesh) feature allows you to set up full mesh of TE P2P tunnels automatically with a minimal set of MPLS traffic engineering configurations. You may configure one or more mesh-groups. Each mesh-group requires a destination-list (IPv4 prefix-list) listing destinations, which are used as destinations for creating tunnels for that mesh-group.

You may configure MPLS TE auto-mesh type attribute-sets (templates) and associate them to mesh-groups. LSR creates tunnels using the tunnel properties defined in the attribute-set.

Auto-Tunnel mesh provides benefits:

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

- Minimizes future configurations resulting due to network growth.
  It eliminates the need to reconfigure each existing TE LSR in order to establish a full mesh of TE tunnels whenever a new TE LSR is added in the network.

**Related Topics**

Configuring Auto-Tunnel Mesh Tunnel ID, on page 241
Configuring Auto-tunnel Mesh Unused Timeout, on page 242
Configuring Auto-Tunnel Mesh Group, on page 243
Configuring Tunnel Attribute-Set Templates, on page 245
Enabling LDP on Auto-Tunnel Mesh, on page 246

---

**Destination List (Prefix-List)**

Auto-mesh tunnels can be automatically created using prefix-list. Each TE enabled router in the network learns about the TE router IDs through a existing IGP extension.

You can view the router IDs on the router using this command:

```
show mpls traffic-eng topology | include TE Id
IGP Id: 0001.0000.0010.00, MPLS TE Id:100.1.1.1 Router Node (ISIS 1 level-2)
```
How to Implement Traffic Engineering

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.

These procedures are used to implement MPLS-TE:

Building MPLS-TE Topology

Perform this task to configure MPLS-TE topology (required for traffic engineering tunnel operations).

Before You Begin

Before you start to build the MPLS-TE topology, you must have enabled:

- IGP such as OSPF or IS-IS for MPLS-TE.
- MPLS Label Distribution Protocol (LDP).
- RSVP on the port interface.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.
- If you are going to use nondefault holdtime or intervals, you must decide the values to which they are set.

Related Topics

- Configuring Auto-Tunnel Mesh Tunnel ID, on page 241
- Configuring Auto-tunnel Mesh Unused Timeout, on page 242
- Configuring Auto-Tunnel Mesh Group, on page 243
- Configuring Tunnel Attribute-Set Templates, on page 245
- Enabling LDP on Auto-Tunnel Mesh, on page 246
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. exit
5. exit
6. router ospf process-name
7. area area-id
8. exit
9. mpls traffic-eng router-id ip-address
10. commit
11. (Optional) show mpls traffic-eng topology
12. (Optional) show mpls traffic-eng link-management advertisements

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>mpls traffic-eng</td>
<td>Enables traffic engineering on a particular interface on the originating node and enters MPLS-TE interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>interface type interface-path-id</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#interface POS0/6/0/0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if)#</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)#</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>router ospf process-name</strong></td>
<td>Enters a name for the OSPF process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>router ospf 1</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>area area-id</strong></td>
<td>Configures an area for the OSPF process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>area 0</strong></td>
<td></td>
</tr>
<tr>
<td>• Backbone areas have an area ID of 0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Non-backbone areas have a non-zero area ID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>exit</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>exit</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>mpls traffic-eng router-id ip-address</strong></td>
<td>Sets the MPLS-TE loopback interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>mpls traffic-eng router-id 192.168.70.1</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>commit</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>show mpls traffic-eng topology</strong></td>
<td>(Optional) Verifies the traffic engineering topology.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>show mpls traffic-eng topology</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>show mpls traffic-eng link-management advertisements</strong></td>
<td>(Optional) Displays all the link-management advertisements for the links on this node.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>show mpls traffic-eng link-management advertisements</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

How MPLS-TE Works, on page 110
Build MPLS-TE Topology and Tunnels: Example, on page 258
Creating an MPLS-TE Tunnel

Creating an MPLS-TE tunnel is a process of customizing the traffic engineering to fit your network topology. Perform this task to create an MPLS-TE tunnel after you have built the traffic engineering topology.

Before You Begin

The following prerequisites are required to create an MPLS-TE tunnel:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.
- If you are going to use nondefault holdtime or intervals, you must decide the values to which they are set.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. destination ip-address
4. ipv4 unnumbered type interface-path-id
5. path-option preference - priority dynamic
6. signalled- bandwidth {bandwidth [class-type ct ] | sub-pool bandwidth}
7. commit
8. (Optional) show mpls traffic-eng tunnels
9. (Optional) show ipv4 interface brief
10. (Optional) show mpls traffic-eng link-management admission-control

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>destination ip-address</td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# destination</td>
<td>The destination address is the remote node’s MPLS-TE router ID.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>192.168.92.125</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv4 unnumbered type interface-path-id</td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/O/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
</tr>
<tr>
<td><strong>Step 5</strong> path-option preference - priority dynamic</td>
<td>Sets the path option to dynamic and assigns the path ID.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/O/RP0/CPU0:router(config-if)# path-option 1 dynamic</td>
</tr>
<tr>
<td><strong>Step 6</strong> signalled-bandwidth { bandwidth [class-type ct ]</td>
<td>sub-pool bandwidth }</td>
<td>Sets the CTo bandwidth required on this interface. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/O/RP0/CPU0:router(config-if)# signalled-bandwidth 100</td>
</tr>
<tr>
<td><strong>Step 7</strong> commit</td>
<td>(Optional)</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show mpls traffic-eng tunnels</td>
<td>Verifies that the tunnel is connected (in the UP state) and displays all configured TE tunnels.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/O/RP0/CPU0:router# show mpls traffic-eng tunnels</td>
</tr>
<tr>
<td><strong>Step 9</strong> show ipv4 interface brief</td>
<td>(Optional) Displays all TE tunnel interfaces.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/O/RP0/CPU0:router# show ipv4 interface brief</td>
</tr>
<tr>
<td><strong>Step 10</strong> show mpls traffic-eng link-management admission-control</td>
<td>(Optional) Displays all the tunnels on this node.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td>RP/O/RP0/CPU0:router# show mpls traffic-eng link-management admission-control</td>
</tr>
</tbody>
</table>
Configuring Forwarding over the MPLS-TE Tunnel

Perform this task to configure forwarding over the MPLS-TE tunnel created in the previous task. This task allows MPLS packets to be forwarded on the link between network neighbors.

Before You Begin

The following prerequisites are required to configure forwarding over the MPLS-TE tunnel:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

**SUMMARY STEPS**

1. configure
2. interface tunnel-te tunnel-id
3. ipv4 unnumbered type interface-path-id
4. autoroute announce
5. exit
6. router static address-family ipv4 unicast prefix mask ip-address interface type
7. commit
8. (Optional) ping {ip-address | hostname}
9. (Optional) show mpls traffic-eng autoroute

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-te tunnel-id</td>
<td>Enters MPLS-TE interface configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
```
### Implementing MPLS Traffic Engineering

#### Configuring Forwarding over the MPLS-TE Tunnel

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td><code>ipv4 unnumbered type interface-path-id</code></td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>autoroute announce</code></td>
<td>Enables messages that notify the neighbor nodes about the routes that are forwarding.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-if)# autoroute announce</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>exit</code></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config-if)# exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>router static address-family ipv4 unicast prefix mask ip-address interface type</code></td>
<td>Enables a route using IP version 4 addressing, identifies the destination address and the tunnel where forwarding is enabled. This configuration is used for static routes when the <code>autoroute announce</code> command is not used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router(config)# router static address-family ipv4 unicast 2.2.2.2/32 tunnel-te 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>`ping {ip-address</td>
<td>hostname}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router# ping 192.168.12.52</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><code>show mpls traffic-eng autoroute</code></td>
<td>(Optional) Verifies forwarding by displaying what is advertised to IGP for the TE tunnel.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RP0/CPU0:router# show mpls traffic-eng autoroute</code></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Overview of MPLS Traffic Engineering, on page 109
- Creating an MPLS-TE Tunnel, on page 153
Protecting MPLS Tunnels with Fast Reroute

Perform this task to protect MPLS-TE tunnels, as created in the previous task.

Note

Although this task is similar to the previous task, its importance makes it necessary to present as part of the tasks required for traffic engineering on Cisco IOS XR software.

Before You Begin

The following prerequisites are required to protect MPLS-TE tunnels:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.
- You must first configure a primary tunnel.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. fast-reroute
4. exit
5. mpls traffic-eng
6. interface type interface-path-id
7. backup-path tunnel-te tunnel-number
8. exit
9. exit
10. interface tunnel-te tunnel-id
11. backup-bw {backup bandwidth | sub-pool {bandwidth | unlimited} | global-pool {bandwidth | unlimited}}
12. ipv4 unnumbered type interface-path-id
13. path-option preference-priority {explicit name explicit-path-name}
14. destination ip-address
15. commit
16. (Optional) show mpls traffic-eng tunnels backup
17. (Optional) show mpls traffic-eng tunnels protection frr
18. (Optional) show mpls traffic-eng fast-reroute database
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>fast-reroute</td>
<td>Enables fast reroute.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# fast-reroute</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>interface type interface-path-id</td>
<td>Enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# interface pos0/6/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>backup-path tunnel-te tunnel-number</td>
<td>Sets the backup path to the backup tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if)# backup-path tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if)# exit</td>
<td></td>
</tr>
</tbody>
</table>

---

**Cisco IOS XR MPLS Configuration Guide for the Cisco CRS Router, Release 4.3.x**
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)#</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>backup-bw {backup bandwidth | sub-pool {bandwidth | unlimited} | global-pool {bandwidth | unlimited} }</td>
<td>Sets the CT0 bandwidth required on this interface.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>Because the default tunnel priority is 7, tunnels use the default TE class map.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# backup-bw global-pool 5000</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ipv4 unnumbered type interface-path-id</td>
<td>Assigns a source address to set up forwarding on the new tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>path-option preference-priority {explicit name explicit-path-name}</td>
<td>Sets the path option to explicit with a given name (previously configured) and assigns the path ID.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-path</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>destination ip-address</td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* Destination address is the remote node's MPLS-TE router ID.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* Destination address is the merge point between backup and protected tunnels.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.</td>
</tr>
<tr>
<td>15</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>show mpls traffic-eng tunnels backup</td>
<td>(Optional) Displays the backup tunnel information.</td>
<td></td>
</tr>
<tr>
<td><code>Example:</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>backup</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>show mpls traffic-eng tunnels protection frr</td>
<td>(Optional) Displays the tunnel protection information for Fast-Reroute (FRR).</td>
<td></td>
</tr>
<tr>
<td><code>Example:</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>protection frr</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>show mpls traffic-eng fast-reroute database</td>
<td>(Optional) Displays the protected tunnel state (for example, the tunnel’s current ready or active state).</td>
<td></td>
</tr>
<tr>
<td><code>Example:</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>fast-reroute database</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**
- Fast Reroute, on page 119
- Fast Reroute Node Protection, on page 126
- Creating an MPLS-TE Tunnel, on page 153
- Configuring Forwarding over the MPLS-TE Tunnel, on page 155

**Enabling an AutoTunnel Backup**

Perform this task to configure the AutoTunnel Backup feature. By default, this feature is disabled. You can configure the AutoTunnel Backup feature for each interface. It has to be explicitly enabled for each interface or link.

**SUMMARY STEPS**

1. configure
2. ipv4 unnumbered mpls traffic-eng Loopback 0
3. mpls traffic-eng
4. auto-tunnel backup timers removal unused frequency
5. auto-tunnel backup tunnel-id min min max
6. commit
7. show mpls traffic-eng auto-tunnel backup summary
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 ipv4 unnumbered mpls traffic-eng Loopback 0</td>
<td>Configures the globally configured IPv4 address that can be used by the AutoTunnel Backup Tunnels. <strong>Note</strong> Loopback 0 is the router ID. The AutoTunnel Backup tunnels will not come up until a global IPv4 address is configured.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)#ipv4 unnumbered mpls traffic-eng Loopback 0</td>
<td></td>
</tr>
<tr>
<td>Step 3 mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 4 auto-tunnel backup timers removal unused frequency</td>
<td>Configures how frequently a timer scans the backup automatic tunnels and removes tunnels that are not in use. <strong>Note</strong> You can also configure the auto-tunnel backup command at mpls traffic-eng interface mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup timers removal unused 20</td>
<td></td>
</tr>
<tr>
<td>Step 5 auto-tunnel backup tunnel-id min max</td>
<td>Configures the range of tunnel interface numbers to be used for automatic backup tunnels. Range is 0 to 65535.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500</td>
<td></td>
</tr>
<tr>
<td>Step 6 commit</td>
<td></td>
</tr>
<tr>
<td>Step 7 show mpls traffic-eng auto-tunnel backup summary</td>
<td>Displays information about configured MPLS-TE backup autotunnels.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# show mpls traffic-eng auto-tunnel backup summary</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- Backup AutoTunnels, on page 111
- Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 270

### Removing an AutoTunnel Backup

To remove all the backup autotunnels, perform this task to remove the AutoTunnel Backup feature.
SUMMARY STEPS

1. clear mpls traffic-eng auto-tunnel backup unused { all | tunnel-te-number }
2. commit
3. show mpls traffic-eng auto-tunnel summary

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>clear mpls traffic-eng auto-tunnel backup unused { all</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# clear mpls traffic-eng auto-tunnel backup unused all</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>commit</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>show mpls traffic-eng auto-tunnel summary</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng auto-tunnel summary</td>
</tr>
</tbody>
</table>

Related Topics

- Backup AutoTunnels, on page 111
- Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 270

Establishing MPLS Backup AutoTunnels to Protect Fast Reroutable TE LSPs

To establish an MPLS backup autotunnel to protect fast reroutable TE LSPs, perform these steps:

SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. auto-tunnel backup
5. attribute-set attribute-set-name
6. commit
7. show mpls traffic-eng auto-tunnel backup summary
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3 interface type interface-path-id</td>
<td>Enables traffic engineering on a specific interface on the originating node.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4 auto-tunnel backup</td>
<td>Enables an auto-tunnel backup feature for the specified interface.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup</td>
<td>Note: You cannot configure the static backup on the similar link.</td>
</tr>
<tr>
<td>Step 5 attribute-set attribute-set-name</td>
<td>Configures attribute-set template for auto-tunnel backup tunnels.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)# attribute-set ab</td>
<td></td>
</tr>
<tr>
<td>Step 6 commit</td>
<td></td>
</tr>
<tr>
<td>Step 7 show mpls traffic-eng auto-tunnel backup summary</td>
<td>Displays information about configured MPLS-TE backup autotunnels.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# show mpls traffic auto-tunnel backup summary</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
- Backup AutoTunnels, on page 111
- Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 270

Establishing Next-Hop Tunnels with Link Protection

To establish a next-hop tunnel and link protection on the primary tunnel, perform these steps:
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. auto-tunnel backup nhop-only
5. auto-tunnel backup exclude srlg [preferred]
6. attribute-set attribute-set-name
7. commit
8. show mpls traffic-eng tunnels number detail

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td>Enables traffic engineering on a specific interface on the originating node.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>auto-tunnel backup nhop-only</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup nhop-only</td>
<td>Enables the creation of dynamic NHOP backup tunnels. By default, both NHOP and NNHOP protection are enabled.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Using this nhop-only option, only link protection is provided.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>auto-tunnel backup exclude srlg [preferred]</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup exclude srlg preferred</td>
<td>Enables the exclusion of SRLG values on a given link for the AutoTunnel backup associated with a given interface. The preferred option allows the AutoTunnel Backup tunnels to come up even if no path excluding all SRLG is found.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>attribute-set attribute-set-name</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if-auto-backup)#attribute-set ab</td>
<td>Configures attribute-set template for auto-tunnel backup tunnels.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>
### Configuring a Prestandard DS-TE Tunnel

Perform this task to configure a Prestandard DS-TE tunnel.

**Before You Begin**

The following prerequisites are required to configure a Prestandard DS-TE tunnel:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

### SUMMARY STEPS

1. configure
2. rsvp interface type interface-path-id
3. bandwidth [total reservable bandwidth] [bc0 bandwidth] [global-pool bandwidth] [sub-pool reservable-bw]
4. exit
5. exit
6. interface tunnel-te tunnel-id
7. signalled-bandwidth {bandwidth [class-type ct] | sub-pool bandwidth}
8. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>2</td>
<td><code>rsvp interface type interface-path-id</code></td>
</tr>
<tr>
<td></td>
<td>Example: <code>RP/0/RP0/CPU0:router(config)# rsvp interface pos0/6/0/0</code></td>
</tr>
<tr>
<td>3</td>
<td><code>bandwidth [total reservable bandwidth] [bc0 bandwidth] [global-pool bandwidth] [sub-pool reservable-bw]</code></td>
</tr>
<tr>
<td></td>
<td>Example: <code>RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 100 150 sub-pool 50</code></td>
</tr>
<tr>
<td>4</td>
<td><code>exit</code></td>
</tr>
<tr>
<td></td>
<td>Example: <code>RP/0/RP0/CPU0:router(config-rsvp-if)# exit</code></td>
</tr>
<tr>
<td>5</td>
<td><code>exit</code></td>
</tr>
<tr>
<td></td>
<td>Example: <code>RP/0/RP0/CPU0:router(config-rsvp)# exit</code></td>
</tr>
<tr>
<td>6</td>
<td><code>interface tunnel-te tunnel-id</code></td>
</tr>
<tr>
<td></td>
<td>Example: <code>RP/0/RP0/CPU0:router(config)# interface tunnel-te 2</code></td>
</tr>
<tr>
<td>7</td>
<td>`signalled-bandwidth {bandwidth [class-type ct]</td>
</tr>
<tr>
<td></td>
<td>Example: <code>RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth sub-pool 10</code></td>
</tr>
<tr>
<td>8</td>
<td><code>commit</code></td>
</tr>
</tbody>
</table>

**Related Topics**

- Configuring Traffic Engineering Tunnel Bandwidth, on page 70
Configuring an IETF DS-TE Tunnel Using RDM

Perform this task to create an IETF mode DS-TE tunnel using RDM.

Before You Begin

The following prerequisites are required to create an IETF mode DS-TE tunnel using RDM:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

SUMMARY STEPS

1. configure
2. rsvp interface type interface-path-id
3. bandwidth rdm \{total-reservable-bw \| bc0 \| global-pool\} \{sub-pool \| bc1 reservable-bw\}
4. exit
5. exit
6. mpls traffic-eng
7. ds-te mode ietf
8. exit
9. interface tunnel-te tunnel-id
10. signalled-bandwidth \{bandwidth \[class-type \| \] \| sub-pool bandwidth\}
11. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters RSVP configuration mode and selects an RSVP interface.</td>
</tr>
<tr>
<td>Step 2 rsvp interface type interface-path-id</td>
<td>Sets the reserved RSVP bandwidth available on this interface by using the Russian Doll Model (RDM) bandwidth constraints model. The range for the total reserve bandwidth argument is 0 to 4294967295.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth</code></td>
<td>Note Physical interface bandwidth is not used by MPLS-TE.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-rsvp-if)# exit</code></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-rsvp) exit</code></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>mpls traffic-eng</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</code></td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>ds-te mode ietf</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-mpls-te)# ds-te mode ietf</code></td>
<td>Enables IETF DS-TE mode and default TE class map. IETF DS-TE mode is configured on all network nodes.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config-mpls-te)# exit</code></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><code>interface tunnel-te tunnel-id</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router(config)# interface tunnel-te 4</code></td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>`signalled-bandwidth {bandwidth [class-type cr]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configures the bandwidth required for an MPLS TE tunnel. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7).</td>
</tr>
</tbody>
</table>
### Configuring an IETF DS-TE Tunnel Using MAM

Perform this task to configure an IETF mode differentiated services traffic engineering tunnel using the Maximum Allocation Model (MAM) bandwidth constraint model.

#### Before You Begin

The following prerequisites are required to configure an IETF mode differentiated services traffic engineering tunnel using the MAM bandwidth constraint model:

- You must have a router ID for the neighboring router.
- Stable router ID is required at either end of the link to ensure that the link is successful. If you do not assign a router ID to the routers, the system defaults to the global router ID. Default router IDs are subject to change, which can result in an unstable link.

#### SUMMARY STEPS

1. configure
2. rsvp interface type interface-path-id
3. bandwidth mam \{total reservable bandwidth | max-reservable-bw maximum-reservable-bw\} \[bc0\|bc1\] reservable bandwidth
4. exit
5. exit
6. mpls traffic-eng
7. ds-te mode ietf
8. ds-te bc-model mam
9. exit
10. interface tunnel-te tunnel-id
11. signalled-bandwidth \{bandwidth [class-type ct] | sub-pool bandwidth\}
12. commit

---

**Example:**

```
RP/0/RP0/CPU0:router(config-if)#
signalled-bandwidth 10 class-type 1
```

**Step 11**

commit

---

**Related Topics**
- Configuring Traffic Engineering Tunnel Bandwidth, on page 70
- Russian Doll Bandwidth Constraint Model, on page 117

---
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>rsvp interface type interface-path-id</td>
<td>Enters RSVP configuration mode and selects the RSVP interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# rsvp interface pos0/6/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>bandwidth mam {total reservable bandwidth</td>
<td>max-reservable-bw maximum-reservable-bw} [bc0 reservable bandwidth] [bc1 reservable bandwidth]</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth mam max-reservable-bw 400 bc0 300 bc1 200</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-rsvp-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-rsvp)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>ds-te mode ietf</td>
<td>Enables IETF DS-TE mode and default TE class map. Configure IETF DS-TE mode on all nodes in the network.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# ds-te mode ietf</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ds-te bc-model mam</td>
<td>Enables the MAM bandwidth constraint model globally.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# ds-te bc-model mam</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td>Exits the current configuration mode.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>signalled-bandwidth {bandwidth [class-type ct]</td>
<td>sub-pool bandwidth}</td>
<td>Configures the bandwidth required for an MPLS TE tunnel. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-rsvp-if)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>signalled-bandwidth 10 class-type 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Configuring Traffic Engineering Tunnel Bandwidth, on page 70
- Maximum Allocation Bandwidth Constraint Model, on page 116

**Configuring MPLS -TE and Fast-Reroute on OSPF**

Perform this task to configure MPLS-TE and Fast Reroute (FRR) on OSPF.
### Before You Begin

Only point-to-point (P2P) interfaces are supported for OSPF multiple adjacencies. These may be either native P2P interfaces or broadcast interfaces on which the OSPF P2P configuration command is applied to force them to behave as P2P interfaces as far as OSPF is concerned. This restriction does not apply to IS-IS.

The tunnel-te interface is not supported under IS-IS.

### SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. path-option [protecting] preference-priority {dynamic [pce [address ipv4 address]] | explicit {name pathname | identifier path-number } } [isis instance name {level level} ] [ospf instance name {area area ID} ] [verbatim] [lockdown]
4. Repeat Step 3 as many times as needed.
5. commit
6. show mpls traffic-eng tunnels [tunnel-number]

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface tunnel-te tunnel-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te 1 RP/0/RP0/CPU0:router(config-if)#</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>path-option [protecting] preference-priority {dynamic [pce [address ipv4 address]]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit identifier 6 ospf green area 0</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Repeat Step 3 as many times as needed.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-if)# path-option 2 explicit</td>
</tr>
</tbody>
</table>
### Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE

Perform this task to configure an overload node avoidance in MPLS-TE. When the overload bit is enabled, tunnels are brought down when the overload node is found in the tunnel path.

#### SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng`
3. `path-selection ignore overload {head | mid | tail}`
4. `commit`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>mpls traffic-eng</code></td>
</tr>
</tbody>
</table>
| **Example:**      | `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`  
|                   | `RP/0/RP0/CPU0:router(config-mpls-te)#` |
| **Step 3**        | `path-selection ignore overload {head | mid | tail}` |
| **Example:**      | `RP/0/RP0/CPU0:router(config-mpls-te)#`  
<p>|                   | <code>path-selection ignore overload head</code> |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td>commit</td>
</tr>
</tbody>
</table>

Related Topics
- Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE, on page 120
- Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 260

Configuring GMPLS

To fully configure GMPLS, you must complete these high-level tasks in order:

- Configuring IPCC Control Channel Information, on page 174
- Configuring Local and Remote TE Links, on page 178
- Configuring Numbered and Unnumbered Optical TE Tunnels, on page 188
- Configuring LSP Hierarchy, on page 192
- Configuring Border Control Model, on page 193
- Configuring Path Protection, on page 193

Note
These high-level tasks are broken down into, in some cases, several subtasks.

Configuring IPCC Control Channel Information

To configure IPCC control channel information, complete these subtasks:

- Configuring Router IDs, on page 174
- Configuring OSPF over IPCC, on page 176

Note
You must configure each subtask on both the headend and tailend router.

Configuring Router IDs

Perform this task to configure the router ID for the headend and tailend routers.
**SUMMARY STEPS**

1. configure  
2. interface type interface-path-id  
3. ipv4 address ipv4-address mask  
4. exit  
5. router ospf process-name  
6. mpls traffic-eng router-id type interface-path-id  
7. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface type interface-path-id</td>
<td>Specifies a primary or secondary IPv4 address for an interface.</td>
</tr>
</tbody>
</table>
| **Example:** RP/0/RP0/CPU0:router(config)# interface POS0/6/0/0 | Network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.  
- Network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash. |
| **Step 3** ipv4 address ipv4-address mask  | Exits the current configuration mode. |
| **Example:** RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0 | Configures an Open Shortest Path First (OSPF) routing process. The process name is any alphanumeric string no longer than 40 characters without spaces. |
| **Step 4** exit  | Configures an Open Shortest Path First (OSPF) routing process. The process name is any alphanumeric string no longer than 40 characters without spaces. |
| **Example:** RP/0/RP0/CPU0:router(config-if)# exit RP/0/RP0/CPU0:router(config)# | |
| **Step 5** router ospf process-name  | |
| **Example:** RP/0/RP0/CPU0:router(config)# router ospf 1 RP/0/RP0/CPU0:router(config-ospf)# | |
### Purpose

**Step 6**  
**Command or Action:**  
```  
mpls traffic-eng router-id type interface-path-id  
```

**Purpose:** Specifies that the TE router identifier for the node is the IP address that is associated with a given interface. The router ID is specified with an interface name or an IP address. By default, MPLS uses the global router ID.

**Example:**
```
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router id Loopback0  
```

**Step 7**  
**Command or Action:** commit

---

### Related Topics

**GMPLS Support,** on page 122

### Configuring OSPF over IPCC

Perform this task to configure OSPF over IPCC on both the headend and tailend routers. The IGP interface ID is configured for control network, specifically for the signaling plane in the optical domain.

#### Note

IPCC support is restricted to routed, out-of-fiber, and out-of-band.

### SUMMARY STEPS

1. configure  
2. router ospf process-name  
3. area area-id  
4. interface type interface-path-id  
5. exit  
6. exit  
7. mpls traffic-eng router-id \{type interface-path-id \ ip-address \}  
8. area area-id  
9. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>router ospf process-name</strong>&lt;br&gt;Example: RP/0/RP0/CPU0:router(config)# router ospf 1</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>area area-id</strong>&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-ospf)# area 0</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>interface type interface-path-id</strong>&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>exit</strong>&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>exit</strong>&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-ospf-ar)# exit</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>**mpls traffic-eng router-id {type interface-path-id</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>area area-id</strong>&lt;br&gt;Example: RP/0/RP0/CPU0:router(config-ospf)# area 0</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>commit</strong></td>
</tr>
</tbody>
</table>
Configuring Local and Remote TE Links

These subtasks describe how to configure local and remote MPLS-TE link parameters for numbered and unnumbered TE links on both headend and tailend routers.

- Configuring Numbered and Unnumbered Links, on page 178
- Configuring Local Reservable Bandwidth, on page 179
- Configuring Local Switching Capability Descriptors, on page 180
- Configuring Persistent Interface Index, on page 181
- Enabling LMP Message Exchange, on page 182
- Disabling LMP Message Exchange, on page 183
- Configuring Remote TE Link Adjacency Information for Numbered Links, on page 184
- Configuring Remote TE Link Adjacency Information for Unnumbered Links, on page 186

Configuring Numbered and Unnumbered Links

Perform this task to configure numbered and unnumbered links.

Note

Unnumbered TE links use the IP address of the associated interface.

SUMMARY STEPS

1. configure
2. interface type interface-path-id
3. Do one of the following:
   - ipv4 address ipv4-address mask
   - ipv4 unnumbered interface type interface-path-id
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
</tbody>
</table>
## Implementing MPLS Traffic Engineering

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>interface</code> <code>type</code> <code>interface-path-id</code></td>
<td>Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
</tbody>
</table>

### Example:

```
RP/0/RP0/CPU0:router(config)# interface POS0/6/0/0
```

### Step 3

Do one of the following:

- **ipv4 address** `ipv4-address mask`
- **ipv4 unnumbered interface** `type` `interface-path-id`

### Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0
```

### Purpose

- Specifies a primary or secondary IPv4 address for an interface.
- Network mask is a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.
- Network mask is indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash.

### Note

- Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.

If you configured a unnumbered GigabitEthernet interface in Step 2 and selected the `ipv4 unnumbered` interface command type option in this step, you must enter the `ipv4 point-to-point` command to configure point-to-point interface mode.

### Step 4

**commit**

### Configuring Local Reservable Bandwidth

Perform this task to configure the local reservable bandwidth for the data bearer channels.

### SUMMARY STEPS

1. `configure`
2. `rsvp interface` `type` `interface-path-id`
3. `bandwidth` `[total reservable bandwidth] [bc0 bandwidth] [global-pool bandwidth] [sub-pool reservable-bw]`
4. `commit`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> rsvp interface type interface-path-id</td>
<td>Enters RSVP configuration mode and selects an RSVP interface ID.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# rsvp interface POS0/6/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bandwidth [total reservable bandwidth] [bc0 bandwidth] [global-pool bandwidth] [sub-pool reservable-bw]</td>
<td>Sets the reserved RSVP bandwidth available on this interface.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-rsvp-if)# bandwidth 2488320 2488320</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Local Switching Capability Descriptors

Perform this task to configure the local switching capability descriptor.

### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. flooding-igp ospf instance-id area area-id
5. switching key value [encoding encoding type]
6. switching key value [capability {psc1 | lsc | fsc}]
7. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 3** interface type interface-path-id | Enters MPLS-TE interface configuration mode and enables traffic engineering on a particular interface on the originating node. |
| **Example:** RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0 | |

| **Step 4** flooding-igp ospf instance-id area area-id | Specifies the IGP OSPF interface ID and area where the TE links are to be flooded. |
| **Example:** RP/0/RP0/CPU0:router(config-mpls-te-if)# flooding-igp ospf 0 area 1 | |

| **Step 5** switching key value [encoding encoding type] | Specifies the switching configuration for the interface and enters switching key mode where you will configure encoding and capability. **Note** The recommended switch key value is 0. |
| **Example:** RP/0/RP0/CPU0:router(config-mpls-te-if)# switching key 1 encoding ethernet | |

| **Step 6** switching key value [capability {psc1 | lsc | fsc}] | Specifies the interface switching capability type. The recommended switch capability type is psc1. |
| **Example:** RP/0/RP0/CPU0:router(config-mpls-te-if)# switching key 1 capability psc1 | |

| **Step 7** commit | |

---

### Configuring Persistent Interface Index

Perform this task to preserve the LMP interface index across all interfaces on the router.

### SUMMARY STEPS

1. configure
2. snmp-server ifindex persist
3. commit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>snmp-server ifindex persist</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>snmp-server ifindex persist</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

### Enabling LMP Message Exchange

Perform the following task to enable LMP message exchange. LMP is enabled by default. You can disable LMP on a per neighbor basis using the `lmp static` command in LMP protocol neighbor mode.

**Note**

LMP is recommended unless the peer optical device does not support LMP (in which case it is necessary to disable it at both ends).

### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. lmp neighbor name
4. ipcc routed
5. remote node-id node-id
6. commit

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>
### Configuring GMPLS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><strong>lmp neighbor name</strong></td>
<td>Configures or updates a LMP neighbor and its associated parameters.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# lmp neighbor OXC1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ipcc routed</strong></td>
<td>Configures a routable Internet Protocol Control Channel (IPCC).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# ipcc routed</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
</tbody>
</table>
| **remote node-id node-id** | Configures the remote node ID for an LMP neighbor.  
In addition, the node-id value can be an IPv4 address. |
| Example:               |                                                                         |
| RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# remote node-id 2.2.2.2 |                                                                         |
| **Step 6**              |                                                                          |
| **commit**              |                                                                         |

**Disabling LMP Message Exchange**

Perform the following task to disable LMP message exchange. LMP is enabled by default. You can disable LMP on a per neighbor basis using the **lmp static** command in LMP protocol neighbor mode.

**Note**

LMP is recommended unless the peer optical device does not support LMP (in which case it is necessary to disable it at both ends).

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. lmp neighbor name
4. lmp static
5. ipcc routed
6. remote node-id node-id
7. commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
<td></td>
</tr>
</tbody>
</table>
| **Step 2** | `mpls traffic-eng` | Enters MPLS-TE configuration mode.  
**Example:**  
RP/0/RP0/CPU0:router(config)# mpls traffic-eng |
| **Step 3** | `lmp neighbor name` | Configures or updates a LMP neighbor and its associated parameters.  
**Example:**  
RP/0/RP0/CPU0:router(config-mpls-te)# lmp neighbor OXC1 |
| **Step 4** | `lmp static` | Disables dynamic LMP procedures for the specified neighbor, including LMP hello and LMP link summary. This command is used for neighbors that do not support dynamic LMP procedures.  
**Example:**  
RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# lmp static |
| **Step 5** | `ipcc routed` | Configures a routable IPCC.  
**Example:**  
RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# ipcc routed |
| **Step 6** | `remote node-id node-id` | Configures the remote node ID for an LMP neighbor. The node ID value must be an IPv4 address.  
**Example:**  
RP/0/RP0/CPU0:router(config-mpls-te-nbr-OXC1)# remote node-id 2.2.2.2 |
| **Step 7** | `commit` | |
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. lmp data-link adjacency
5. remote switching-capability {fsc | lsc | psc1}
6. remote interface-id unnum value
7. remote node-id node-id
8. neighbor name
9. remote node-id address
10. commit
11. show mpls lmp

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enters MPLS-TE interface configuration mode and enables TE on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3 interface type interface-path-id</td>
<td>Configures LMP neighbor remote TE links.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4 lmp data-link adjacency</td>
<td>Configures the remote LMP MPLS-TE interface switching capability.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# lmp data-link adjacency</td>
<td></td>
</tr>
<tr>
<td>Step 5 remote switching-capability {fsc</td>
<td>lsc</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote switching-capability lsc</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Configures the unnumbered interface identifier. Identifiers, which you specify by using this command, are the values assigned by the neighbor at the remote side.</td>
</tr>
<tr>
<td>remote interface-id unnum value</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote interface-id unnum 7</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Configures the remote node ID.</td>
</tr>
<tr>
<td>remote node-id node-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote node-id 10.10.10.10</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Configures or updates an LMP neighbor and its associated parameters.</td>
</tr>
<tr>
<td>neighbor name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# neighbor OXC1</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Configures the remote node ID.</td>
</tr>
<tr>
<td>remote node-id address</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote node-id 10.10.10.10</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>commit</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Verifies the assigned value for the local interface identifiers.</td>
</tr>
<tr>
<td>show mpls lmp</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router# show mpls lmp</td>
</tr>
</tbody>
</table>

### Configuring Remote TE Link Adjacency Information for Unnumbered Links

Perform this task to configure remote TE link adjacency information for unnumbered links.

**Note**

To display the assigned value for the local interface identifiers, use the `show mpls lmp` command.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. lmp data link adjacency
5. neighbor name
6. remote te-link-id unnum
7. remote interface-id unnum interface-identifier
8. remote switching-capability {fsc | lsc | psc1}
9. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
<td>Enters MPLS-TE interface configuration mode and enables TE on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0</td>
</tr>
<tr>
<td>Step 4</td>
<td>lmp data link adjacency</td>
<td>Configures LMP neighbor remote TE links.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if)# lmp data-link adjacency</td>
</tr>
<tr>
<td>Step 5</td>
<td>neighbor name</td>
<td>Configures or updates a LMP neighbor and its associated parameters.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# neighbor OXC1</td>
</tr>
<tr>
<td>Step 6</td>
<td>remote te-link-id unnum</td>
<td>Configures the unnumbered interface and identifier.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><code>te-link-id unnum 111</code></td>
<td>Configure the unnumbered interface identifier.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 7**

**Example:**

```
RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote interface-id unnum 7
```

Configures the unnumbered interface identifier. Identifiers, which you specify by using this command, are the values assigned by the neighbor at the remote side.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`remote switching-capability {fsc</td>
<td>lsc</td>
</tr>
</tbody>
</table>

**Step 8**

**Example:**

```
RP/0/RP0/CPU0:router(config-mpls-te-if-adj)# remote switching-capability lsc
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>commit</code></td>
<td></td>
</tr>
</tbody>
</table>

## Configuring Numbered and Unnumbered Optical TE Tunnels

These subtasks are included:

- Configuring an Optical TE Tunnel Using Dynamic Path Option, on page 188
- Configuring an Optical TE Tunnel Using Explicit Path Option, on page 191

**Note**

Before you can successfully bring optical TE tunnels “up,” you must complete the procedures in the preceding sections.

The following characteristics can apply to the headend (or, signaling) router:

- Tunnels can be numbered or unnumbered.
- Tunnels can be dynamic or explicit.

The following characteristics can apply to the tailend (or, passive) router:

- Tunnels can be numbered or unnumbered.
- Tunnels must use the explicit path-option.

## Configuring an Optical TE Tunnel Using Dynamic Path Option

Perform this task to configure a numbered or unnumbered optical tunnel on a router; in this example, the dynamic path option on the headend router. The dynamic option does not require that you specify the different hops to be taken along the way. The hops are calculated automatically.
The examples describe how to configure optical tunnels. It does not include procedures for every option available on the headend and tailend routers.

### SUMMARY STEPS

1. configure
2. interface tunnel-gte tunnel-id
3. ipv4 address ip-address/prefix or ipv4 unnumbered type interface-path-id
4. switching transit switching type encoding encoding type
5. priority setup-priority hold-priority
6. signalled-bandwidth bandwidth [class-type ct] | sub-pool bandwidth
7. destination ip-address
8. path-option path-id dynamic
9. direction [bidirectional]
10. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Configures an MPLS-TE tunnel for GMPLS interfaces.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-gte tunnel-id</td>
<td>Specifies a primary or secondary IPv4 address for an interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-gte1</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>ipv4 address ip-address/prefix or ipv4 unnumbered type interface-path-id</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0</td>
<td>Specifies a primary or secondary IPv4 address for an interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>4</td>
<td><strong>switching transit</strong> <strong>switching type encoding</strong> encoding type</td>
<td>Specifies the switching capability and encoding types for all transit TE links used to signal the optical tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# switching transit lsc encoding sonetsdh</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>priority setup-priority</strong> hold-priority</td>
<td>Configures setup and reservation priorities for MPLS-TE tunnels.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# priority 1 1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>signalled-bandwidth</strong> {bandwidth [class-type ct]</td>
<td>Sets the CT0 bandwidth required on this interface. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 1, priority 7).</td>
</tr>
<tr>
<td></td>
<td>sub-pool bandwidth}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10 class-type 1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>destination ip-address</strong></td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Destination address is the remote node’s MPLS-TE router ID.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Destination address is the merge point between backup and protected tunnels.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><strong>path-option path-id</strong> dynamic</td>
<td>Configures the dynamic path option and path ID.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><strong>direction</strong> [bidirectional]</td>
<td>Configures a bidirectional optical tunnel for GMPLS.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# direction bidirection</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>commit</strong></td>
<td></td>
</tr>
</tbody>
</table>
Configuring an Optical TE Tunnel Using Explicit Path Option

Perform this task to configure a numbered or unnumbered optical TE tunnel on a router. This task can be applied to both the headend and tailend router.

Note
You cannot configure dynamic tunnels on the tailend router.

SUMMARY STEPS

1. configure
2. interface tunnel-gte tunnel-id
3. ipv4 address ipv4-address mask or ipv4 unnumbered type interface-path-id
4. passive
5. match identifier tunnel number
6. destination ip-address
7. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Configures an MPLS-TE tunnel interface for GMPLS interfaces.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-gte tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface for GMPLS interfaces.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>interface tunnel-gte 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>ipv4 address ipv4-address mask or ipv4 unnumbered type interface-path-id</td>
<td>Specifies a primary or secondary IPv4 address for an interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ipv4 address 127.0.0.1 255.0.0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>enables IPv4 processing on a point-to-point interface</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>without assigning an explicit IPv4 address to that</td>
<td></td>
</tr>
<tr>
<td></td>
<td>interface.</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>passive</td>
<td>Configures a passive interface.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# passive</td>
<td>The tailend (passive) router does not signal the tunnel, it simply accepts a connection from the headend router. The tailend router supports the same configuration as the headend router.</td>
</tr>
</tbody>
</table>

### Step 5

**match identifier tunnel number**

<table>
<thead>
<tr>
<th>Example:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# match identifier gmplsl_t1</td>
<td>Configures the match identifier. You must enter the hostname for the head router then underscore <code>_</code>, and the tunnel number for the head router. If tunnel-te1 is configured on the head router with a hostname of gmplsl1, CLI is match identifier gmplsl_t1.</td>
</tr>
</tbody>
</table>

### Step 6

**destination ip-address**

<table>
<thead>
<tr>
<th>Example:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router(config-if)# destination 10.1.1.1</td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
</tbody>
</table>

### Step 7

**commit**

### Configuring LSP Hierarchy

These tasks describe the high-level steps that are required to configure LSP hierarchy.

LSP hierarchy allows standard MPLS-TE tunnels to be established over GMPLS-TE tunnels.

Consider the following information when configuring LSP hierarchy:

- LSP hierarchy supports numbered optical TE tunnels with IPv4 addresses only.
- LSP hierarchy supports numbered optical TE tunnels using numbered or unnumbered TE links.

### Note

Before you can successfully configure LSP hierarchy, you must first establish a numbered optical tunnel between the headend and tailend routers.

To configure LSP hierarchy, you must perform a series of tasks that have been previously described in this GMPLS configuration section. The tasks, which must be completed in the order presented, are as follows:

1. Establish an optical TE tunnel.
2. Configure an optical TE tunnel under IGP.
3. Configure the bandwidth on the optical TE tunnel.
4. Configure the optical TE tunnel as a TE link.
5 Configure an MPLS-TE tunnel.

Related Topics

Configuring Numbered and Unnumbered Optical TE Tunnels, on page 188

Configuring Border Control Model

Border control model lets you specify the optical core tunnels to be advertised to edge packet topologies. Using this model, the entire topology is stored in a separate packet instance, allowing packet networks where these optical tunnels are advertised to use LSP hierarchy to signal an MPLS tunnel over the optical tunnel.

Consider the following information when configuring protection and restoration:

- GMPLS optical TE tunnel must be numbered and have a valid IPv4 address.
- Router ID, which is used for the IGP area and interface ID, must be consistent in all areas.
- OSPF interface ID may be a numeric or alphanumeric.

Note

Border control model functionality is provided for multiple IGP instances in one area or in multiple IGP areas.

To configure border control model functionality, you will perform a series of tasks that have been previously described in this GMPLS configuration section. The tasks, which must be completed in the order presented, are as follows:

1 Configure two optical tunnels on different interfaces.

Note

When configuring IGP, you must keep the optical and packet topology information in separate routing tables.

2 Configure OSPF adjacency on each tunnel.

3 Configure bandwidth on each tunnel.

4 Configure packet tunnels.

Configuring Path Protection

These tasks describe how to configure path protection:

- Configuring an LSP, on page 194
- Forcing Reversion of the LSP, on page 196
Configuring an LSP

Perform this task to configure an LSP for an explicit path. Path protection is enabled on a tunnel by adding an additional path option configuration at the active end. The path can be configured either explicitly or dynamically.

Note
When the dynamic option is used for both working and protecting LSPs, CSPF extensions are used to determine paths with different degrees of diversity. When the paths are computed, they are used over the lifetime of the LSPs. The nodes on the path of the LSP determine if the PSR is or is not for a given LSP. This determination is based on information that is obtained at signaling.

SUMMARY STEPS

1. configure
2. interface tunnel-gte number
3. ipv4 address ipv4-address mask or ipv4 unnumbered type interface-path-id
4. signalled-name name
5. switching transit capability-switching-type encoding encoding-type
6. switching endpoint capability-switching-type encoding encoding-type
7. priority setup-priority hold-priority
8. signalled-bandwidth [bandwidth [class-type ct] | sub-pool bandwidth]
9. destination ip-address
10. path-option path-id explicit {name pathname | path-number}
11. path-option protecting path-id explicit {name pathname | path-number}
12. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-gte number</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-gte 1</td>
</tr>
<tr>
<td>Step 3</td>
<td>ipv4 address ipv4-address mask or ipv4 unnumbered type interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 address 99.99.99.2 255.255.255.254</td>
</tr>
<tr>
<td></td>
<td>Specifies a primary or secondary IPv4 address for an interface.</td>
</tr>
<tr>
<td></td>
<td>• Network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>• Network mask can be indicated as a slash (/) and a number (prefix length). The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash must precede the decimal value, and there is no space between the IP address and the slash. or • Enables IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 4**  
**signalled-name name**  
*Example:*  
RP/0/RP0/CPU0:router(config-if)#  
signalled-name tunnel-gte1

**Step 5**  
**switching transit capability-switching-type encoding encoding-type**  
*Example:*  
RP/0/RP0/CPU0:router(config-if)# switching transit lsc encoding sonetsdh

**Step 6**  
**switching endpoint capability-switching-type encoding encoding-type**  
*Example:*  
RP/0/RP0/CPU0:router(config-if)# switching endpoint psc1 encoding sonetsdh

**Step 7**  
**priority setup-priority hold-priority**  
*Example:*  
RP/0/RP0/CPU0:router(config-if)# priority 2 2

**Step 8**  
**signalled-bandwidth {bandwidth [class-type ct] | sub-pool bandwidth}**  
*Example:*  
RP/0/RP0/CPU0:router(config-if)# signalling-bandwidth 2488320

**Step 9**  
**destination ip-address**  
*Example:*  
Assigns a destination address on the new tunnel.
### Purpose

Command or Action

Example:

```plaintext
RP/0/RP0/CPU0:router(config-if)# destination 24.24.24.24
```

- Destination address is the remote node’s MPLS-TE router ID.
- Destination address is the merge point between backup and protected tunnels.

### Purpose

**Step 10**

`path-option path-id explicit {name pathname | path-number}`

Example:

```plaintext
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name po4
```

**Step 11**

`path-option protecting path-id explicit {name pathname | path-number}`

Example:

```plaintext
RP/0/RP0/CPU0:router(config-if)# path-option protecting 1 explicit name po6
```

**Step 12**

`commit`

### Forcing Reversion of the LSP

Perform this task to allow a forced reversion of the LSPs, which is only applicable to 1:1 LSP protection.

### SUMMARY STEPS

1. `mpls traffic-eng path-protection switchover {gmpls tunnel-name | tunnel-te tunnel-id}
2. `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**

`mpls traffic-eng path-protection switchover {gmpls tunnel-name | tunnel-te tunnel-id}

Example:

```plaintext
RP/0/RP0/CPU0:router# mpls traffic-eng path-protection switchover tunnel-te 1
```

Specifies a manual switchover for path protection for a GMPLS optical LSP. The tunnel ID is configured for a switchover.

The `mpls traffic-eng path-protection switchover` command must be issued on both head and tail router of the GMPLS LSP to achieve the complete path switchover at both ends.

<table>
<thead>
<tr>
<th>Step 2</th>
<th>commit</th>
</tr>
</thead>
</table>

Configuring Flexible Name-based Tunnel Constraints

To fully configure MPLS-TE flexible name-based tunnel constraints, you must complete these high-level tasks in order:

1. Assigning Color Names to Numeric Values, on page 197
2. Associating Affinity-Names with TE Links, on page 198
3. Associating Affinity Constraints for TE Tunnels, on page 199

Assigning Color Names to Numeric Values

The first task in enabling the new coloring scheme is to assign a numerical value (in hexadecimal) to each value (color).

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enter MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
</tbody>
</table>

Step 3 affinity-map affinity name {affinity value | bit-position value} Enables an affinity name and a map value by using a color name (repeat this command to assign multiple colors up to a maximum of 64 colors). An affinity color name cannot exceed 64 characters. An affinity value cannot exceed a single digit. For example, magenta1.
## Configuring Flexible Name-based Tunnel Constraints

### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. attribute-names attribute name
5. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface type interface-path-id</td>
</tr>
</tbody>
</table>

### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>exceed 64 characters. The value you assign to a color name must be a single digit.</td>
</tr>
</tbody>
</table>

### Related Topics

- Flexible Name-based Tunnel Constraints, on page 123
- Configure Flexible Name-based Tunnel Constraints: Example, on page 262

### Associating Affinity-Names with TE Links

The next step in the configuration of MPLS-TE Flexible Name-based Tunnel Constraints is to assign affinity names and values to TE links. You can assign up to a maximum of 32 colors. Before you assign a color to a link, you must define the name-to-value mapping for each color.
Purpose | Command or Action | | --- | --- | --- | --- |
Assigns colors to TE links over the selected interface. | RP/0/RP0/CPU0:router(config-mpls-te-if)# attribute-names | | | Example: | RP/0/RP0/CPU0:router(config-mpls-te-if)# attribute-names red | | | | Step 4 | | | | | Step 5 | commit | | Related Topics | Flexible Name-based Tunnel Constraints, on page 123 | Configure Flexible Name-based Tunnel Constraints: Example, on page 262 | Assigning Color Names to Numeric Values, on page 197 | 

**Associating Affinity Constraints for TE Tunnels**

The final step in the configuration of MPLS-TE Flexible Name-based Tunnel Constraints requires that you associate a tunnel with affinity constraints.

Using this model, there are no masks. Instead, there is support for four types of affinity constraints:

- include
- include-strict
- exclude
- exclude-all

---

**Note**

For the affinity constraints above, all but the exclude-all constraint may be associated with up to 10 colors.

**SUMMARY STEPS**

1. configure
2. interface tunnel-te tunnel-id
3. affinity {affinity-value mask mask-value | exclude name | exclude -all | include name | include-strict name}
4. commit
### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-te <em>tunnel-id</em></td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config)# interface tunnel-te 1</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><em>affinity</em> {<em>affinity-value</em> *mask* *mask-value* | *exclude* *name* | *exclude-all* | *include* *name* | *include-strict* *name*}</td>
<td>Configures link attributes for links comprising a tunnel. You can have up to ten colors. Multiple include statements can be specified under tunnel configuration. With this configuration, a link is eligible for CSPF if it has at least a red color or has at least a green color. Thus, a link with red and any other colors as well as a link with green and any additional colors meet the above constraint.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config-if)# affinity include red</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**
- Flexible Name-based Tunnel Constraints, on page 123
- Configure Flexible Name-based Tunnel Constraints: Example, on page 262

### Configuring IS-IS to Flood MPLS-TE Link Information

Perform this task to configure a router running the Intermediate System-to-Intermediate System (IS-IS) protocol to flood MPLS-TE link information into multiple IS-IS levels.
This procedure shows how to enable MPLS-TE in both IS-IS Level 1 and Level 2.

### Summary Steps

1. `configure`
2. `router isis instance-id`
3. `net network-entity-title`
4. `address-family {ipv4 | ipv6} {unicast}`
5. `metric-style wide`
6. `mpls traffic-eng level`
7. `commit`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>router isis instance-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# router isis 1</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>net network-entity-title</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-isis)# net 47.0001.0000.0000.0002.00</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>address-family {ipv4</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>metric-style wide</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-isis-af)# metric-style wide</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>mpls traffic-eng level</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-isis-af)# mpls traffic-eng level-1-2</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

### Configuring an OSPF Area of MPLS-TE

Perform this task to configure an OSPF area for MPLS-TE in both the OSPF backbone area 0 and area 1.
### SUMMARY STEPS

1. configure  
2. router ospf **process-name**  
3. mpls traffic-eng router-id **ip-address**  
4. area **area-id**  
5. interface **type interface-path-id**  
6. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>router ospf <strong>process-name</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config)# router ospf 100</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>mpls traffic-eng router-id <strong>ip-address</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>area <strong>area-id</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-ospf)# area 0</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>interface <strong>type interface-path-id</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-ospf-ar)# interface POS 0/2/0/0</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

---

**Configuring Explicit Paths with ABRs Configured as Loose Addresses**

Perform this task to specify an IPv4 explicit path with ABRs configured as loose addresses.
SUMMARY STEPS

1. configure
2. explicit-path name name
3. index index-id next-address [loose] ipv4 unicast ip-address
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 explicit-path name name</td>
<td>Enters a name for the explicit path.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# explicit-path name interarea1</td>
<td></td>
</tr>
<tr>
<td>Step 3 index index-id next-address [loose] ipv4 unicast ip-address</td>
<td>Includes an address in an IP explicit path of a tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-expl-path)# index 1 next-address loose ipv4 unicast 10.10.10.10</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring MPLS-TE Forwarding Adjacency

Perform this task to configure forwarding adjacency on a specific tunnel-te interface.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. forwarding-adjacency holdtime value
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
</tbody>
</table>
## Purpose

**Command or Action**

**Purpose**

<table>
<thead>
<tr>
<th>Step 2</th>
<th>interface tunnel-te tunnel-id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enters MPLS-TE interface configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```bash
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
```

<table>
<thead>
<tr>
<th>Step 3</th>
<th>forwarding-adjacency holdtime value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configures forwarding adjacency using an optional specific holdtime value. By default, this value is 0 (milliseconds).</td>
</tr>
</tbody>
</table>

**Example:**

```bash
RP/0/RP0/CPU0:router(config-if)# forwarding-adjacency holdtime 60
```

<table>
<thead>
<tr>
<th>Step 4</th>
<th>commit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics

- MPLS-TE Forwarding Adjacency Benefits, on page 127
- Configure Forwarding Adjacency: Example, on page 264

## Configuring Unequal Load Balancing

Perform these tasks to configure unequal load balancing:

- Setting Unequal Load Balancing Parameters, on page 204
- Enabling Unequal Load Balancing, on page 205

### Setting Unequal Load Balancing Parameters

The first step you must take to configure unequal load balancing requires that you set the parameters on each specific interface. The default load share for tunnels with no explicit configuration is the configured bandwidth.

**Note**

Equal load-sharing occurs if there is no configured bandwidth.

### SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. load-share value
4. commit
5. show mpls traffic-eng tunnels
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
</tbody>
</table>
| **Step 2**        | interface tunnel-te `tunnel-id`  
  Example:  
  `RP/0/RP0/CPU0:router(config)# interface tunnel-te 1`  
  Configures an MPLS-TE tunnel interface configuration mode and enables traffic engineering on a particular interface on the originating node.  
  **Note** Only tunnel-te interfaces are permitted. |
| **Step 3**        | load-share `value`  
  Example:  
  `RP/0/RP0/CPU0:router(config-if)# load-share 1000`  
  Configures the load-sharing parameters for the specified interface. |
| **Step 4**        | commit |
| **Step 5**        | show mpls traffic-eng tunnels  
  Example:  
  `RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels`  
  Verifies the state of unequal load balancing, including bandwidth and load-share values. |

**Related Topics**

Unequal Load Balancing, on page 128  
Configure Unequal Load Balancing: Example, on page 265

**Enabling Unequal Load Balancing**

This task describes how to enable unequal load balancing. (For example, this is a global switch used to turn unequal load-balancing on or off.)

**SUMMARY STEPS**

1. configure  
2. mpls traffic-eng  
3. load-share unequal  
4. commit  
5. show mpls traffic-eng tunnels
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>mpls traffic-eng</td>
<td>Enters the MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>load-share unequal</td>
<td>Enables unequal load sharing across TE tunnels to the same destination.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# load-share unequal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>show mpls traffic-eng tunnels</td>
<td>Verifies the state of unequal load balancing, including bandwidth and load-share values.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Unequal Load Balancing, on page 128
- Configure Unequal Load Balancing: Example, on page 265

**Configuring a Path Computation Client and Element**

Perform these tasks to configure Path Computation Client (PCC) and Path Computation Element (PCE):

- Configuring a Path Computation Client, on page 206
- Configuring a Path Computation Element Address, on page 207
- Configuring PCE Parameters, on page 208

**Configuring a Path Computation Client**

Perform this task to configure a TE tunnel as a PCC.
Only one TE-enabled IGP instance can be used at a time.

**SUMMARY STEPS**

1. configure
2. interface tunnel-te tunnel-id
3. path-option preference-priority dynamic pce
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>ENTERS MPLS-TE INTERFACE CONFIGURATION MODE AND ENABLES TRAFFIC ENGINEERING ON A PARTICULAR INTERFACE ON THE ORIGINATING NODE.</td>
</tr>
<tr>
<td>2</td>
<td>interface tunnel-te tunnel-id</td>
<td>CONFIGURES A TE TUNNEL AS A PCC.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te 6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>path-option preference-priority dynamic pce</td>
<td>CONFIGURES A TE TUNNEL AS A PCC.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic pce</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Path Computation Element, on page 128
- Configure PCE: Example, on page 266

**Configuring a Path Computation Element Address**

Perform this task to configure a PCE address.

**Note**

Only one TE-enabled IGP instance can be used at a time.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. pce address ipv4 *address*
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>pce address ipv4 <em>address</em></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce address ipv4 10.1.1.1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

Related Topics

- Path Computation Element, on page 128
- Configure PCE: Example, on page 266

Configuring PCE Parameters

Perform this task to configure PCE parameters, including a static PCE peer, periodic reoptimization timer values, and request timeout values.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. pce address ipv4 *address*
4. pce peer ipv4 *address*
5. pce keepalive *interval*
6. pce deadtimer *value*
7. pce reoptimize *value*
8. pce request-timeout *value*
9. pce tolerance keepalive *value*
10. commit
11. show mpls traffic-eng pce peer [address | all]
12. show mpls traffic-eng pce tunnels

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>pce address ipv4 <em>address</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Configures a PCE IPv4 address.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce address ipv4 10.1.1.1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>pce peer ipv4 <em>address</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Configures a static PCE peer address. PCE peers are also discovered dynamically through OSPF or ISIS.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce peer address ipv4 10.1.1.1</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>pce keepalive <em>interval</em></td>
</tr>
<tr>
<td>Example:</td>
<td>Configures a PCEP keepalive interval. The range is from 0 to 255 seconds. When the keepalive interval is 0, the LSR does not send keepalive messages.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce keepalive 10</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong> pce deadtimer value</td>
<td>Configures a PCE deadtimer value. The range is from 0 to 255 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce deadtimer 50</td>
</tr>
<tr>
<td><strong>Step 7</strong> pce reoptimize value</td>
<td>Configures a periodic reoptimization timer value. The range is from 60 to 604800 seconds. When the dead interval is 0, the LSR does not timeout a PCEP session to a remote peer.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce reoptimize 200</td>
</tr>
<tr>
<td><strong>Step 8</strong> pce request-timeout value</td>
<td>Configures a PCE request-timeout. Range is from 5 to 100 seconds. PCC or PCE keeps a pending path request only for the request-timeout period.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce request-timeout 10</td>
</tr>
<tr>
<td><strong>Step 9</strong> pce tolerance keepalive value</td>
<td>Configures a PCE tolerance keepalive value (which is the minimum acceptable peer proposed keepalive).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# pce tolerance keepalive 10</td>
</tr>
<tr>
<td><strong>Step 10</strong> commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> show mpls traffic-eng pce peer [address</td>
<td>all]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng pce peer</td>
</tr>
<tr>
<td><strong>Step 12</strong> show mpls traffic-eng pce tunnels</td>
<td>Displays the status of the PCE tunnels.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RP0/CPU0:router# show mpls traffic-eng pce tunnels</td>
</tr>
</tbody>
</table>

**Related Topics**
- Path Computation Element, on page 128
- Configure PCE: Example, on page 266
Configuring Policy-based Tunnel Selection

Perform this task to configure policy-based tunnel selection (PBTS).

**SUMMARY STEPS**

1. configure
2. interface tunnel-te tunnel-id
3. ipv4 unnumbered type interface-path-id
4. signalled-bandwidth \( \{bandwidth [class-type cr] | \text{sub-pool bandwidth}\} \)
5. autoroute announce
6. destination ip-address
7. policy-class \( \{1 - 7\} | \{\text{default}\} \)
8. path-option preference-priority \( \{\text{explicit name explicit-path-name}\} \)
9. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Step 2 interface tunnel-te tunnel-id</td>
<td>( \text{Example:} \ RP/0/RP0/CPU0:router(config)# \text{interface tunnel-te 6} )</td>
</tr>
<tr>
<td>Step 3 ipv4 unnumbered type interface-path-id</td>
<td>( \text{Example:} \ RP/0/RP0/CPU0:router(config-if)# \text{ipv4 unnumbered Loopback0} )</td>
</tr>
<tr>
<td>Step 4 signalled-bandwidth ( {bandwidth [class-type cr]</td>
<td>\text{sub-pool bandwidth}} )</td>
</tr>
<tr>
<td>Step 5 autoroute announce</td>
<td>( \text{Example:} \ RP/0/RP0/CPU0:router(config-if)# \text{autoroute} )</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>announce</td>
<td></td>
</tr>
</tbody>
</table>

**Step 6**

**destination ip-address**

*Example:*

```
RP/0/RP0/CPU0:router(config-if)# destination 10.1.1.1
```

Assigns a destination address on the new tunnel.

- Destination address is the remote node’s MPLS-TE router ID.
- Destination address is the merge point between backup and protected tunnels.

**Step 7**

**policy-class \{1 - 7\} | {default}\**

*Example:*

```
RP/0/RP0/CPU0:router(config-if)# policy-class 1
```

Configures PBTS to direct traffic into specific TE tunnels or default class.

Multiple EXP values can be specified as part of a policy-class, separated by spaces. The EXP values configured to a TE tunnel effectively form a monolithic policy-class, which should not overlap with other policy-classes. Once an EXP value is used in a policy-class configuration, it can only be reused if the subsequent policy-class configurations containing that EXP value are identical. For example, if the configuration `policy-class 1 2 3` is applied to one or more tunnels, configurations such as `policy-class 1`, `policy-class 2 3`, or `policy-class 3 4 5` become invalid.

**Step 8**

**path-option preference-priority \{explicit name explicit-path-name\}\**

*Example:*

```
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-path
```

Sets the path option to explicit with a given name (previously configured) and assigns the path ID.

**Step 9**

**commit**

---

**Related Topics**

- Policy-Based Tunnel Selection Functions, on page 130
- Policy-Based Tunnel Selection, on page 130
- Configure Policy-based Tunnel Selection: Example, on page 267

---

**Configuring the Automatic Bandwidth**

Perform these tasks to configure the automatic bandwidth:

---

**Configuring the Collection Frequency**

Perform this task to configure the collection frequency. You can configure only one global collection frequency.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. auto-bw collect frequency minutes
4. commit
5. show mpls traffic-eng tunnels [auto-bw]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>auto-bw collect frequency minutes</td>
<td>Configures the automatic bandwidth collection frequency, and controls</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# auto-bw</td>
<td></td>
</tr>
<tr>
<td></td>
<td>collect frequency 1</td>
<td>Configures the interval between automatic bandwidth adjustments in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>minutes. Range is from 1 to 10080.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td>Displays information about MPLS-TE tunnels for the automatic bandwidth.</td>
</tr>
<tr>
<td>Step 5</td>
<td>show mpls traffic-eng tunnels [auto-bw]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router# show mpls traffic tunnels auto-bw</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

MPLS-TE Automatic Bandwidth Overview, on page 133
Configure Automatic Bandwidth: Example, on page 268
Forcing the Current Application Period to Expire Immediately

Perform this task to force the current application period to expire immediately on the specified tunnel. The highest bandwidth is applied on the tunnel before waiting for the application period to end on its own.

SUMMARY STEPS

1. `mpls traffic-eng auto-bw apply {all | tunnel-te tunnel-number}`
2. `commit`
3. `show mpls traffic-eng tunnels [auto-bw]`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure the highest bandwidth available on a tunnel without waiting for the current application period to end.</strong></td>
</tr>
<tr>
<td>`mpls traffic-eng auto-bw apply {all</td>
<td>tunnel-te tunnel-number}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>all</strong></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# mpls traffic-eng auto-bw apply tunnel-te 1</code></td>
<td>Configures the highest bandwidth available instantly on all the tunnels.</td>
</tr>
<tr>
<td><strong>tunnel-te</strong></td>
<td><strong>tunnel-te</strong></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# mpls traffic-eng auto-bw apply tunnel-te 1</code></td>
<td>Configures the highest bandwidth instantly to the specified tunnel. Range is from 0 to 65535.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>commit</strong></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>show mpls traffic-eng tunnels [auto-bw]</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>show mpls traffic-eng tunnels auto-bw</strong></td>
</tr>
<tr>
<td><code>RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-bw</code></td>
<td>Displays information about MPLS-TE tunnels for the automatic bandwidth.</td>
</tr>
</tbody>
</table>

Related Topics

Restrictions for MPLS-TE Automatic Bandwidth, on page 136

Configuring the Automatic Bandwidth Functions

Perform this task to configure the following automatic bandwidth functions:

**Application frequency**

Configures the application frequency in which a tunnel bandwidth is updated by the automatic bandwidth.
Bandwidth collection
Configures only the bandwidth collection.

Bandwidth parameters
Configures the minimum and maximum automatic bandwidth to set on a tunnel.

Adjustment threshold
Configures the adjustment threshold for each tunnel.

Overflow detection
Configures the overflow detection for each tunnel.

**SUMMARY STEPS**

1. configure
2. interface tunnel-te tunnel-id
3. auto-bw
4. application minutes
5. bw-limit {min bandwidth} {max bandwidth}
6. adjustment-threshold percentage [min minimum-bandwidth]
7. overflow threshold percentage [min bandwidth] limit limit
8. commit
9. show mpls traffic-eng tunnels [auto-bw]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel-te tunnel-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-te 6 RP/0/RP0/CPU0:router(config-if)#</td>
</tr>
<tr>
<td><strong>Step 3</strong> auto-bw</td>
<td>Configures automatic bandwidth on a tunnel interface and enters MPLS-TE automatic bandwidth interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# auto-bw RP/0/RP0/CPU0:router(config-if-tunte-autobw)#ab</td>
</tr>
</tbody>
</table>
### Configuring the Automatic Bandwidth

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>application</strong> minutes</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-tunte-autobw)# application 1000</td>
</tr>
<tr>
<td></td>
<td>Configures the application frequency in minutes for the applicable tunnel.</td>
</tr>
<tr>
<td><strong>minutes</strong></td>
<td>Frequency in minutes for the automatic bandwidth application. Range is from 5 to 10080 (7 days). The default value is 1440 (24 hours).</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>bw-limit</strong> {min} bandwidth {max} bandwidth</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-tunte-autobw)# bw-limit min 30 max 80</td>
</tr>
<tr>
<td></td>
<td>Configures the minimum and maximum automatic bandwidth set on a tunnel.</td>
</tr>
<tr>
<td><strong>min</strong></td>
<td>Applies the minimum automatic bandwidth in kbps on a tunnel. Range is from 0 to 4294967295.</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>Applies the maximum automatic bandwidth in kbps on a tunnel. Range is from 0 to 4294967295.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>adjustment-threshold</strong> percentage [min minimum-bandwidth]</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-tunte-autobw)# adjustment-threshold 50 min 800</td>
</tr>
<tr>
<td></td>
<td>Configures the tunnel bandwidth change threshold to trigger an adjustment.</td>
</tr>
<tr>
<td><strong>percentage</strong></td>
<td>Bandwidth change percent threshold to trigger an adjustment if the largest sample percentage is higher or lower than the current tunnel bandwidth. Range is from 1 to 100 percent. The default value is 5 percent.</td>
</tr>
<tr>
<td><strong>min</strong></td>
<td>Configures the bandwidth change value to trigger an adjustment. The tunnel bandwidth is changed only if the largest sample is higher or lower than the current tunnel bandwidth. Range is from 10 to 4294967295 kilobits per second (kbps). The default value is 10 kbps.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>overflow threshold</strong> percentage [min bandwidth] limit</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-tunte-autobw)# overflow threshold 100 limit 1</td>
</tr>
<tr>
<td></td>
<td>Configures the tunnel overflow detection.</td>
</tr>
<tr>
<td><strong>percentage</strong></td>
<td>Bandwidth change percent to trigger an overflow. Range is from 1 to 100 percent.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>limit</td>
<td>Configures the number of consecutive collection intervals that exceeds the threshold. The bandwidth overflow triggers an early tunnel bandwidth update. Range is from 1 to 10 collection periods. The default value is none.</td>
</tr>
<tr>
<td>min</td>
<td>Configures the bandwidth change value in kbps to trigger an overflow. Range is from 10 to 4294967295. The default value is 10.</td>
</tr>
</tbody>
</table>

**Step 8**

- `commit`

**Step 9**

- `show mpls traffic-eng tunnels [auto-bw]`

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

  **Displays the MPLS-TE tunnel information only for tunnels in which the automatic bandwidth is enabled.**

**Related Topics**

- [MPLS-TE Automatic Bandwidth Overview](#), on page 133
- [Configure Automatic Bandwidth: Example](#), on page 268

**Configuring the Shared Risk Link Groups**

To activate the MPLS traffic engineering SRLG feature, you must configure the SRLG value of each link that has a shared risk with another link.

**Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link**

Perform this task to configure the SRLG value for each link that has a shared risk with another link.

**Note**

You can configure up to 30 SRLGs per interface.
SUMMARY STEPS

1. configure
2. srlg
3. interface type interface-path-id
4. value value
5. commit
6. show srlg interface type interface-path-id
7. show srlg

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>srlg</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# srlg</td>
<td></td>
</tr>
<tr>
<td>Configures SRLG configuration commands on a specific interface configuration mode and assigns this SRLG a value.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-srlg)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Configures an interface type and path ID to be associated with an SRLG and enters SRLG interface configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>value value</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-srlg-if)# value 100 RP/0/RP0/CPU0:router(config-srlg-if)# value 200 RP/0/RP0/CPU0:router(config-srlg-if)# value 300</td>
<td></td>
</tr>
<tr>
<td>Configures SRLG network values for a specific interface. Range is 0 to 4294967295. Note You can also set SRLG values on multiple interfaces including bundle interface.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>commit</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>show srlg interface type interface-path-id</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# show srlg interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>(Optional) Displays the SRLG values configured for a specific interface.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>show srlg</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# show srlg</td>
<td></td>
</tr>
<tr>
<td>(Optional) Displays the SRLG values for all the configured interfaces. Note You can configure up to 250 interfaces.</td>
<td></td>
</tr>
</tbody>
</table>
Creating an Explicit Path With Exclude SRLG

Perform this task to create an explicit path with the exclude SRLG option.

**SUMMARY STEPS**

1. configure
2. explicit-path {identifier number [disable | index]}{ name explicit-path-name}
3. index 1 exclude-address 192.168.92.1
4. index 2 exclude-srlg 192.168.92.2
5. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>`explicit-path {identifier number [disable</td>
<td>index]}{ name explicit-path-name}`</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>index 1 exclude-address 192.168.92.1</code></td>
<td>Specifies the IP address to be excluded from the explicit path.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config-expl-path)# index 1 exclude-address 192.168.92.1</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>index 2 exclude-srlg 192.168.92.2</code></td>
<td>Specifies the IP address to extract SRLGs to be excluded from the explicit path.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RP0/CPU0:router(config-expl-path)# index 2 exclude-srlg 192.168.192.2</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
Using Explicit Path With Exclude SRLG

Perform this task to use an explicit path with the exclude SRLG option on the static backup tunnel.

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. backup-path tunnel-te tunnel-number
5. exit
6. exit
7. interface tunnel-te tunnel-id
8. ipv4 unnumbered type interface-path-id
9. path-option preference-priority { dynamic | explicit {identifier | name explicit-path-name} }
10. destination ip-address
11. exit
12. commit
13. show run explicit-path name name
14. show mpls traffic-eng topology path destination name explicit-path name

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
</tr>
</tbody>
</table>

Example:

RP/0/RP0/CPU0:router(config)# mpls traffic-eng
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td>Enables traffic engineering on a specific interface on the originating node.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>backup-path tunnel-te tunnel-number</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2</td>
<td>Configures an MPLS TE backup path for a specific interface.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>interface tunnel-tetunnel-id</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 2</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>ipv4 unnumbered type interface-path-id</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td>Assigns a source address to set up forwarding on the new tunnel.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>path-option preference-priority{ dynamic</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# path-option l explicit name backup-srlg</td>
<td>Sets the path option to explicit with a given name (previously configured) and assigns the path ID.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>You can use the dynamic option to dynamically assign a path.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>destination ip-address</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125</td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.</td>
</tr>
<tr>
<td>• Destination address is the remote node’s MPLS-TE router ID.</td>
<td></td>
</tr>
<tr>
<td>• Destination address is the merge point between backup and protected tunnels.</td>
<td></td>
</tr>
</tbody>
</table>
### Creating a Link Protection on Backup Tunnel with SRLG Constraint

Perform this task to create an explicit path with the exclude SRLG option on the static backup tunnel.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. backup-path tunnel-te tunnel-number
5. exit
6. exit
7. interface tunnel-te tunnel-id
8. ipv4 unnumbered type interface-path-id
9. path-option preference-priority { dynamic | explicit { identifier | name explicit-path-name } }
10. destination ip-address
11. exit
12. explicit-path { identifier number [disable | index] } { name explicit-path-name }
13. index 1 exclude-srlg 192.168.92.2
14. commit
15. show mpls traffic-eng tunnelstunnel-number detail

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3 interface type interface-path-id</td>
<td>Sets the backup path to the primary tunnel outgoing interface.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4 backup-path tunnel-te tunnel-number</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>Step 5 exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 6 exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Configures an MPLS-TE tunnel interface.**

**Example:**
```
RP/0/RP0/CPU0:router(config)# interface tunnel-te 2
```

### Step 7

**Command or Action:**
`interface tunnel-te tunnel-id`

**Purpose:**
Assigns a source address to set up forwarding on the new tunnel.

**Example:**
```
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered type interface-path-id
```

### Step 8

**Command or Action:**
`ipv4 unnumbered type interface-path-id`

**Purpose:**
Sets the path option to explicit with a given name (previously configured) and assigns the path ID. Identifier range is from 1 to 4294967295.

**Example:**
```
RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg
```

### Step 9

**Command or Action:**
`destination ip-address`

**Purpose:**
Assigns a destination address on the new tunnel.

- Destination address is the remote node’s MPLS-TE router ID.
- Destination address is the merge point between backup and protected tunnels.

**Note**
When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.

**Example:**
```
RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125
```

### Step 10

**Command or Action:**
`exit`

**Purpose:**
Exits the current configuration mode.

**Example:**
```
RP/0/RP0/CPU0:router(config-if)# exit
```

### Step 11

**Command or Action:**
`explicit-path {identifier number [disable | index]}{name explicit-path-name}`

**Purpose:**
Enters the explicit path configuration mode. Identifier range is 1 to 65535.

**Example:**
```
RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg-nodep
```

### Step 12

**Command or Action:**
`index 1 exclude-srlg 192.168.92.2`

**Purpose:**
 Specifies the protected link IP address to get SRLGs to be excluded from the explicit path.

**Example:**
```
RP/0/RP0/CPU0:router:router(config-if)# index 1 exclude-srlg 192.168.192.2
```

### Step 13

**Command or Action:**
`commit`

**Purpose:**
Confirms the configuration changes.
### Purpose

**Command or Action**

- `show mpls traffic-eng tunnels tunnel-number detail`

**Purpose**

Display the tunnel details with SRLG values that are configured for the link.

**Example:**

`RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 2 detail`

---

### Related Topics

- **MPLS Traffic Engineering Shared Risk Link Groups**, on page 141
- **Explicit Path**, on page 142
- **Fast ReRoute with SRLG Constraints**, on page 142
- **Importance of Protection**, on page 144
- **Delivery of Packets During a Failure**, on page 145
- **Multiple Backup Tunnels Protecting the Same Interface**, on page 145
- **SRLG Limitations**, on page 145
- **Configure the MPLS-TE Shared Risk Link Groups: Example**, on page 268

---

### Creating a Node Protection on Backup Tunnel with SRLG Constraint

Perform this task to configure node protection on backup tunnel with SRLG constraint.

**SUMMARY STEPS**

1. configure
2. `mpls traffic-eng`
3. `interface type interface-path-id`
4. `backup-path tunnel-te tunnel-number`
5. `exit`
6. `exit`
7. `interface tunnel-te tunnel-id`
8. `ipv4 unnumbered type interface-path-id`
9. `path-option preference-priority\{ dynamic | explicit \{identifier | name explicit-path-name\}\}`
10. `destination ip-address`
11. `exit`
12. `explicit-path \{identifier number [disable | index]\}\{ name explicit-path-name\}`
13. `index 1 exclude-address 192.168.92.1`
14. `index 2 exclude-srlg 192.168.92.2`
15. `commit`
16. `show mpls traffic-eng tunnels topology path destination ip-address explicit-path-name name`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3 interface type interface-path-id</td>
<td>Sets the backup path for the primary tunnel outgoing interface.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4 backup-path tunnel-te tunnel-number</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>Step 5 exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 6 exit</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 7 interface tunnel-te tunnel-id</td>
<td>Assigns a source address to set up forwarding on the new tunnel.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# interface tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>Step 8 ipv4 unnumbered type interface-path-id</td>
<td>Sets the path option to explicit with a given name (previously configured) and assigns the path ID. Identifier range is 1 to 4294967295.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td>Note: You can use the dynamic option to dynamically assign path.</td>
</tr>
<tr>
<td>Step 9 path-option preference-priority { dynamic</td>
<td>explicit {identifier</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg</td>
<td></td>
</tr>
<tr>
<td>Step 10 destination ip-address</td>
<td>- Destination address is the remote node’s MPLS-TE router ID.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# destination 192.168.92.125</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• Destination address is the merge point between backup and protected tunnels.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> When you configure TE tunnel with multiple protection on its path and merge point is the same node for more than one protection, you must configure record-route for that tunnel.</td>
</tr>
<tr>
<td>Step 11 exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 12 explicit-path {identifier number [disable</td>
<td>index]}{ name explicit-path-name}</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# explicit-path name backup-srlg-nodep</td>
<td></td>
</tr>
<tr>
<td>Step 13 index 1 exclude-address 192.168.92.1</td>
<td>Specifies the protected node IP address to be excluded from the explicit path.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router:router(config-if)# index 1 exclude-address 192.168.92.1</td>
<td></td>
</tr>
<tr>
<td>Step 14 index 2 exclude-srlg 192.168.92.2</td>
<td>Specifies the protected link IP address to get SRLGs to be excluded from the explicit path.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-if)# index 2 exclude-srlg 192.168.192.2</td>
<td></td>
</tr>
<tr>
<td>Step 15 commit</td>
<td></td>
</tr>
<tr>
<td>Step 16 show mpls traffic-eng tunnels topology path destination ip-address explicit-path-name name</td>
<td>Displays the path to the destination with the constraint specified in the explicit path.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels topology path destination 192.168.92.125 explicit-path-name backup-srlg-nodep</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- [MPLS Traffic Engineering Shared Risk Link Groups](#), on page 141
- [Explicit Path](#), on page 142
- [Fast ReRoute with SRLG Constraints](#), on page 142
- [Importance of Protection](#), on page 144
- [Delivery of Packets During a Failure](#), on page 145
- [Multiple Backup Tunnels Protecting the Same Interface](#), on page 145
- [SRLG Limitations](#), on page 145
Configuring Point-to-Multipoint TE

You must enable multicast routing on the edge router before performing Point-to-Multipoint (P2MP) TE configurations. To configure Point-to-Multipoint TE, perform these procedures:

**Enabling Multicast Routing on the Router**

Perform this task to enable multicast routing on the router to configure P2MP tunnels.

**Before You Begin**

- To configure Point-to-Multipoint (P2MP) tunnels, you must enable multicast routing on the router.
- The customer-facing interface must enable multicast.

**SUMMARY STEPS**

1. configure
2. multicast-routing
3. address-family {ipv4 | ipv6 }
4. interface tunnel-mte tunnel-id
5. enable
6. exit
7. interface type interface-path-id
8. enable
9. commit
10. show pim ipv6 interface type interface-path-id

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters multicast routing configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>multicast-routing</td>
<td>Configures the available IPv4 or IPv6 address prefixes to enable multicast routing and forwarding on all router interfaces.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# multicast-routing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mcast)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>address-family {ipv4</td>
<td>ipv6 }</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mcast)# address-family ipv6</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mcast-default-ipv6)#</td>
<td>Configures an MPLS-TE P2MP tunnel interface.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> interface tunnel-mte tunnel-id</td>
<td>Enables multicast routing on the tunnel-mte interface.</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6)# interface tunnel-mte 1 RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> enable</td>
<td>Exits the current configuration mode.</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)# enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Enables multicast routing on the GigabitEthernet interface.</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)# exit RP/0/RP0/CPU0:router(config-mcast-default-ipv6)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> interface type interface-path-id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6)# interface GigabitEthernet0/2/0/3 RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mcast-default-ipv6-if)# enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> commit</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> show pim ipv6 interface type interface-path-id</td>
<td>Displays the output for the P2MP-TE tunnel interface that has IPv6 multicast enabled.</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router# show pim ipv6 interface tunnel-mte 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuring the Static Group for the Point-to-Multipoint Interface

Perform this task to configure the static group on the Point-to-Multipoint (P2MP) interface to forward specified multicast traffic over P2MP LSP.

**SUMMARY STEPS**

1. `configure`
2. `router mld`
3. `vrf vrf-name`
4. `interface tunnel-mte tunnel-id`
5. `static-group group-address`
6. `commit`
7. `show mrib ipv6 route source-address`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>router mld</code></td>
<td>Enters router MLD configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# <code>router mld</code></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mld)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>vrf vrf-name</code></td>
<td>Configures a virtual private network (VRF) instance.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mld)#<code>vrf default</code></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mld-default)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>interface tunnel-mte tunnel-id</code></td>
<td>Configures an MPLS-TE P2MP tunnel interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mld-default-if)#</td>
<td></td>
</tr>
<tr>
<td><code>interface tunnel-mte 1</code></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mld-default-if)#</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Destinations for the Tunnel Interface

Perform this task to configure three destinations for the tunnel interface for Point-to-Multipoint (P2MP). These variations are listed to ensure that the destination and path option configurations are separate from the tunnel interface.

- Different path option is used for different destinations. This task shows three destinations.
- Explicit path option is based on an ID or a name.
- Default path option is similar to the Point-to-Point (P2P) LSP.

### Before You Begin

These prerequisites are required to configure destinations for the tunnel interface.

- Multicast routing must be enabled on both the tunnel-mte interface and customer-facing interface from the source.
- Static-group must be configured on the tunnel-mte interface to forward specified multicast traffic over P2MP LSP.
SUMMARY STEPS

1. configure
2. interface tunnel-mte tunnel-id
3. destination ip-address
4. path-option preference-priority explicit identifier path-number
5. path-option preference-priority dynamic
6. exit
7. destination ip-address
8. path-option preference-priority explicit name pathname
9. path-option preference-priority dynamic
10. exit
11. destination ip-address
12. path-option preference-priority explicit name pathname [verbatim]
13. commit
14. show mpls traffic-eng tunnels [brief] [p2mp tunnel-number]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Configures an MPLS-TE P2MP tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel-mte tunnel-id</td>
<td>Sets the destination address for tunnel-mte 10 to 172.16.255.1. This destination uses the explicit path identified by explicit path ID 10. If destination 172.16.255.1 cannot come with explicit path ID 10, the fallback path option is dynamic.</td>
</tr>
<tr>
<td><strong>Step 3</strong> destination ip-address</td>
<td>Configures the path number of the IP explicit path.</td>
</tr>
<tr>
<td><strong>Step 4</strong> path-option preference-priority explicit identifier path-number</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>path-option preference-priority dynamic</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 2 dynamic</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Specifies that label switched paths (LSP) are dynamically calculated.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)# exit RP/0/RP0/CPU0:router(config-if)#</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>destination ip-address</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# destination 172.16.255.2 RP/0/RP0/CPU0:router(config-if-p2mp-dest)#</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Sets the destination address for tunnel-mte 10 to 172.16.255.2.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>path-option preference-priority explicit name pathname</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 explicit name how-to-get-to-172.16.255.2</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Specifies the path name of the IP explicit path. Destination 172.16.255.2 uses the explicit path that is identified by the explicit path name &quot;how-to-get-to-172.16.255.2.&quot;</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>path-option preference-priority dynamic</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 2 dynamic</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Sets the fall back path option as dynamic when the destination cannot come to the explicit path.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)# exit RP/0/RP0/CPU0:router(config-if)#</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>destination ip-address</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# destination 172.16.255.3 RP/0/RP0/CPU0:router(config-if-p2mp-dest)#</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Specifies that destination 172.16.255.3 uses only the dynamically computed path.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>path-option preference-priority explicit name pathname [verbatim]</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 explicit name how-to-get-to-172.16.255.3 verbatim</code>&lt;br&gt;Specifies that destination 172.16.255.3 uses the explicit path identified by the explicit path name &quot;how-to-get-to-172.16.255.3&quot; in verbatim mode.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>commit</strong>&lt;br&gt;<strong>Step 14</strong></td>
</tr>
</tbody>
</table>

**Related Topics**

- Path Option for Point-to-Multipoint RSVP-TE, on page 139
- Configure the Point-to-Multipoint Tunnel: Example, on page 278
- Configure the Point-to-Multipoint Solution: Example, on page 279
- Enabling Multicast Routing on the Router, on page 228
- Configuring the Static Group for the Point-to-Multipoint Interface, on page 230

**Disabling Destinations**

Perform this task to disable the given destination for the Point-to-Multipoint (P2MP) tunnel interface.

**SUMMARY STEPS**

1. `configure`
2. `interface tunnel-mte tunnel-id`
3. `ipv4 unnumbered type interface-path-id`
4. `destination ip-address`
5. `disable`
6. `path-option preference-priority dynamic`
7. `path-option preference-priority explicit name pathname`
8. `commit`
## Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Configures an MPLS-TE P2MP tunnel interface.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-mte tunnel-id</td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config)# interface tunnel-mte 101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>ipv4 unnumbered type interface-path-id</td>
<td>Sets the destination address for tunnel-mte 10 to 140.140.140.140.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>destination ip-address</td>
<td>Disables destination 140.140.140.140 for tunnel-mte 10.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)#disable</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>path-option preference-priority dynamic</td>
<td>Specifies that label switched paths (LSP) are dynamically calculated.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)#path-option 1 dynamic</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>path-option preference-priority explicit name pathname</td>
<td>Specifies that destination 140.140.140.140 uses the explicit path identified by the explicit path name &quot;to4.&quot;</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if-p2mp-dest)#path-option 2 explicit name to4</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

Cisco IOS XR MPLS Configuration Guide for the Cisco CRS Router, Release 4.3.x
Logging Per Destinations for Point-to-Multipoint

Perform this task to log destinations for Point-to-Multipoint (P2MP).

SUMMARY STEPS

1. configure
2. interface tunnel-mte tunnel-id
3. ipv4 unnumbered type interface-path-id
4. destination ip-address
5. logging events lsp-status state
6. logging events lsp-status reroute
7. path-option preference-priority explicit name pathname
8. exit
9. fast-reroute
10. commit
11. show mpls traffic-eng tunnels [p2mp]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Configures an MPLS-TE P2MP tunnel interface.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-mte tunnel-id</td>
<td>Configures the MPLS-TE tunnel to use the IPv4 address on loopback interface 0.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# interface tunnel-mte 1000 RP/0/RP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>ipv4 unnumbered type interface-path-id</td>
<td>Sets the destination address for tunnel-mte from 1000 to 100.0.0.3.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback0</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>100.0.0.3 RP/0/RP0/CPU0:router(config-if-p2mp-dest)#</td>
<td>Sends out the log message when the tunnel LSP goes up or down when the software is enabled.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 5**

**logging events lsp-status state**

**Example:**

```
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# logging events lsp-status state
```

**Step 6**

**logging events lsp-status reroute**

**Example:**

```
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# logging events lsp-status reroute
```

**Step 7**

**path-option preference-priority explicit name pathname**

**Example:**

```
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# path-option 1 explicit name path123
```

**Step 8**

**exit**

**Example:**

```
RP/0/RP0/CPU0:router(config-if-p2mp-dest)# exit
```

**Step 9**

**fast-reroute**

**Example:**

```
RP/0/RP0/CPU0:router(config-if)# fast-reroute
```

**Step 10**

**commit**

**Step 11**

**show mpls traffic-eng tunnels [p2mp]**

**Example:**

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels p2mp
```

**Related Topics**

- Point-to-Multipoint Traffic-Engineering Overview, on page 136
Enabling Soft-Preemption on a Node

Perform this task to enable the soft-preemption feature in the MPLS TE configuration mode. By default, this feature is disabled. You can configure the soft-preemption feature for each node. It has to be explicitly enabled for each node.

### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. soft-preemption
4. timeout *seconds*
5. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>soft-preemption</td>
<td>Enables soft-preemption on a node.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>soft-preemption</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong> If soft-preemption is enabled, the head-end node tracks whether an LSP desires the soft-preemption treatment. However, when a soft-preemption feature is disabled on a node, this node continues to track all LSPs desiring soft-preemption. This is needed in a case when soft-preemption is re-enabled, TE will have the property of the existing LSPs without any re-signaling.</td>
</tr>
<tr>
<td>Step 4</td>
<td>timeout <em>seconds</em></td>
<td>Specifies the timeout for the soft-preempted LSP, in seconds. The range is from 1 to 300.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-soft-preemption)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>timeout 20</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
Enabling Soft-Preemption on a Tunnel

Perform this task to enable the soft-preemption feature on a MPLS TE tunnel. By default, this feature is disabled. It has to be explicitly enabled.

**SUMMARY STEPS**

1. configure
2. interface tunnel-te *tunnel-id*
3. soft-preemption
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface tunnel-te <em>tunnel-id</em></td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RP0/CPU0:router# interface tunnel-te 10</td>
</tr>
<tr>
<td></td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>soft-preemption</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RP0/CPU0:router(config-if)# soft-preemption</td>
</tr>
<tr>
<td></td>
<td>Enables soft-preemption on a tunnel.</td>
</tr>
<tr>
<td></td>
<td>When soft preemption is enabled on a tunnel, these actions occur:</td>
</tr>
<tr>
<td></td>
<td>• A path-modify message is sent for the current LSP with the <strong>soft</strong></td>
</tr>
<tr>
<td></td>
<td><strong>preemption desired</strong> property.</td>
</tr>
<tr>
<td></td>
<td>• A path-modify message is sent for the reopt LSP with the <strong>soft</strong></td>
</tr>
<tr>
<td></td>
<td><strong>preemption desired</strong> property.</td>
</tr>
<tr>
<td></td>
<td>• A path-modify message is sent for the path protection LSP with the</td>
</tr>
<tr>
<td></td>
<td><strong>soft preemption desired</strong> property.</td>
</tr>
<tr>
<td></td>
<td>• A path-modify message is sent for the current LSP in FRR active state</td>
</tr>
<tr>
<td></td>
<td>with the <strong>soft preemption desired</strong> property.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

**Note** The soft-preemption is not available in the interface tunnel-mte and interface tunnel-gte configuration modes.

**Related Topics**

Soft-Preemption, on page 146
Configuring Attributes within a Path-Option Attribute

Perform this task to configure attributes within a path option attribute-set template.

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. attribute-set path-option attribute-set-name
4. affinity affinity-value mask mask-value
5. signalled-bandwidth kbps class-type class-type number
6. commit
7. show mpls traffic-eng attribute-set
8. show mpls traffic-eng tunnels detail

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enters attribute-set path option configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**  
RP/0/RP0/CPU0:router(config)# mpls traffic-eng | |
| **Step 3** attribute-set path-option attribute-set-name | Configures affinity attribute under a path option attribute-set.  
The attribute values that are required for links to carry this tunnel.  
**Note** The configuration at the path-option level takes precedence over the values configured at the level of the tunnel, and therefore is applied. |
| **Example:**  
RP/0/RP0/CPU0:router(config-mpls-te)# attribute-set path-option myset | |
| **Step 4** affinity affinity-value mask mask-value | |
| **Example:**  
RP/0/RP0/CPU0:router(config-te-attribute-set)# affinity 0xBEEF mask 0xBEEF | |
| **Step 5** signalled-bandwidth kbps class-type class-type number | |
| **Example:**  
RP/0/RP0/CPU0:router(config-te-attribute-set)# signalled-bandwidth 1000 class-type 0 | |
| **Step 6** commit | |

You can configure the class type of the tunnel bandwidth request. The class-type 0 is strictly equivalent to global-pool and class-type 1 is strictly equivalent to subpool.
### Configuring Auto-Tunnel Mesh Tunnel ID

Perform this activity to configure the tunnel ID range that can be allocated to Auto-tunnel mesh tunnels.

**SUMMARY STEPS**

1. **configure**
2. **mpls traffic-eng**
3. **auto-tunnel mesh**
4. **tunnel-id min value max value**
5. **commit**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
```
### Configuring Auto-tunnel Mesh Unused Timeout

Perform this task to configure a global timer to remove unused auto-mesh tunnels.

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. auto-tunnel mesh
4. timer removal unused *timeout*
5. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RP0/CPU0:router(config)# mpls traffic-eng
```
### Configuring Auto-Tunnel Mesh Group

Perform this task to configure an auto-tunnel mesh group globally on the router.

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. auto-tunnel mesh
4. group value
5. disable
6. attribute-set name
7. destination-list
8. commit

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>auto-tunnel mesh</td>
<td>Enables auto-tunnel mesh groups globally.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel mesh</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>timer removal unused timeout</td>
<td>Specifies a timer, in minutes, after which a down auto-tunnel mesh gets deleted whose destination was not in TE topology. The default value for this timer is 60. The timer gets started when these conditions are met: • Tunnel destination node is removed from the topology • Tunnel is in down state</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# timers removal unused 10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

Auto-Tunnel Mesh, on page 149
Destination List (Prefix-List), on page 149
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> auto-tunnel mesh</td>
<td>Enables auto-tunnel mesh groups globally.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:router(config-mpls-te)# auto-tunnel mesh</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> group value</td>
<td>Specifies the membership of auto-tunnel mesh. The range is from 0 to 4294967295.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh)# group 65</code></td>
<td><strong>Note</strong> When the destination-list is not supplied, head-end will automatically build destination list belonging for the given mesh-group membership using TE topology.</td>
</tr>
<tr>
<td><strong>Step 5</strong> disable</td>
<td>Disables the meshgroup and deletes all tunnels created for this meshgroup.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# disable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> attribute-set name</td>
<td>Specifies the attributes used for all tunnels created for the meshgroup. If it is not defined, this meshgroup does not create any tunnel.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# attribute-set am-65</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> destination-list</td>
<td>This is a mandatory configuration under a meshgroup. If a given destination-list is not defined as a prefix-list, this meshgroup create tunnels to all nodes available in TE topology.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RP0/CPU0:router(config-mpls-te-auto-mesh-group)# destination-list dl-65</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Auto-Tunnel Mesh, on page 149
- Destination List (Prefix-List), on page 149
# Configuring Tunnel Attribute-Set Templates

Perform this task to define attribute-set templates for auto-mesh tunnels.

## SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng`
3. `attribute-set auto-mesh attribute-set-name`
4. `affinity value mask mask-value`
5. `signalled-bandwidth kbps class-type class-type number`
6. `autoroute announce`
7. `fast-reroute protect bandwidth node`
8. `auto-bw collect-bw-only`
9. `logging events lsp-status {state | insufficient-bandwidth | reoptimize | reroute}`
10. `commit`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**        | `configure`
| **Step 2**        | `mpls traffic-eng`
| Example:          | `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`
| **Step 3**        | `attribute-set auto-mesh attribute-set-name`
| Example:          | `RP/0/RP0/CPU0:router(config-te)# attribute-set auto-mesh attribute-set-name`
| **Step 4**        | `affinity value mask mask-value`
| Example:          | `RP/0/RP0/CPU0:router(config-te)# affinity 0101 mask 320`
| **Step 5**        | `signalled-bandwidth kbps class-type class-type number`
| Example:          | `RP/0/RP0/CPU0:router(config-te-attribute-set)# signalled-bandwidth 1000 class-type 0`

Configures the bandwidth attribute required for an MPLS-TE tunnel under an auto-mesh attribute-set. Because the default tunnel priority is 7, tunnels use the default TE class map (namely, class-type 0, priority 7).
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> autoroute announce</td>
<td>Enables parameters for IGP routing over tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-te-attribute-set)# autoroute announce</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> fast-reroute protect bandwidth node</td>
<td>Enables fast-reroute bandwidth protection and node protection for auto-mesh tunnels.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-te-attribute-set)# fast-reroute</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> auto-bw collect-bw-only</td>
<td>Enables automatic bandwidth collection frequency, and controls the manner in which the bandwidth for a tunnel collects output rate information, but does not adjust the tunnel bandwidth.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-te-attribute-set)# auto-bw collect-bw-only</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> logging events lsp-status {state</td>
<td>insufficient-bandwidth</td>
</tr>
<tr>
<td>Example:</td>
<td>Sends out the log message when the tunnel LSP undergoes setup or reoptimize failure due to bandwidth issues.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-te-attribute-set)# logging events lsp-status state</td>
<td>Sends out the log message for the LSP reoptimize change alarms.</td>
</tr>
<tr>
<td></td>
<td>Sends out the log message for the LSP reroute change alarms.</td>
</tr>
<tr>
<td><strong>Step 10</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Auto-Tunnel Mesh, on page 149
- Destination List (Prefix-List), on page 149

**Enabling LDP on Auto-Tunnel Mesh**

Perform this task to enable LDP on auto-tunnel mesh group.
SUMMARY STEPS

1. configure
2. mpls ldp
3. traffic-eng auto-tunnel mesh
4. group id all
5. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls ldp</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>traffic-eng auto-tunnel mesh</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters auto-tunnel mesh configuration mode. You can configure TE auto-tunnel mesh groups from this mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp-te-auto-mesh)# traffic-eng auto-tunnel mesh</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>group id all</td>
</tr>
<tr>
<td>Example:</td>
<td>Configures an auto-tunnel mesh group of interfaces in LDP. You can enable LDP on all TE mesh group interfaces or you can specify the TE mesh group ID on which the LDP is enabled. The range of group ID is from 0 to 4294967295.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-ldp-te-auto-mesh)# group all</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
</tr>
</tbody>
</table>

Related Topics

- Auto-Tunnel Mesh, on page 149
- Destination List (Prefix-List), on page 149

Implementing Associated Bidirectional Label Switched Paths

This section describes how to configure MPLS Traffic Engineering Associated Bidirectional Label Switched Paths (MPLS-TE LSPs).

Associated Bidirectional Label Switched Paths are LSP instances where the forward and the reverse direction paths are setup, monitored and protected independently and associated together during signaling. You use a RSVP Association object to bind the two forward and reverse LSPs together to form either a co-routed or non co-routed associated bidirectional TE tunnel.
Signaling Methods and Object Association for Bidirectional LSPs, on page 248, Associated Bidirectional Non Co-routed and Co-routed LSPs, on page 249 provides details.

You can associate a protecting MPLS-TE tunnel with either a working MPLS-TE LSP, protecting MPLS-TE LSP, or both. The working LSP is the primary LSP backed up by the protecting LSP. When a working LSP goes down, the protecting LSP is automatically activated. You can configure a MPLS-TE tunnel to operate without protection as well.

Path Protection, on page 253 provides details.

**Signaling Methods and Object Association for Bidirectional LSPs**

This section provides an overview of the association signaling methods for the bidirectional LSPs. Two unidirectional LSPs can be bound to form an associated bidirectional LSP in the following scenarios:

- No unidirectional LSP exists, and both must be established.
- Both unidirectional LSPs exist, but the association must be established.
- One unidirectional LSP exists, but the reverse associated LSP must be established.

Configuration information regarding the LSPs can be provided at one or both endpoints of the associated bidirectional LSP. Depending on the method chosen, there are two models of creating an associated bidirectional LSP; single-sided provisioning, and double-sided provisioning.

- **Single-sided Provisioning:** For the single-sided provisioning, the TE tunnel is configured only on one side. An LSP for this tunnel is initiated by the initiating endpoint with the Association Object inserted in the Path message. The other endpoint then creates the corresponding reverse TE tunnel and signals the reverse LSP in response to this. Currently, there is no support available for configuring single-sided provisioning.

- **Double-sided Provisioning:** For the double-sided provisioning, two unidirectional TE tunnels are configured independently on both sides. The LSPs for the tunnels are signaled with Association Objects inserted in the Path message by both sides to indicate that the two LSPs are to be associated to form a bidirectional LSP.

Consider this topology (an example of associated bidirectional LSP):

Here, LSP1 from A to B, takes the path A,D,B and LSP2 from B to A takes the path B,D,C,A. These two LSPs, once established and associated, form an associated bidirectional LSP between node A and node B. For the double sided provisioning model, both LSP1 and LSP2 are signaled independently with (Extended) Association Object inserted in the Path message, in which the Association Type indicating double-sided provisioning. In this case, the two unidirectional LSPs are bound together to form an associated bidirectional LSP based on identical Association Objects in the two LSPs’ Path messages.

**Association Object:** An Association Object is used to bind unidirectional LSPs originating from both endpoints. The Association Object takes the following values:
• **Association Type:** In order to bind two reverse unidirectional LSPs to be an associated bidirectional LSP, the Association Type must be set to indicate either single sided or double sided LSPs.

• **Association ID:** For both single sided and double sided provisioning, Association ID must be set to a value assigned by the node that originates the association for the bidirectional LSP. This is set to the Tunnel ID of the bound LSP or the Tunnel ID of the binding LSP.

• **Association Source:** For double sided provisioning, Association Source must be set to an address selected by the node that originates the association for the bidirectional LSP. For single sided provisioning, Association Source must be set to an address assigned to the node that originates the LSP.

• **Global ID:** This is the global ID for the association global source. This must be set to the global ID of the node that originates the association for the bidirectional LSP.

---

**Note**
You must provide identical values for the content of the Association Object on either end of the participating LSPs to ensure successful binding of the LSPs.

Configure Associated Bidirectional Co-routed LSPs, on page 251 describes the procedure to create associated bidirectional co-routed LSPs.

---

**Associated Bidirectional Non Co-routed and Co-routed LSPs**

This section provides an overview of associated bidirectional non co-routed and co-routed LSPs. Establishment of MPLS TE-LSP involves computation of a path between a head-end node to a tail-end node, signaling along the path, and modification of intermediate nodes along the path. The signaling process ensures bandwidth reservation (if signaled bandwidth is lesser than 0 and programming of forwarding entries).

Path computation is performed by the head-end nodes of both the participating LSPs using Constrained Shortest Path First (CSPF). CSPF is the ’shortest path (measured in terms of cost) that satisfies all relevant LSP TE constraints or attributes, such as required bandwidth, priority and so on.

**Associated Bidirectional Non Co-routed LSPs:** A non co-routed bidirectional TE LSP follows two different paths, that is, the forward direction LSP path is different than the reverse direction LSP path. Here is an illustration.
In the above topology:

- The outer paths (in green) are working LSP pairs.
- The inner paths (in red) are protecting LSP pairs.
- Router 1 sets up working LSP to Router 3 and protecting LSP to Router 3 independently.
- Router 3 sets up working LSP to Router 1 and protecting LSP to Router 1 independently.

Non co-routed bidirectional TE LSP is available by default, and no configuration is required.

**Note**

In case of non co-routed LSPs, the head-end nodes relax the constraint on having identical forward and reverse paths. Hence, depending on network state you can have identical forward and reverse paths, though the bidirectional LSP is co-routed.

**Associated Bidirectional Co-routed LSPs:** A co-routed bidirectional TE LSP denotes a bidirectional tunnel where the forward direction LSP and reverse direction LSP must follow the same path, for example, the same nodes and paths. Here is an illustration.

In the above topology:

- Paths at the top of the figure (in green) indicate working co-routed LSP pairs.
• Paths at the bottom of the figure (in red) indicate protecting co-routed LSP pairs.

• Router 1 sets up working LSP to Router 3 (in red) after performing bidirectional CSPF and sends reverse explicit route object (ERO) to Router 3. Node Router 3 uses the received reverse ERO to set up reverse red working LSP to Router 1.

• Router 3 sets up protecting LSP to Router 1 (in green) after performing bidirectional CSPF and sends reverse ERO to Router 1. Node Router 1 uses the received reverse ERO to set up reverse green protecting LSP to Router 3.

Configure Associated Bidirectional Co-routed LSPs, on page 251, describes the procedure to configure an associated bidirectional co-routed LSP.

Configure Associated Bidirectional Co-routed LSPs

A co-routed bidirectional packet LSP is a combination of two LSPs (one in the forward direction and the other in reverse direction) sharing the same path between a pair of ingress and egress nodes. It is established using the extensions to RSVP-TE. This type of LSP can be used to carry any of the standard types of MPLS-based traffic, including Layer 2 VPNs, Layer 2 circuits, and Layer 3 VPNs. You can configure a single BFD session for the bidirectional LSP (that is, you do not need to configure a BFD session for each LSP in each direction). You can also configure a single standby bidirectional LSP to provide a backup for the primary bidirectional LSP.

Before You Begin

• You must have symmetric source and destination TE router IDs in order for bidirectional LSPs to be associated.

• Tunnels attributes must be configured identically on both sides of co-routed bidirectional LSP.

<table>
<thead>
<tr>
<th>SUMMARY STEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. configure</td>
</tr>
<tr>
<td>2. interface tunnel-te tunnel-id</td>
</tr>
<tr>
<td>3. bidirectional</td>
</tr>
<tr>
<td>4. association {id &lt;0-65535&gt;</td>
</tr>
<tr>
<td>5. association type co-routed</td>
</tr>
<tr>
<td>6. commit</td>
</tr>
<tr>
<td>7. show mpls traffic-eng tunnels bidirectional-associated co-routed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
</tbody>
</table>

Example:
```
RP/0/RSP0/CPU0:router# interface tunnel-te 1
```
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>bidirectional</td>
<td>Configure the ingress router for the LSP and include the bidirectional statement to specify that the LSP be established as a bidirectional LSP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-if)# bidirectional</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>association {id &lt;0-65535&gt;</td>
<td>Set the association ID that uniquely identifies the association of LSPs, which is the tunnel ID of the bound LSP (master/slave mode) or the tunnel ID of the binding LSP (peer mode). Also, set the source address to the tunnel sender address of the bound LSP (master/slave mode) or the tunnel sender address of the binding LSP (peer mode). Optionally, specify the global ID for association global source.</td>
</tr>
<tr>
<td></td>
<td>source-address &lt;IP address&gt;} [global-id &lt;0-4294967295&gt;]</td>
<td>Note: Association ID, association source and global ID must be configured identically on both the endpoints.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-if-bidir)# association id 1 source-address 11.0.0.1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>association type co-routed</td>
<td>Specify that the LSP be established as a associated co-routed bidirectional LSP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router(config-if-bidir)# association type co-routed</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>show mpls traffic-eng tunnels bidirectional-associated co-routed</td>
<td>Shows details of an associated co-routed bidirectional LSP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/0/CPU0:router# show mpls traffic-eng tunnels bidirectional-associated co-routed</td>
<td></td>
</tr>
</tbody>
</table>

**Show output for an associated co-routed bidirectional LSP configuration**

This is a sample of the output for the `show mpls traffic-eng tunnels role head` command.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels role head
Name: tunnel-tel Destination: 49.49.49.2
Signalled-Name: IMC0_t1
Status:
  Admin: up Oper: up Path: valid Signalling: connected
  path option 1, type dynamic (Basis for Setup, path weight 20 (reverse 20))
  path option 1, type dynamic (Basis for Standby, path weight 20 (reverse 20))
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 0 kbps CT0
  Creation Time: Sun May 4 12:09:56 2014 (03:24:11 ago)
  Config Parameters:
    Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
    Metric Type: TE (default)
    Hop-limit: disabled
    Cost-limit: disabled
    AutoRoute: disabled LockDown: disabled Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Loadshare: 0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
```
Association Type: Single Sided Bidirectional LSPs, Co-routed: YES
Association ID: 100, Source: 49.49.49.2
Reverse Bandwidth: 0 kbps (CT0), Standby: 0 kbps (CT0)
BFD Fast Detection: Enabled
BFD Parameters: Min-interval 100 ms (default), Multiplier 3 (default)
BFD Bringup Timeout: Interval 60 seconds (default)
BFD Initial Dampening: 16000 ms (default)
BFD Maximum Dampening: 600000 ms (default)
BFD Secondary Dampening: 20000 ms (default)
Periodic LSP Ping: Interval 120 seconds (default)
Session Down Action: ACTION_REOPTIMIZE, Reopt Timeout: 300
BFD Encap Mode: GAL
Reoptimization after affinity failure: Enabled
Soft Preemption: Disabled

Path Protection

Path protection provides an end-to-end failure recovery mechanism (that is, full path protection) for associated bidirectional MPLS-TE LSPs. Associated bidirectional MPLS-TE LSPs support 1:1 path protection. You can configure the working and protecting LSPs as part of configuring the MPLS-TE tunnel. The working LSP is the primary LSP used to route traffic, while the protecting LSP is a backup for a working LSP. If the working LSP fails, traffic is switched to the protecting LSP until the working LSP is restored, at which time traffic forwarding reverts back to the working LSP.

When FRR is not enabled on a tunnel, and when GAL-BFD and/or Fault OAM is enabled on an associated bidirectional co-routed LSP, path-protection is activated by the FIB running on the line card that hosts the working LSP. The failure on the working LSP can be detected using BFD or Fault OAM.

Configure Path Protection for Associated Bidirectional LSPs, on page 253 provides procedural details.

You can use the show mpls traffic-eng fast-reroute log command to confirm whether protection switching has been activated by FIB.

Configure Path Protection for Associated Bidirectional LSPs

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. ipv4 unnumbered type interface-path-id
4. bfd {fast-detect | encap-mode}
5. destination ip-address
6. bidirectional
7. bidirectional association {id <0-65535> | source-address <IP address>} [global-id <0-4294967295>]
8. association type co-routed
9. path-protection
10. path-option preference - priority {dynamic | explicit}
11. commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>interface tunnel-te &lt;tunnel-id&gt;</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ipv4 unnumbered &lt;type interface-path-id&gt;</td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>bfd {fast-detect</td>
<td>encap-mode}</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:IMC0(config-if)# bfd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:IMC0(config-if-tunte-bfd)# fast-detect</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:IMC0(config-if-tunte-bfd)# encap-mode gal</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>destination &lt;ip-address&gt;</td>
<td>Assigns a destination address on the new tunnel. The destination address is the remote node's MPLS-TE router ID.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-if)# destination 49.49.49.2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>bidirectional</td>
<td>Configure the ingress router for the LSP and include the bidirectional statement to specify that the LSP be established as a bidirectional LSP.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-if)# bidirectional</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>bidirectional association {id &lt;0-65535&gt;</td>
<td>source-address &lt;IP address&gt;}</td>
</tr>
<tr>
<td></td>
<td>[global-id &lt;0-4294967295&gt;]</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-if-bidir)# association id 1 source-address 11.0.0.1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>association type co-routed</td>
<td>Specify that the LSP be established as an associated co-routed bidirectional LSP.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-if-bidir)# association type co-routed</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Step 9</th>
<th>path-protection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:IMC0(config-if-bidir-co-routed)#path-protection</td>
</tr>
</tbody>
</table>

**Purpose**

Enable path protection.

**Step 10**

**path-option preference - priority {dynamic | explicit}**

**Example:**

RP/0/RP0/CPU0:router(config-if)# path-option 1 dynamic

**Sets the path option and assigns the path-option ID.**

**Note**

Both sides of the co-routed bidirectional LSPs must use dynamic or matching co-routed strict-hop explicit path-option.

**Step 11**

commit

Here is a sample configuration with path protection defined for the Associated Bidirectional LSP.

RP/0/RSP0/CPU0:IMC0#config
RP/0/RSP0/CPU0:IMC0(config)#interface tunnel-te 1
RP/0/RSP0/CPU0:IMC0(config-if)#ipv4 unnumbered loopback0
RP/0/RSP0/CPU0:IMC0(config-if)#destination 49.49.49.2
RP/0/RSP0/CPU0:IMC0(config-if)#bidirectional
RP/0/RSP0/CPU0:IMC0(config-if-bidir)#association id 100 source-address 49.49.49.2
RP/0/RSP0/CPU0:IMC0(config-if-bidir)#association type co-routed
RP/0/RSP0/CPU0:IMC0(config-if-bidir-co-routed)#path-protection
RP/0/RSP0/CPU0:IMC0(config-if-bidir-co-routed)#path-option 1 dynamic
RP/0/RSP0/CPU0:IMC0(config-if-bidir-co-routed)#commit

---

### OAM Support for Associated Bidirectional LSPs

You can opt to configure operations, administration and management (OAM) support for Associated Bidirectional LSPs in the following areas:

- **Continuity check:** You can configure bidirectional forwarding detection (BFD) over a Generic Associated Channel (G-ACh) with hardware assist. This allows for BFD Hello packets to be generated and processed in hardware making smaller Hello intervals such as 3.3 ms feasible. For more information on BFD and BFD hardware offload see *Implementing BFD* module in the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

- **Fault notification:** You can run Fault OAM over associated bidirectional co-routed LSPs to convey fault notification from mid-point to end-point of the LSP. The following fault OAM messages are supported:
  
  - **Link Down Indication (LDI):** generated when an interface goes down (for example, to fiber-cut) at mid-point.
  - **Lock Report (LKR):** generated when an interface is shutdown at mid-point.

  You can configure fault OAM to generate OAM message at mid-point or enable protection switching due to fault OAM at end-point. *Generate Fault OAM Messages at Mid-point, on page 256* and *Generate Fault OAM Messages at End-point, on page 256* provides procedural details.
• **Fault diagnostics**: You can use the ping and traceroute features as a means to check connectivity and isolate failure points for both co-routed and non-co-routed bidirectional TE tunnels. *MPLS Network Management with MPLS LSP Ping and MPLS SP Traceroute* provides details.

### Generate Fault OAM Messages at Mid-point

To program all bi-directional LSPs to generate fault OAM message at mid-point use the following steps:

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. fault-oam
4. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enables fault OAM for an associated bidirectional LSP.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:IMO(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> fault-oam</td>
<td>Enables fault OAM for an associated bidirectional LSP.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:IMO(config-mpls-te)#fault-oam</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

### Generate Fault OAM Messages at End-point

In order to enable protection switching due to fault OAM at end-point use the following steps:

**SUMMARY STEPS**

1. configure
2. interface tunnel-te *tunnel-id*
3. bidirectional association type co-routed fault-oam
4. commit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:IMC0(config)#interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>bidirectional association type co-routed fault-oam</td>
<td>Enable fault OAM for an associated co-routed bidirectional LSP.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:IMC0(config-if)#bidirectional association type co-routed fault-oam</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

### Pseudowire Call Admission Control

You can use the Pseudowire Call Admission Control (PW CAC) process to check for bandwidth constraints and ensure that once the path is signaled, the links (pseudowires) participating in the bidirectional LSP association have the required bandwidth. Only pseudowires with sufficient bandwidth are admitted in the bidirectional LSP association process. *Configure Pseudowire Bandwidth* in the *Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide* provides procedural details.

### Configuration Examples for Cisco MPLS-TE

These configuration examples are used for MPLS-TE:

### Configure Fast Reroute and SONET APS: Example

When SONET Automatic Protection Switching (APS) is configured on a router, it does not offer protection for tunnels; because of this limitation, fast reroute (FRR) still remains the protection mechanism for MPLS-TE.

When APS is configured in a SONET core network, an alarm might be generated toward a router downstream. If this router is configured with FRR, the hold-off timer must be configured at the SONET level to prevent FRR from being triggered while the core network is performing a restoration. Enter the following commands to configure the delay:

```
RP/0/RP0/CPU0:router(config)# controller sonet 0/6/0/0 delay trigger line 250
RP/0/RP0/CPU0:router(config)# controller sonet 0/6/0/0 path delay trigger 300
```
Build MPLS-TE Topology and Tunnels: Example

The following examples show how to build an OSPF and IS-IS topology:

(OSPF)

configure
  mpls traffic-eng
  interface pos 0/6/0/0
  router id loopback 0
  router ospf 1
  router-id 192.168.25.66
  area 0
  interface pos 0/6/0/0
  interface loopback 0
  mpls traffic-eng router-id 192.168.70.1
  mpls traffic-eng area 0
  rsvp
  interface pos 0/6/0/0
  bandwidth 100
  commit
  show mpls traffic-eng topology
  show mpls traffic-eng link-management advertisement

(IS-IS)

configure
  mpls traffic-eng
  interface pos 0/6/0/0
  router id loopback 0
  router isis lab
  address-family ipv4 unicast
  mpls traffic-eng level 2
  mpls traffic-eng router-id 192.168.70.2

The following example shows how to configure tunnel interfaces:

interface tunnel-te1
  destination 192.168.92.125
  ipv4 unnumbered loopback 0
  path-option 1 dynamic
  bandwidth 100
  commit
  show mpls traffic-eng tunnels
  show ipv4 interface brief
  show mpls traffic-eng link-management admission-control

interface tunnel-te1
  autoroute announce
  route ipv4 192.168.12.52/32 tunnel-te1
  commit
  ping 192.168.12.52
  show mpls traffic autoroute

interface tunnel-te1
  fast-reroute
  mpls traffic-eng interface pos 0/6/0/0
  backup-path tunnel-te 2
  interface tunnel-te2
  backup-bw global-pool 5000
  ipv4 unnumbered loopback 0
  path-option 1 explicit name backup-path
  destination 192.168.92.125
commit
show mpls traffic-eng tunnels backup
show mpls traffic-eng fast-reroute database
!
rssvp
  interface pos 0/6/0/0
  bandwidth 100 150 sub-pool 50
  interface tunnel-te1
  bandwidth sub-pool 10
commit

Related Topics
Building MPLS-TE Topology, on page 150
Creating an MPLS-TE Tunnel, on page 153
How MPLS-TE Works, on page 110

Configure IETF DS-TE Tunnels: Example

The following example shows how to configure DS-TE:

rssvp
  interface pos 0/6/0/0
  bandwidth rdm 100 150 bc1 50
  mpls traffic-eng
ds-te mode ietf
  interface tunnel-te 1
  bandwidth 10 class-type 1
commit

configure
rssvp interface 0/6/0/0
  bandwidth mam max-reservable-bw 400 bc0 300 bc1 200
  mpls traffic-eng
ds-te mode ietf
ds-te model mam
  interface tunnel-te 1 bandwidth 10 class-type 1
commit

Related Topics
Configuring a Prestandard DS-TE Tunnel, on page 165
Prestandard DS-TE Mode, on page 116

Configure MPLS-TE and Fast-Reroute on OSPF: Example

CSPF areas are configured on a per-path-option basis. The following example shows how to use the traffic-engineering tunnels (tunnel-te) interface and the active path for the MPLS-TE tunnel:

configure
  interface tunnel-te 0
  path-option 1 explicit id 6 ospf 126 area 0
  path-option 2 explicit name 234 ospf 3 area 7 verbatim
  path-option 3 dynamic isis mtbf level 1 lockdown
commit
Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example

This example shows how to configure the IS-IS overload bit setting in MPLS-TE:

This figure illustrates the IS-IS overload bit scenario:

![Figure 18: IS-IS overload bit](image)

Consider a MPLS TE topology in which usage of nodes that indicated an overload situation was restricted. In this topology, the router R7 exhibits overload situation and hence this node cannot be used during TE CSPF. To overcome this limitation, the IS-IS overload bit avoidance (OLA) feature was introduced. This feature allows network administrators to prevent RSVP-TE label switched paths (LSPs) from being disabled when a router in that path has its Intermediate System-to-Intermediate System (IS-IS) overload bit set.

The IS-IS overload bit avoidance feature is activated at router R1 using this command:

```
mpls traffic-eng path-selection ignore overload
```

```
configure
mpls traffic-eng
path-selection ignore overload
commit
```

Configure GMPLS: Example

This example shows how to set up headend and tailend routers with bidirectional optical unnumbered tunnels using numbered TE links:
Headend Router

```
router ospf roswell
router-id 11.11.11.11
nsf cisco
area 23
!
area 51
  interface Loopback 0
  !
  interface MgmtEth0/0/CPU0/1
  !
  interface POS0/4/0/1
  !
mpls traffic-eng router-id Loopback 0
mpls traffic-eng area 51
!
rsvp
  interface POS0/2/0/3
    bandwidth 2000
    !
  interface tunnel-gte 1
    ipv4 unnumbered Loopback 0
    switching transit fsc encoding
sonetdsdh
  switching endpoint psc1 encoding packet
  priority 3 3
  signalled-bandwidth 500
  destination 55.55.55.55
  path-option 1 dynamic
!
mpls traffic-eng
  interface POS0/2/0/3
    flooding-igp ospf roswell area 51
    switching key 1
    encoding packet
    capability psc1
    !
    switching link
    encoding
sonetdsdh
  capability fsc
  !
  lmp data-link adjacency
  neighbor gmpls5
    remote te-link-id ipv4 10.0.0.5
    remote interface-id unnum 12
    remote switching-capability psc1
    !
  !
  lmp neighbor gmpls5
    ipcc routed
    remote node-id 55.55.55.55
    !
```

Tailend Router

```
router ospf roswell
router-id 55.55.55.55
nsf cisco
area 23
!
area 51
  interface Loopback 0
```
interface MgmtEth0/0/CPU0/1
interface POS0/4/0/2

mpls traffic-eng router-id Loopback 0
mpls traffic-eng area 51

mpls traffic-eng
interface POS0/2/0/3
flooding-igp ospf roswell area 51
switching key 1
encoding packet
capability psc1

switching link
encoding
sonet-sdh
capability fsc

lmp data-link adjacency
neighbor gmpls1
remote te-link-id ipv4 10.0.0.1
remote interface-id unnum 12
remote switching-capability psc1

lmp neighbor gmpls1
ipcc routed
remote node-id 11.11.11.11

rsvp
interface POS0/2/0/3
bandwidth 2000

interface tunnel-gte 1
ipv4 unnumbered Loopback 0
passive
match identifier head_router_hostname_t1
destination 11.11.11.11

Configure Flexible Name-based Tunnel Constraints: Example

The following configuration shows the three-step process used to configure flexible name-based tunnel constraints.

R2
line console
exec-timeout 0 0
width 250

logging console debugging
explicit-path name mypath
index 1 next-address loose ipv4 unicast 3.3.3.3

explicit-path name ex_path1
index 10 next-address loose ipv4 unicast 2.2.2.2 index 20 next-address loose ipv4 unicast 3.3.3.3

interface Loopback0
ipv4 address 22.22.22.22 255.255.255.255

interface tunnel-te1
ipv4 unnumbered Loopback0
signalled-bandwidth 1000000
destination 3.3.3.3
affinity include green
affinity include yellow
affinity exclude white
affinity exclude orange
path-option 1 dynamic

router isis 1
is-type level-1
net 47.0001.0000.0000.0001.0
nsf cisco
address-family ipv4 unicast
metric-style wide
mpls traffic-eng level-1
mpls traffic-eng router-id 192.168.70.1

interface Loopback0
passive
address-family ipv4 unicast

interface GigabitEthernet0/1/0/0
address-family ipv4 unicast

interface GigabitEthernet0/1/0/1
address-family ipv4 unicast

interface GigabitEthernet0/1/0/2
address-family ipv4 unicast

interface GigabitEthernet0/1/0/3
address-family ipv4 unicast

rsvp
interface GigabitEthernet0/1/0/0
bandwidth 1000000 1000000
interface GigabitEthernet0/1/0/1
bandwidth 1000000 1000000
interface GigabitEthernet0/1/0/2
bandwidth 1000000 1000000
interface GigabitEthernet0/1/0/3
bandwidth 1000000 1000000

mpls traffic-eng
interface GigabitEthernet0/1/0/0
attribute-names red purple
interface GigabitEthernet0/1/0/1
attribute-names red orange
interface GigabitEthernet0/1/0/2
attribute-names green purple
interface GigabitEthernet0/1/0/3
attribute-names green orange

affinity-map red 1
affinity-map blue 2
affinity-map black 80
affinity-map green 4
affinity-map white 40
affinity-map orange 20
affinity-map purple 10
affinity-map yellow 8
Configure an Interarea Tunnel: Example

The following configuration example shows how to configure a traffic engineering interarea tunnel. Router R1 is the headend for tunnel1, and router R2 (20.0.0.20) is the tailend. Tunnel1 is configured with a path option that is loosely routed through Ra and Rb.

Note

Specifying the tunnel tailend in the loosely routed path is optional.

```
configure
  interface Tunnel-te1
    ipv4 unnumbered Loopback0
    destination 192.168.20.20
    signalled-bandwidth 300
    path-option 1 explicit name path-tunnel1
    explicit-path name path-tunnel1
    index 10 next-address loose ipv4 unicast 192.168.40.40
    index 20 next-address loose ipv4 unicast 192.168.60.60
    index 30 next-address loose ipv4 unicast 192.168.20.20
```

The following configuration example shows how to configure loose-path retry period (range is 30 to 600 seconds) on headend router.

```
config
  mpls traffic-eng
  timers loose-path retry-period 120
```

The following configuration example shows the global configuration for loose hop expansion affinity or metric on ABR.

```
config
  mpls traffic-eng path-selection loose-expansion affinity 0xff
  mpls traffic-eng path-selection loose-expansion metric te class-type 5
```

Configure Forwarding Adjacency: Example

The following configuration example shows how to configure an MPLS-TE forwarding adjacency on tunnel-te 68 with a holdtime value of 60:

```
configure
  interface tunnel-te 68
  forwarding-adjacency holdtime 60
  commit
```
Related Topics

Configuring MPLS-TE Forwarding Adjacency, on page 203
MPLS-TE Forwarding Adjacency Benefits, on page 127

Configure Unequal Load Balancing: Example

The following configuration example illustrates unequal load balancing configuration:

```
configure
interface tunnel-te0
   destination 1.1.1.1
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
interface tunnel-te1
   destination 1.1.1.1
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   load-share 5
interface tunnel-te2
   destination 1.1.1.1
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 5
interface tunnel-te10
   destination 2.2.2.2
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 10
interface tunnel-te11
   destination 2.2.2.2
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 10
interface tunnel-te12
   destination 2.2.2.2
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 20
interface tunnel-te20
   destination 3.3.3.3
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 10
interface tunnel-te21
   destination 3.3.3.3
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 10
   load-share 20
interface tunnel-te30
   destination 4.4.4.4
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 10
   load-share 5
interface tunnel-te31
   destination 4.4.4.4
   path-option 1 dynamic
   ipv4 unnumbered Loopback0
   signalled-bandwidth 10
   load-share 20
mpls traffic-eng
   load-share unequal
end
```
Configure PCE: Example

The following configuration example illustrates a PCE configuration:

```
configure
  mpls traffic-eng
    interface pos 0/6/0/0
    pce address ipv4 192.168.25.66
    router id loopback 0
    router ospf 1
    router-id 192.168.25.66
    area 0
    interface pos 0/6/0/0
    interface loopback 0
    mpls traffic-eng router-id 192.168.70.1
    mpls traffic-eng area 0
    rsvp
    interface pos 0/6/0/0
    bandwidth 100
    commit
```

The following configuration example illustrates PCC configuration:

```
configure
  interface tunnel-te 10
  ipv4 unnumbered loopback 0
  destination 1.2.3.4
  path-option 1 dynamic pce
  mpls traffic-eng
    interface pos 0/6/0/0
    router id loopback 0
    router ospf 1
    router-id 192.168.25.66
    area 0
    interface pos 0/6/0/0
    interface loopback 0
    mpls traffic-eng router-id 192.168.70.1
    mpls traffic-eng area 0
    rsvp
    interface pos 0/6/0/0
    bandwidth 100
    commit
```

Related Topics

- Configuring a Path Computation Client, on page 206
- Configuring a Path Computation Element Address, on page 207
- Configuring PCE Parameters, on page 208
- Path Computation Element, on page 128
Configure Policy-based Tunnel Selection: Example

The following configuration example illustrates a PBTS configuration:

```plaintext
configure
   interface tunnel-te0
   ipv4 unnumbered Loopback3
   signalled-bandwidth 50000
   autoroute announce
   destination 1.5.177.2
   policy-class 2
   path-option 1 dynamic
```

Related Topics
- Configuring Policy-based Tunnel Selection, on page 211
- Policy-Based Tunnel Selection Functions, on page 130
- Policy-Based Tunnel Selection, on page 130

Configure Tunnels for Path Protection: Example

The path protection feature is configured on only the source router. The dynamic path option is a prerequisite to configure a path protection.

```plaintext
interface tunnel-te150
   ipv4 unnumbered Loopback150
   autoroute announce
   destination 151.151.151.151
   affinity 11 mask 11
   path-protection
   path-option 2 explicit name p2mp3-p2mp4-p2mp5_1
   path-option 10 dynamic
```

Related Topics
- Path Protection, on page 131
- Pre-requisites for Path Protection, on page 132
- Restrictions for Path Protection, on page 132
- Restrictions for Explicit Path Protection, on page 132

Configure Tunnels for Explicit Path Protection: Example

The path protection feature is configured on only the source router. The `protected-by` keyword configures path protection for an explicit path that is protected by another explicit path.

```plaintext
interface tunnel-te150
   ipv4 unnumbered Loopback150
   autoroute announce
   destination 151.151.151.151
   affinity 11 mask 11
   path-protection
   path-option 2 explicit name p2mp3-p2mp4-p2mp5_1 protected-by 10
   path-option 10 explicit
```
Configure Tunnels for Co-existence of Path Protection with Fast Reroute: Example

The path protection feature is configured on only the source router. The dynamic path option is a prerequisite to configure a path protection.

```plaintext
interface tunnel-te1
fast-reroute
ipv4 unnumbered Loopback150
autoroute announce
destination 151.151.151.151
affinity 11 mask 11
path-protection
path-option 2 explicit name p2mp3-p2mp4-p2mp5_1
path-option 10 dynamic
```

Configure Automatic Bandwidth: Example

The following configuration example illustrates an automatic bandwidth configuration:

```plaintext
configure
interface tunnel-te6
auto-bw
bw-limit min 10000 max 500000
overflow threshold 50 min 1000 limit 3
adjustment-threshold 20 min 1000
application 180
```

Related Topics

- Configuring the Collection Frequency, on page 212
- Configuring the Automatic Bandwidth Functions, on page 214
- MPLS-TE Automatic Bandwidth Overview, on page 133

Configure the MPLS-TE Shared Risk Link Groups: Example

The following configuration example shows how to specify the SRLG value of each link that has a shared risk with another link:

```plaintext
config t
srlg
  interface POS0/4/0/0
   value 10
   value 11
  |
  interface POS0/4/0/1
   value 10
  |
```

The following example shows the SRLG values configured on a specific link.

```plaintext
RP/0/RP0/CPU0:router# show mpls traffic-eng topology brief
My_System_id: 100.0.0.2 (OSPF 0 area 0)
```
My_System_id: 0000.0000.0002.00 (IS-IS 1 level-1)
My_System_id: 0000.0000.0002.00 (IS-IS 1 level-2)
My_BC_Model_Type: RDM

Signalling error holddown: 10 sec Global Link Generation 389225

IGP Id: 0000.0000.0002.00, MPLS TE Id: 100.0.0.2 Router Node (IS-IS 1 level-1)
IGP Id: 0000.0000.0002.00, MPLS TE Id: 100.0.0.2 Router Node (IS-IS 1 level-2)

Link[1]:Broadcast, DR:0000.0000.0002.07, Nbr Node Id:21, gen:389193
Frag Id:0, Intf Address:51.2.3.2, Intf Id:0
Nbr Intf Address:51.2.3.2, Nbr Intf Id:0
TE Metric:10, IGP Metric:10, Attribute Flags:0x0
Attribute Names:
SRLGs: 1, 4, 5
Switching Capability:, Encoding:
BC Model ID:RDM
Physical BW:1000000 (kbps), Max Reservable BW Global:10000 (kbps)
Max Reservable BW Sub:10000 (kbps)

The following example shows the configured tunnels and associated SRLG values.

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels
<snip>
Signalling Summary:
  LSP Tunnels Process: running
  RSVP Process: running
  Forwarding: enabled
  Periodic reoptimization: every 3600 seconds, next in 1363 seconds
  Periodic FRR Promotion: every 300 seconds, next in 181 seconds
  Auto-bw enabled tunnels: 0 (disabled)

Name: tunnel-te1 Destination: 100.0.0.3
Status:
  Admin: up Oper: up Path: valid Signalling: recovered
  path option 1, type explicit path123 (Basis for Setup, path weight 2)
  OSPF 0 area 0
  G-PID: 0x0800 (derived from egress interface properties)
  SRLGs excluded: 2,3,4,5
  6,7,8,9
  Bandwidth Requested: 0 kbps CT0
<snip>

The following example shows all the interfaces associated with SRLG.

RP/0/RP0/CPU0:router# show mpls traffic-eng topo srlg
My_System_id: 100.0.0.5 (OSPF 0 area 0)
My_System_id: 0000.0000.0005.00 (IS-IS 1 level-2)
My_System_id: 0000.0000.0005.00 (IS-IS ISIS-instance-123 level-2)

<table>
<thead>
<tr>
<th>SRLG</th>
<th>Interface Addr</th>
<th>TE Router ID</th>
<th>IGP Area ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
<tr>
<td>11</td>
<td>50.2.3.3</td>
<td>100.0.0.3</td>
<td>IS-IS 1 level-2</td>
</tr>
<tr>
<td>12</td>
<td>50.2.3.3</td>
<td>100.0.0.3</td>
<td>IS-IS 1 level-2</td>
</tr>
<tr>
<td>30</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
<tr>
<td>77</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
<tr>
<td>88</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
<tr>
<td>1500</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
<tr>
<td>10000000</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
<tr>
<td>4294967290</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
<tr>
<td>4294967295</td>
<td>50.4.5.5</td>
<td>100.0.0.5</td>
<td>IS-IS ISIS-instance-123 level-2</td>
</tr>
</tbody>
</table>
The following examples show the NHOP and NNHOP backup tunnels with excluded SRLG values.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng topology path dest 100.0.0.5 exclude-srlg ipaddr
Path Setup to 100.0.0.2:
bw 0 (CT0), min_bw 0, metric: 30
setup_pri 7, hold_pri 7
affinity_bits 0x0, affinity_mask 0xffff
Exclude SRLG Intf Addr : 50.4.5.5
SRLGs Excluded : 10, 30, 1500, 10000000, 4294967290, 4294967295
Hop0:50.5.1.5
Hop1:50.1.3.1
Hop2:50.1.3.3
Hop4:50.2.3.3
Hop5:50.2.3.2
Hop6:100.0.0.2
```

The following example shows an extract of explicit-path set to protect a specific interface.

```
RP/0/RP0/CPU0:router#sh mpls traffic-eng topology path dest 10.0.0.5 explicit-path name
Path Setup to 100.0.0.5:
bw 0 (CT0), min_bw 9999, metric: 2
setup_pri 7, hold_pri 7
affinity_bits 0x0, affinity_mask 0xffff
SRLGs Excluded: 10, 30, 77, 88, 1500, 10000000, 4294967290, 4294967295
Hop0:50.3.4.3
Hop1:50.3.4.4
Hop2:50.4.5.4
Hop3:50.4.5.5
Hop4:100.0.0.5
```

Related Topics
- Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 217
- Creating an Explicit Path With Exclude SRLG, on page 219
- Using Explicit Path With Exclude SRLG, on page 220
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 222
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 225
- MPLS Traffic Engineering Shared Risk Link Groups, on page 141
- Explicit Path, on page 142
- Fast ReRoute with SRLG Constraints, on page 142
- Importance of Protection, on page 144
- Delivery of Packets During a Failure, on page 145
- Multiple Backup Tunnels Protecting the Same Interface, on page 145
- SRLG Limitations, on page 145

Configure the MPLS-TE Auto-Tunnel Backup: Example

The following example shows the auto-tunnel backup configuration for core or edge routers.

```
RP/0/RP0/CPU0:router(config)#
mpls traffic-eng
```
auto-tunnel backup
tunnel-id min 60000 max 61000
interface pos 0/1/0/0
auto-tunnel backup
attribute-set ab

The following example shows the protection (NNHOP and SRLG) that was set on the auto-tunnel backup.

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 1
Signing Summary:
 LSP Tunnels Process: running
 RSVP Process: running
 Forwarding: enabled
 Periodic reoptimization: every 3600 seconds, next in 2524 seconds
 Periodic FRR Promotion: every 300 seconds, next in 49 seconds
 Auto-bw enabled tunnels: 1

Name: tunnel-te1 Destination: 200.0.0.3 (auto backup)
 Status: 
 admin: up
 oper: up
 path: valid
 signalling: connected

 path option 10, type explicit (autob_nnhop_srlg_tunnel1) (Basis for Setup, path weight 11)
 path option 20, type explicit (autob_nnhop_tunnel1)
   G-PID: 0x0800 (derived from egress interface properties)
   Bandwidth Requested: 0 kbps
   Creation Time: Fri Jul 10 01:53:25.581 PST (1h 25m 17s ago)

Config Parameters:
 Bandwidth: 0 kbps (CT0)
 Priority: 7
 Affinity: 0x0/0xffff
 Metric Type: TE (default)
 AutoRoute: disabled
 LockDown: disabled
 Policy class: not set
 Forwarding-Adjacency: disabled
 Loadshare: 0 equal loadshares
 Auto-bw: disabled
 Fast Reroute: Disabled
 Protection Desired: None
 Path Protection: Not Enabled
 Auto Backup:
 Protected LSFs: 4
 Protected S2L Sharing Families: 0
 Protected S2Ls: 0
 Protected i/f: G10/1/0/0
 Protected node: 20.0.0.2
 Protection: NNHOP+SRLG
 Unused removal timeout: not running

History:
 Tunnel has been up for: 00:00:08
 Current LSP:
   Uptime: 00:00:08
 Prior LSP:
   ID: path option 1 [545]
   Removal Trigger: configuration changed

Path info (OSPF 0 area 0):
 Hop0: 10.0.0.2
 Hop1: 100.0.0.2
 Hop2: 100.0.0.3
 Hop3: 200.0.0.3

The following example shows automatically created path options for this backup auto-tunnel.

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels 1 detail
Signing Summary:
 LSP Tunnels Process: running
 RSVP Process: running
 Forwarding: enabled
 Periodic reoptimization: every 3600 seconds, next in 2524 seconds
 Periodic FRR Promotion: every 300 seconds, next in 49 seconds
Auto-bw enabled tunnels: 1

Name: tunnel-te1  Destination: 200.0.0.3 (auto backup)
Status:
  Admin: up  Oper: up  Path: valid  Signalling: connected
  path option 10, type explicit (autob_nnhop_srlg_tunnel1) (Basis for Setup, path weight 11)
  path option 20, type explicit (autob_nnhop_tunnel1)
G-PID: 0x0800 (derived from egress interface properties)
Bandwidth Requested: 0 kbps  CT0
Creation Time: Fri Jul 10 01:53:25.581 PST (1h 25m 17s ago)

Config Parameters:
  Bandwidth: 0 kbps (CT0)  Priority: 7 7  Affinity: 0x0/0xffffffff
  Metric Type: TE (default)
  AutoRoute: disabled  LockDown: disabled  Policy class: not set
  Forwarding-Adjacency: disabled
  Loadshare: 0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: disabled, Protection Desired: None
  Path Protection: Not Enabled

Auto Backup (NNHOP+SRLG):
  Protected LSPs: 4
  Protected S2L Sharing Families: 0
  Protected S2Ls: 0
  Protected i/f: G10/1/0/0  Protected node: 20.0.0.2
  Protection: NNHOP+SRLG
  Unused removal timeout: not running

Path Options Details:
  10: Explicit Path Name: (autob_nnhop_srlg_te1)
    1: exclude-srlg 50.0.0.1
    2: exclude-address 50.0.0.2
    3: exclude-node 20.0.0.2
  20: Explicit Path Name: (autob_nnhop_te1)
    1: exclude-address 50.0.0.1
    2: exclude-address 50.0.0.2
    3: exclude-node 20.0.0.2

History:
  Tunnel has been up for: 00:00:08
  Current LSP:
    Uptime: 00:00:08
  Prior LSP:
    ID: path option 1 [545]
      Removal Trigger: configuration changed

Path info (OSPF 0 area 0):
  Hop0: 10.0.0.2
  Hop1: 100.0.0.2
  Hop2: 100.0.0.3
  Hop3: 200.0.0.3

This example shows the automatically created backup tunnels.

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

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>LSP</th>
<th>Destination</th>
<th>Source</th>
<th>FRR</th>
<th>LSP</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>200.0.0.3</td>
<td>200.0.0.1</td>
<td>up</td>
<td>Inact</td>
<td>Head</td>
</tr>
</tbody>
</table>
This example shows the auto-tunnel backup details.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-tunnel backup detail

Name: tunnel-te400 Destination: 1.1.1.1 (auto-tunnel backup)
Status:
  Admin: up Oper: up Path: valid Signalling: connected

  path option 20, type explicit (autob_nnhop_te400) (Basis for Setup, path weight 2)
  path option 10, type explicit (autob_nnhop_srlg_te400) [disabled]
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 0 kbps CT0
  Creation Time: Thu Aug 16 18:30:41 2012 (00:01:28 ago)

Config Parameters:
  Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
  Metric Type: TE (default)
  Hop-limit: disabled
  AutoRoute: disabled LockDown: disabled Policy class: not set
  Forwarding-Adjacency: disabled
  Loadshare: 0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Disabled, Protection Desired: None
  Path Protection: Not Enabled
  Soft Preemption: Disabled

Auto Backup:
  Protected LSPs: 1
  Protected S2L Sharing Families: 0
  Protected S2L: 0
  Protected i/f: Gi0/1/0/3 Protected node: 3.3.3.3
  Attribute-set: ab1
  Protection: NNHOP
  Unused removal timeout: not running

Path Option Details:
  10: Explicit Path Name: (autob_nnhop_srlg_te400)
    1: exclude-srlg 34.9.0.4
    2: exclude-address 34.9.0.3
    3: exclude-node 3.3.3.3
  20: Explicit Path Name: (autob_nnhop_te400)
    1: exclude-address 34.9.0.4
    2: exclude-address 34.9.0.3
    3: exclude-node 3.3.3.3

SNMP Index: 221
History:
  Tunnel has been up for: 00:00:34 (since Thu Aug 16 18:31:35 EST 2012)
  Current LSP:
    Uptime: 00:00:34 (since Thu Aug 16 18:31:35 EST 2012)
  Current LSP Info:
    Instance: 2, Signaling Area: OSPF 100 area 1.2.3.4
    Uptime: 00:00:34 (since Thu Aug 16 18:31:35 EST 2012)
    Outgoing Interface: GigabitEthernet0/1/0/2, Outgoing Label: 16000
    Router-IDs: local 4.4.4.4
    downstream 2.2.2.2
    Soft Preemption: None
  Path Info:
    Outgoing:
      Explicit Route:
        Strict, 24.9.0.2
        Strict, 12.9.1.1
        Strict, 1.1.1.1
    Record Route: Empty
    Tspec: avg rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
    Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
    Soft Preemption Desired: Not Set

Resv Info:
```
This example shows the automatically created backup tunnels.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels auto-tunnel backup tabular

    Tunnel Name   LSP ID Destination Address Source Address State  Tun FRR LSP Path
----------------- ----- --------------- --------------- ------ ------ ---- ----
    *tunnel-te400    2       1.1.1.1       4.4.4.4      up   Inact  Head  Inact
    *tunnel-te401    2       3.3.3.3       4.4.4.4      up   Inact  Head  Inact

* = automatically created backup tunnel
```

This example shows the attribute-set for auto-backup tunnels.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng attribute-set auto-backup

Attribute Set Name: ab (Type: auto-backup)
Number of affinity constraints: 2
  Include bit map : 0x4
  Include name : blue
  Exclude bit map : 0x2
  Exclude name : red
Priority: 7 7 (Default)
Record-route: Enabled
Policy-class: 1
Logging: reoptimize, state
  List of protected interfaces (count 1)
    POSO_3_0_1
  List of tunnel IDs (count 1)
    3000
```

This example shows the attribute-set for auto-mesh tunnels.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng attribute-set auto-mesh

Attribute Set Name: am (Type: auto-mesh)
Bandwidth: 100 kbps (CT0)
Number of affinity constraints: 2
  Include bit map : 0x8
  Include name : yellow
  Exclude bit map : 0x2
  Exclude name : red
Priority: 2 2
Interface Bandwidth: 0 kbps (Default)
AutoRoute Announce: Disabled
Auto-tw: Disabled
Soft Preemption: Disabled
Fast Reroute: Enabled, Protection Desired: Node, Bandwidth
Record-route: Enabled
Policy-class: 0 (Not configured)
Logging: None
  List of Mesh Groups (count 1)
    1
```

This example shows the details about the tunnel that is using auto-backup type of attribute-set.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels attribute-set auto-backup ab
```
Name: tunnel-te3000  Destination: 1.1.1.1 (auto-tunnel backup)
Status:  
  Admin: up  Oper: up  Path: valid  Signalling: connected  
  path option 20, type explicit (autob_nhop_te3000) (Basis for Setup, path weight 2)  
  path option 10, type explicit (autob_nhop_srlg_te3000) [disabled]  
  G-FID: 0x0800 (derived from egress interface properties)  
  Bandwidth Requested: 0 kbps  CT0  
  Creation Time: Tue Aug 14 23:24:27 2012 (00:05:28 ago)  
Config Parameters:  
  Bandwidth: 0 kbps (CT0)  Priority: 7 7  
  Number of affinity constraints: 2  
  Include bit map : 0x4  
  Include name : blue  
  Exclude bit map : 0x2  
  Exclude name : red  
  Metric Type: TE (default)  
  Hop-limit: disabled  
  AutoRoute: disabled  LockDown: disabled  Policy class: 1  
  Forwarding-Adjacency: disabled  
  Loadshare: 0 equal loadshares  
  Auto-bw: disabled  
  Fast Reroute: Disabled, Protection Desired: None  
  Path Protection: Not Enabled  
  Soft Preemption: Disabled  
  Auto Backup:  
    Protected LSPs: 2  
    Protected S2L Sharing Families: 0  
    Protected S2L: 0  
    Protected i/f: PO0/3/0/1  
    Attribute-set: ab  
    Protection: NHOP  
    Unused removal timeout: not running  
History:  
  Tunnel has been up for: 00:04:57 (since Tue Aug 14 23:24:58 EST 2012)  
  Current LSP:  
    Uptime: 00:04:57 (since Tue Aug 14 23:24:58 EST 2012)  
Path info (OSPF 100 area 16909060):  
  Node hop count: 2  
  Hop0: 23.9.0.2  
  Hop1: 12.9.0.2  
  Hop2: 12.9.0.1  
  Hop3: 1.1.1.1  
Displayed 1 (of 7) heads, 0 (of 3) midpoints, 0 (of 0) tails  
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads  

This example shows the protected interface for auto-backup auto-tunnels.

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels backup protected-interface  
Interface: Gi0/2/0/1 (auto-tunnel backup)  
  SRLG: N/A, NHOP-only: No  
  Attribute-set: Not configured  
  Auto-tunnel backup recreate time remaining: timer not running  
  No backup tunnel found  
Interface: Gi0/2/0/3  
  tunnel-te340  PROTECTED : out i/f: PO0/3/0/2  Admin: up  Oper: up  
Interface: PO0/3/0/1 (auto-tunnel backup)  
  SRLG: N/A, NHOP-only: No  
  Attribute-set: ab  
  Auto-tunnel backup recreate time remaining: timer not running  
  *tunnel-te3000  NHOP : out i/f: Gi0/2/0/2  Admin: up  Oper: up  
* = automatically created backup tunnel

This example shows the details about all the tunnels that are using auto-mesh type of attribute-set.

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels attribute-set auto-mesh all  
Name: tunnel-te3501  Destination: 1.1.1.1 (auto-tunnel mesh)
Status:
Admin: up  Oper: up  Path: valid  Signalling: connected
path option 10, type dynamic (Basis for Setup, path weight 2)
G-PID: 0x0800 (derived from egress interface properties)
Bandwidth Requested: 100 kbps  CT0
Creation Time: Tue Aug 14 23:25:41 2012 (00:06:13 ago)
Config Parameters:
  Bandwidth:  100 kbps (CT0)  Priority:  2  2
  Number of affinity constraints: 2
    Include bit map  : 0x8
    Include name  : yellow
    Exclude bit map  : 0x2
    Exclude name  : red

Metric Type: TE (default)
Hop-limit: disabled
AutoRoute: disabled  LockDown: disabled  Policy class: not set
Forwarding-Adjacency: disabled
Loadshare:  0 equal loadshares
Auto-bw: disabled
Fast Reroute: Enabled, Protection Desired: Node, Bandwidth
Path Protection: Not Enabled
Attribute-set: am (type auto-mesh)
Soft Preemption: Disabled

Auto-tunnel Mesh:
  Group ID: 1
  Destination list: blah
  Unused removal timeout: not running
History:
  Tunnel has been up for: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)

Current LSP:
  Uptime: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)

Path info (OSPF 100 area 16909060):
Node hop count: 2
Hop0: 23.9.0.2
Hop1: 12.9.0.2
Hop2: 12.9.0.1
Hop3: 1.1.1.1
Name: tunnel-te3502  Destination: 2.2.2.2 (auto-tunnel mesh)
Status:
Admin: up  Oper: up  Path: valid  Signalling: connected
path option 10, type dynamic (Basis for Setup, path weight 1)
G-PID: 0x0800 (derived from egress interface properties)
Bandwidth Requested: 100 kbps  CT0
Creation Time: Tue Aug 14 23:25:41 2012 (00:06:13 ago)
Config Parameters:
  Bandwidth:  100 kbps (CT0)  Priority:  2  2
  Number of affinity constraints: 2
    Include bit map  : 0x8
    Include name  : yellow
    Exclude bit map  : 0x2
    Exclude name  : red

Metric Type: TE (default)
Hop-limit: disabled
AutoRoute: disabled  LockDown: disabled  Policy class: not set
Forwarding-Adjacency: disabled
Loadshare:  0 equal loadshares
Auto-bw: disabled
Fast Reroute: Enabled, Protection Desired: Node, Bandwidth
Path Protection: Not Enabled
Attribute-set: am (type auto-mesh)
Soft Preemption: Disabled

Auto-tunnel Mesh:
  Group ID: 1
  Destination list: blah
  Unused removal timeout: not running
History:
  Tunnel has been up for: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)
Configure Point-to-Multipoint TE: Examples

These configuration examples show how to configure Point-to-Multipoint TE:

Configure Point-to-Multipoint for the Source: Example

At the source, multicast routing must be enabled on both the tunnel-mte interface and customer-facing interface. Then, the static-group must be configured on the tunnel-mte interface to forward specified multicast traffic over P2MP LSP.
The multicast group address, which is in Source-Specific Multicast (SSM) address range (ff35::/16), must be used on the static-group configuration because Cisco IOS XR software supports only SSM for Label Switch Multicast (LSM). Additionally, the customer-facing interface must have an IPv6 address.

```
multicast-routing
  address-family ipv6
  interface tunnel-mte 1
    enable
    !
    interface GigabitEthernet0/2/0/3
      enable
      !
    !
  router mld
    vrf default
    interface tunnel-mte 1
      static-group ff35::1 2000::1 3eFF::A
      !
      !
    interface tunnel-mte 1
      ipv4 unnumbered Loopback0
      destination 3.3.3.3
      path-option 1 dynamic
      destination 4.4.4.4
      path-option 1 dynamic
      !
      !
```

### Related Topics
- Enabling Multicast Routing on the Router, on page 228
- Point-to-Multipoint Traffic-Engineering Overview, on page 136
- Configuring the Static Group for the Point-to-Multipoint Interface, on page 230
- Point-to-Multipoint RSVP-TE, on page 138

### Configure the Point-to-Multipoint Tunnel: Example

There is no difference between logging events at the tunnel level for both P2P and P2MP. The P2MP tunnel reoptimizes only at the per tunnel level.

```
interface tunnel-mte1
  ipv4 unnumbered Loopback0
  destination 60.60.60.60
  logging events lsp-status state
  logging events lsp-status reroute
  path-option 10 explicit name toR6_via_R2andR3
  !
  logging events lsp-status reoptimize
  logging events lsp-status state
  logging events lsp-status reroute
  fast-reroute
  record-route
  !
  explicit-path name PATH7
    index 1 next-address strict ipv4 unicast 192.168.7.2
    index 2 next-address strict ipv4 unicast 192.168.7.1
    index 3 next-address strict ipv4 unicast 192.168.16.1
```
Disable a Destination: Example

From the tunnel-mte interface, you can disable the destination.

    interface tunnel-mte101
    ipv4 unnumbered Loopback0
    destination 150.150.150.150
    disable
    path-option 10 dynamic
    *
    destination 150.150.150.150
    path-option 2 dynamic
    *

Related Topics

Disabling Destinations, on page 234
Point-to-Multipoint Traffic-Engineering Overview, on page 136

Configure the Point-to-Multipoint Solution: Example

Requirements for MPLS-TE Configuration

Before the Point-to-Multipoint (P2MP) tunnel is defined, these MPLS-TE requirements must be configured:

- Multiprotocol Label Switching traffic engineering (MPLS-TE)
- Resource ReSerVation Protocol (RSVP)
- Open Shortest Path First (OSPF)

This example shows the entire P2MP solution:

- Source is the location where the P2MP-TE tunnel interface is created.
- Tunnel contains multiple destinations. For example, the P2MP-TE tunnel is configured with two leaf node destinations by using the dynamic and explicit path options.
- Fast-Reroute (FRR) is specified on the P2MP tunnel.
- All regular TE tunnel options such as affinity or bandwidth are configured.
- Static mapping of the group address to the P2MP tunnel is done in IGMP.
  - Internet Group Management Protocol (IGMP).
- The P2MP-TE midpoint configuration requires only TE and Interior Gateway Protocol (IGP) information.
The P2MP-TE receiver configuration requires a static group and RPF map.

```
! explicit-path name g2-r2-r1
  index 1 next-address strict ipv4 unicast 10.2.15.1
! explicit-path name g2-r2-r3
  index 1 next-address strict ipv4 unicast 10.2.25.1
  index 2 next-address strict ipv4 unicast 10.2.23.2
! explicit-path name g2-r2-r4
  index 1 next-address strict ipv4 unicast 10.2.25.1
  index 2 next-address strict ipv4 unicast 10.2.24.2
! ipv4 access-list ssm
  10 permit ipv4 232.1.0.0/16 any
  20 permit ipv4 232.3.0.0/16 any
  30 permit ipv4 232.4.0.0/16 any
! ipv4 access-list ssm-test
  10 permit ipv4 235.0.0.0/8 any
!
interface Loopback0
  ipv4 address 192.168.1.2 255.255.255.255
! interface tunnel-mte221
  ipv4 unnumbered Loopback0
  destination 192.168.1.1
  path-option 1 dynamic
!  destination 192.168.1.3
  path-option 1 dynamic
!  destination 192.168.1.4
  path-option 1 dynamic
!
interface tunnel-mte222
  ipv4 unnumbered Loopback0
  destination 192.168.1.1
  path-option 1 explicit name g2-r2-r1
!  destination 192.168.1.3
  path-option 1 explicit name g2-r2-r3
!  destination 192.168.1.4
  path-option 1 explicit name g2-r2-r4
!  signalled-bandwidth 1000
!
interface MgmtEth0/RP0/CPU0/0
  ipv4 address 172.20.163.12 255.255.255.128
!
interface MgmtEth0/RP1/CPU0/0
  shutdown
!
interface GigabitEthernet0/0/0/0
  ipv4 address 172.2.1.2 255.255.255.0
  load-interval 30
!
interface GigabitEthernet0/0/0/1
  ipv4 address 10.1.15.2 255.255.255.0
!
interface GigabitEthernet0/0/1.2
  ipv4 address 10.2.15.2 255.255.255.0
  encapsulation dot1q 2
!
interface GigabitEthernet0/0/0/2
  ipv4 address 10.1.25.2 255.255.255.0
!
interface GigabitEthernet0/0/0/2.2
```
ipv4 address 10.2.25.2 255.255.255.0
encapsulation dot1q 2
!
interface GigabitEthernet0/0/0/3
shutdown
!
interface GigabitEthernet0/0/0/4
shutdown
!
interface GigabitEthernet0/0/0/5
shutdown
!
interface GigabitEthernet0/0/0/6
shutdown
!
interface GigabitEthernet0/0/0/7
shutdown
!
router static
address-family ipv4 unicast
 0.0.0.0/0 1.56.0.1
 0.0.0.0/0 172.20.163.1
!
!
router ospf 100
nsr
router-id 192.168.70.1
area 0
mpls traffic-eng
interface Loopback0
!
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/1.2
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/2.2
!
!
mpls traffic-eng router-id 192.168.70.1
mpls oam
!
rsvp
interface GigabitEthernet0/0/0/0
  bandwidth 20000
!
interface GigabitEthernet0/0/0/1
  bandwidth 20000
!
interface GigabitEthernet0/0/0/2
  bandwidth 20000
!
interface GigabitEthernet0/0/0/1.2
  bandwidth 20000
!
interface GigabitEthernet0/0/0/2.2
  bandwidth 20000
!
!
mpls traffic-eng
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/2
!
interface GigabitEthernet0/0/0/1.2
!
interface GigabitEthernet0/0/0/2.2
mpls ldp
  router-id 192.168.1.2
nsr
graceful-restart
interface GigabitEthernet0/0/0/0
  !
interface GigabitEthernet0/0/0/1
  !
interface GigabitEthernet0/0/0/1.2
  !
interface GigabitEthernet0/0/0/2
  !
interface GigabitEthernet0/0/0/2.2
  !
  multicast-routing
  address-family ipv4
    core-tree-protocol rsvp-te
    ssm range ssm
    static-rpf 172.1.1.1 32 mpls 192.168.1.1
    static-rpf 172.3.1.1 32 mpls 192.168.1.3
    static-rpf 172.4.1.1 32 mpls 192.168.1.4
    interface all enable
  !
  router igmp
  !
  interface tunnel-mte221
    static-group 232.2.2.1 172.2.1.1
  !
  interface tunnel-mte222
    static-group 232.2.2.2 172.2.1.1
  !
  interface GigabitEthernet0/0/0/0
    static-group 232.2.2.1 172.2.1.1
    static-group 232.1.2.1 172.1.1.1
    static-group 232.1.2.2 172.1.1.1
    static-group 232.3.2.1 172.3.1.1
    static-group 232.3.2.2 172.3.1.1
    static-group 232.4.2.1 172.4.1.1
    static-group 232.4.2.2 172.4.1.1
  !
end

Related Topics

  Enabling Multicast Routing on the Router, on page 228
  Point-to-Multipoint Traffic-Engineering Overview, on page 136
  Configuring the Static Group for the Point-to-Multipoint Interface, on page 230
  Point-to-Multipoint RSVP-TE, on page 138
  Configuring Destinations for the Tunnel Interface, on page 231
  Path Option for Point-to-Multipoint RSVP-TE, on page 139
  Logging Per Destinations for Point-to-Multipoint, on page 236
  Point-to-Multipoint Traffic-Engineering Overview, on page 136

Additional References

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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS-TE commands</td>
<td><em>MPLS Traffic Engineering Commands</em> module in <em>Cisco IOS XR MPLS Command Reference for the Cisco CRS Router.</em></td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
</table>
(Format: TXT=79265 bytes) (Status: PROPOSED STANDARD) |
(Format: TXT=22585 bytes) (Status: EXPERIMENTAL) |
<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Format: TXT=23694 bytes) (Status: EXPERIMENTAL)</td>
</tr>
</tbody>
</table>

**Technical Assistance**

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
Implementing GMPLS UNI

The Generalized Multiprotocol Label Switching (GMPLS) User Network Interface (UNI) creates a circuit connection between two clients (UNI-C) of an optical network. This connection is achieved by signaling exchanges between UNI Client (UNI-C) and UNI Network (UNI-N) nodes, where UNI-C nodes are router nodes and UNI-N nodes are optical nodes.

A GMPLS overlay model is required to connect packet routers with the optical network in these scenarios:

• Different groups within a service provider are responsible for managing packet and optical networks.
• The optical and packet network are managed by different service providers.
• There is a weak trust model between the entities operating the optical and packet networks.

Feature History for Implementing GMPLS UNI

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.3.0</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

Prerequisites for Implementing GMPLS UNI

The following prerequisites are required to implement GMPLS UNI:

• Prerequisites for Implementing GMPLS UNI, page 285
• Restrictions for Implementing GMPLS UNI, page 286
• Information About Implementing GMPLS UNI, page 286
• How to Implement GMPLS UNI, page 288
• Configuration Examples for GMPLS UNI, page 300
• Additional References, page 302
You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- Router that runs Cisco IOS XR software.
- Installation of the Cisco IOS XR software mini-image on the router.
- Installation of the Cisco IOS XR MPLS software package on the router.

Restrictions for Implementing GMPLS UNI

- The total number of configured GMPLS UNI controllers should not exceed the platform scale limit of 500 GMPLS interfaces.
- Each UNI-N (ingress or egress) should be routable from its adjacent UNI-C. The UNI-C nodes need to be routable from the UNI-N nodes too.
- GMPLS UNI is supported only over DWDM controllers and so, over POS and GigabitEthernet interfaces.
- GMPLS UNI is supported only with these Cisco CRS Line Cards:
  - Cisco CRS-1 OC768 (C/L-band) DWDM PLIM
  - Cisco CRS-1 OC768 DPSK (C/L-band) STD CHAN PLIM
  - Cisco CRS-1 4 port 10GE (C/L-band) DWDM PLIM
  - Cisco CRS 1-Port 100GE CP-DQSPK Full C-Band Tunable DWDM Interface Module

Information About Implementing GMPLS UNI

To implement GMPLS UNI, you should understand these concepts:

GMPLS UNI vs GMPLS NNI

In case of GMPLS NNI, the optical network topology is known and path calculations are performed at the NNI head. In case of GMPLS UNI, the optical network topology is unknown to the UNI-C nodes and path calculations are performed by the UNI-N nodes.

GMPLS LSP Signaling

The GMPLS overlay model architecture is used for LSP signaling for GMPLS connections. In GMPLS UNI, UNI-C nodes send a request for a connection to UNI-N node. The connection request does not contain an end-to-end path. This is because, as mentioned previously, UNI-C nodes do not have knowledge of the topology of the optical network and therefore cannot determine the end-to-end path. The UNI-C node signals a connection request without an ERO.

The LSP diversity is signalled on a GMPLS UNI tunnel with a path-option. A path-option is permitted on a GMPLS UNI tunnel with a "no ERO" and an optional "XRO" attribute sets to specify LSP diversity.
requirements. If multiple LSP exclusions are configured in the attribute-set, they can be added to the path message along with an appropriate LSP connection diversity sub-object.

**Path Message without an ERO**

In GMPLS UNI, UNI-C nodes send a request for a connection to UNI-N node. The connection request does not contain an end-to-end path, because, UNI-C nodes do not have knowledge of the topology of the optical network and therefore cannot determine the end-to-end path. The UNI-C node signals a connection request without an ERO.

When no ERO is present in a received path message, the UNI-N node calculates a route to the destination and includes that route in an ERO, before forwarding the path message. If no route is found, the UNI-N returns a path error message with an error code and subcode of 24,5 - "No route available toward destination".

The destination address of a GMPLS LSP can be either the optical router-id of the tail UNI-C node, or the optical address of the ingress interface to the tail UNI-C node. Supplying the router-id allows the UNI-N to route the tunnel to the tail UNI-C node via any attached UNI-N node; supplying the UNI-C's ingress interface address forces the tunnel's path to traverse the UNI-N node attached to that interface.

**XRO Attribute-set**

An optional XRO attribute-set can be specified as part of the path-option to specify LSP diversity requirements. An empty XRO attribute set results in the GMPLS tunnel being signaled with no exclusions, and therefore no XRO.

A non-existent XRO attribute-set can be configured in the GMPLS UNI tunnel path-option; in this case no attempt will be made to bring up the GMPLS tunnel until the configuration is complete.

**Connection Diversity**

Connection diversity is required to ensure that GMPLS tunnels can be established without sharing resources, thus, greatly reducing the probability of simultaneous connection failures. For example, an edge-node wishes to establish multiple LSPs towards the same destination edge-node, and these LSPs need to have few or no resources in common.

Connection diversity supports the establishment of a GMPLS LSP which is diverse from the path taken by an existing LSP. An XRO is added to the tunnel's path message with appropriate LSP diversity sub-objects or exclusions. A maximum of 20 connection diversity exclusions per XRO is supported.

**DWDM Transponder Integration**

A GMPLS UNI based solution preserves all the advantages of the integration of the DWDM transponder into the router blade. These advantages include:
• improved CAPEX and OPEX models
• component, space and power savings
• improved IP availability through pro-active protection.

How to Implement GMPLS UNI

A new submode is introduced under the main TE submode to enable GMPLS UNI and to contain GMPLS UNI configuration.

To implement GMPLS UNI, follow these procedures:

Configuring TE for GMPLS UNI

TE configuration specific to packet tunnels does not affect GMPLS UNI tunnels.

To implement TE configuration for GMPLS UNI, follow these procedures:

Enabling GMPLS UNI Submode

Perform this task to enable GMPLS UNI configuration submode and to configure GMPLS UNI tunnels.

Note
Removal of the GMPLS UNI submode results in the removal of all configuration within it, including any other parser submode, and the immediate destruction of all GMPLS UNI tunnels.

SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. gmpls optical-uni
4. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CP00:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 3</th>
<th>gmpls optical-uni</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# <code>gmpls optical-uni</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls)#</td>
</tr>
</tbody>
</table>

| Step 4 | commit |

### Configuring GMPLS UNI Controller

Perform this task to setup a GMPLS tail in MPLS-TE configuration. This task enables GMPLS UNI controller submode to configure controllers for establishing GMPLS UNI tunnels. This is the minimal configuration required at the tunnel tail.

**Note**

Removal of the GMPLS UNI controller submode results in the immediate destruction of any GMPLS tunnel established over the controller referenced.

### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. gmpls optical-uni
4. controller dwdm `interface`
5. commit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# <code>mpls traffic-eng</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>gmpls optical-uni</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# <code>gmpls optical-uni</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring GMPLS UNI Controller as a Tunnel Head

Perform this task to configure the tunnel properties for a GMPLS UNI controller.

This configuration designates the controller as a tunnel-head, rather than a tunnel tail. After the tunnel properties are configured, the incoming path messages are rejected and any existing tail-end tunnel is torn down.

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. mpls optical-uni
4. controller dwdm interface
5. tunnel-properties
6. tunnel-id number
7. destination ipv4 unicast address
8. path-option 10 no-ero lockdown
9. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
<td>Enters GMPLS UNI controller submode.</td>
</tr>
</tbody>
</table>

Example:

```text
RP/0/RP0/CPU0:router(config-te-gmpls)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:router(config-te-gmpls-cntl)#
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> gmpls optical-uni</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# gmpls</td>
</tr>
<tr>
<td></td>
<td>optical-uni</td>
</tr>
<tr>
<td><strong>Step 4</strong> controller dwdm interface</td>
<td>Enters GMPLS UNI controller submode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls)#</td>
</tr>
<tr>
<td></td>
<td>controller dwdm 0/1/0/1</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-cntl)#</td>
</tr>
<tr>
<td><strong>Step 5</strong> tunnel-properties</td>
<td>Enters the submode to configure tunnel-specific information for a GMPLS UNI controller.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-cntl)#</td>
</tr>
<tr>
<td></td>
<td>tunnel-properties</td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-tun)#</td>
</tr>
<tr>
<td><strong>Step 6</strong> tunnel-id number</td>
<td>Specifies a tunnel-id for a headend router of a GMPLS tunnel. The tunnel-id is a 16-bit number ranging from 0 to 65535.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-tun)#</td>
</tr>
<tr>
<td></td>
<td>tunnel-id 100</td>
</tr>
<tr>
<td><strong>Step 7</strong> destination ipv4 unicast address</td>
<td>Specifies a tunnel destination for a headend router of a GMPLS tunnel. The destination argument is an IPv4 address.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-tun)#</td>
</tr>
<tr>
<td></td>
<td>destination ipv4 unicast 10.10.3.4</td>
</tr>
<tr>
<td><strong>Step 8</strong> path-option 10 no-ero lockdown</td>
<td>Specifies the path-option for a headend router of a GMPLS tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-tun)#</td>
</tr>
<tr>
<td></td>
<td>path-option 10 no-ero lockdown</td>
</tr>
<tr>
<td><strong>Step 9</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Other Tunnel Properties for a GMPLS UNI Tunnel

Perform this task to configure the optional tunnel properties for a GMPLS UNI tunnel. This configuration is optional, and if omitted, the GMPLS tunnel is established with the default property values.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. gmpls optical-uni
4. controller dwdm interface
5. tunnel-properties
6. priority setup-priority hold-priority
7. record-route
8. signalled-name name
9. logging events lsp-status state
10. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config) # mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> gmpls optical-uni</td>
<td>Enters GMPLS UNI controller submode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te) # gmpls optical-uni</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> controller dwdm interface</td>
<td>Enters the submode to configure tunnel-specific information for a GMPLS UNI controller.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-te-gmpls-cntl) # controller dwdm 0/1/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> tunnel-properties</td>
<td>Specifies the priority for a GMPLS tunnel. The default priority value is 7 for both setup and hold priorities.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-te-gmpls-cntl) # tunnel-properties</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> priority setup-priority hold-priority</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-te-gmpls-tun) # priority 3 2</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring TE for GMPLS UNI

#### Command or Action | Purpose
---|---
| **Step 7** record-route | Enables record-route functionality for a GMPLS tunnel. |
| **Example:** RP/0/RP0/CPU0:router(config-te-gmpls-tun)# record-route | |
| **Step 8** signalled-name name | Configures signalled-name for a GMPLS tunnel. |
| **Example:** RP/0/RP0/CPU0:router(config-te-gmpls-tun)# signalled-name sign1 | |
| **Step 9** logging events lsp-status state | Configure events to generate system log messages when state changes occur on the GMPLS tunnel. If omitted, no events will result in the generation of system log messages. |
| **Example:** RP/0/RP0/CPU0:router(config-te-gmpls-tun)# logging events lsp-status state | |
| **Step 10** commit | |

#### Configuring LSP Diversity

To configure an XRO attribute-set as part of the path-option for MPLS-TE, and to specify exclusions for an attribute set for LSP diversity, follow these procedures:

#### Configuring XRO Attribute-set

Perform this task to configure XRO attribute set in the GMPLS UNI tunnel path-option, under MPLS-TE submode.

### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. gmpls optical-uni
4. controller dwdm interface
5. tunnel-properties
6. path-option 10 no-ero [xro-attribute-set name] lockdown
7. commit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> gmpls optical-uni</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-mpls-te)# gmpls optical-uni</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> controller dwdm interface</td>
<td>Enters GMPLS UNI controller submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-cntl)# controller dwdm 0/1/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> tunnel-properties</td>
<td>Enters the submode to configure tunnel-specific information for a GMPLS UNI controller.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-cntl)# tunnel-properties</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> path-option 10 no-ero [xro-attribute-set name ] lockdown</td>
<td>Specifies the path-option for a headend router of a GMPLS tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-te-gmpls-tun)# path-option 10 no-ero xro-attribute-set A01 lockdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Connection Diversity

Perform this task to specify exclusions for an attribute set for LSP diversity, under MPLS-TE attribute-set configuration mode.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. attribute-set xro name
4. exclude {best-effort | strict} lsp source source-address destination destination-address tunnel-id tunnel-id extended-tunnel-id extended-tunnel-id [lsp-id lsp-id]
5. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Configures an XRO attribute-set for a GMPLS tunnel.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3 attribute-set xro name</td>
<td>Specifies exclusions for an attribute set for LSP diversity.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-mpls-te)# attribute-set xro attrset01</td>
<td></td>
</tr>
<tr>
<td>Step 4 exclude {best-effort</td>
<td>strict} lsp source source-address destination destination-address tunnel-id tunnel-id extended-tunnel-id extended-tunnel-id [lsp-id lsp-id]</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-te-attribute-set)# exclude best-effort lsp source 10.10.1.2 destination 10.20.4.4 tunnel-id 17 extended-tunnel-id 10.20.3.3 lsp-id 17 RP/0/RP0/CPU0:router(config-te-attribute-set)#</td>
<td></td>
</tr>
<tr>
<td>Step 5 commit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring LMP for GMPLS UNI

To implement LMP configuration for GMPLS UNI, follow these procedures:

Configuring Optical Router ID

Perform this task to enable GMPLS UNI LMP functionality and to configure LMP unicast router ID.
SUMMARY STEPS

1. configure
2. lmp
3. gmpls optical-uni
4. router-id ipv4 unicast address
5. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>lmp</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters LMP configuration mode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config)# lmp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>gmpls optical-uni</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-lmp)# gmpls optical-uni</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>router-id ipv4 unicast address</td>
</tr>
<tr>
<td>Example:</td>
<td>Configures the LMP unicast router ID for GMPLS.</td>
</tr>
<tr>
<td>RP/0/RP0/CPU0:router(config-lmp-gmpls-uni)# router-id ipv4 unicast 10.10.4.4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

Configuring an LMP Neighbor

Perform this task to configure an LMP neighbor for a GMPLS UNI tunnel.
SUMMARY STEPS

1. configure
2. lmp
3. gmpls optical-uni
4. neighbor name
5. ipcc routed
6. router-id ipv4 unicast address
7. commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters LMP configuration mode.</td>
</tr>
<tr>
<td>Step 2 lmp</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
</tbody>
</table>

Example:
```
RP/0/RP0/CPU0:router(config)# lmp
```

| Step 3 gmpls optical-uni | Specifies an LMP neighbor for GMPLS and enters LMP GMPLS UNI neighbor configuration submode. |

Example:
```
RP/0/RP0/CPU0:router(config-lmp)# gmpls optical-uni
```

| Step 4 neighbor name | Specifies the LMP neighbor IPCC configuration for GMPLS UNI. |

Example:
```
RP/0/RP0/CPU0:router(config-lmp-gmpls-uni)# neighbor nbr1
```

| Step 5 ipcc routed | Configures the LMP unicast router ID for GMPLS. |

Example:
```
RP/0/RP0/CPU0:router(config-lmp-gmpls-uni-nbr-nbr1)# ipcc routed
```

| Step 6 router-id ipv4 unicast address | |

Example:
```
RP/0/RP0/CPU0:router(config-lmp-gmpls-uni-nbr-nbr1)# router-id ipv4 unicast 10.10.4.4
```

| Step 7 commit | |

Cisco IOS XR MPLS Configuration Guide for the Cisco CRS Router, Release 4.3.x
Configuring an LMP Controller

Perform this task to configure an LMP link for a GMPLS UNI controller.

**SUMMARY STEPS**

1. configure
2. lmp
3. gmpls optical-uni
4. controller dwdm controller
5. neighbor name
6. link-id ipv4 unicast address
7. neighbor link-id ipv4 unicast address
8. neighbor interface-id unnumbered interface-id
9. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters LMP configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> lmp</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config)# lmp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> gmpls optical-uni</td>
<td>Specifies a controller for GMPLS UNI.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-lmp)# gmpls optical-uni</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> controller dwdm controller</td>
<td>Specifies an LMP neighbor for GMPLS and enters LMP GMPLS UNI neighbor configuration submode.</td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-lmp-gmpls-uni)# controller dwdm 0/4/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor name</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RP0/CPU0:router(config-lmp-gmpls-uni-cntl)# neighbor nbr1</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring RSVP Optical Refresh Interval and Missed Count

Perform this task to configure optical refresh interval under the RSVP controller submode and to configure the number of missed refresh messages allowed before optical tunnel states are deleted.

**SUMMARY STEPS**

1. `configure`
2. `rsvp`
3. `controller dwdm interface`
4. `signalling refresh out-of-band interval interval`
5. `signalling refresh out-of-band missed miss-count`
6. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
</tbody>
</table>

---

### Purpose

#### Step 6

`link-id ipv4 unicast address`

**Example:**

```
RP/0/RP0/CPU0:router(config-lmp-gmpls-uni-cntl)#
link-id ipv4 unicast 10.2.2.4
```

**Purpose:** Specifies the optical interface address for an LMP link for a GMPLS UNI controller.

#### Step 7

`neighbor link-id ipv4 unicast address`

**Example:**

```
RP/0/RP0/CPU0:router(config-lmp-gmpls-uni-cntl)#
neighbor link-id ipv4 unicast 10.10.4.4
```

**Purpose:** Specifies the neighbor's optical address of an LMP link for a GMPLS UNI controller.

#### Step 8

`neighbor interface-id unnumbered interface-id`

**Example:**

```
RP/0/RP0/CPU0:router(config-lmp-gmpls-uni-cntl)#
neighbor interface-id unnumbered 17
```

**Purpose:** Specifies the neighbor's optical interface ID of an LMP link for a GMPLS UNI controller.

#### Step 9

`commit`

---
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td><code>rsvp</code></td>
<td>Enters RSVP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config)# rsvp</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>controller dwdm interface</code></td>
<td>Configures a controller for establishing a GMPLS UNI tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config-rsvp)# controller dwdm 0/1/0/1</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>signalling refresh out-of-band interval interval</code></td>
<td>Configures optical refresh interval.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config-rsvp-cntl)# signalling refresh out-of-band interval 200</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>signalling refresh out-of-band missed miss-count</code></td>
<td>Configures number of missed refresh messages allowed before optical tunnel states are deleted.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RP0/CPU0:router(config-rsvp-cntl)# signalling refresh out-of-band missed 30</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>commit</code></td>
<td></td>
</tr>
</tbody>
</table>

### Configuration Examples for GMPLS UNI

These configuration examples are provided for GMPLS UNI:

### Configuring Head UNI-C for a GMPLS Tunnel: Example

This example shows the minimal head UNI-C configuration require to establish a GMPLS tunnel:

```
rsvp
  controller dwdm 0/1/0/1
  signalling refresh out-of-band interval 3600
  signalling refresh out-of-band missed 24
!
! mpls traffic-eng
mpls optical-uni
  controller dwdm 0/1/0/1
  tunnel-properties
```
Configuring Tail UNI-C for a GMPLS Tunnel: Example

This example shows the minimal tail UNI-C configuration require to establish a GMPLS tunnel:

```plaintext
Note
The controller must be specified under the GMPLS UNI submode to inform TE that incoming GMPLS path messages are to be accepted and processed.
```

```plaintext
tunnel-id 100
  destination 100.20.20.20
  path-option 10 no-ero

lmp
gmpls optical-uni
  router-id 100.11.11.11
  neighbor nbr_A
    ipcc routed
    neighbor router-id ipv4 unicast 100.12.12.12
  controller dwdm 0/1/0/1
    neighbor nbr_A
      link-id ipv4 unicast 192.168.100.1
      neighbor link-id ipv4 unicast 192.168.100.2
      neighbor interface-id unnumbered 13

mpls traffic-eng
  gmpls optical-uni
    controller dwdm 0/1/0/1

lmp
gmpls optical-uni
  router-id 100.20.20.20
  neighbor nbr_B
    ipcc routed
    neighbor router-id ipv4 unicast 100.19.19.19
  controller dwdm 0/1/0/1
    neighbor nbr_B
      link-id ipv4 unicast 192.168.103.2
      neighbor link-id ipv4 unicast 192.168.103.1
      neighbor interface-id unnumbered 22
```
Configuring LSP Diversity: Example

This example shows the configuration for two diverse LSPs:

```
config]
  mpls traffic-eng
  attribute-set xro exclude-tun1
  exclude best-effort lsp source 88.0.0.8 destination 10.0.0.2 tunnel-id 1
  extended-tunnel-id 88.0.0.8
  !
  attribute-set xro exclude-tun2
  exclude strict lsp source 88.0.0.8 destination 10.0.1.2 tunnel-id 2 extended-tunnel-id 88.0.0.8 lsp-id 2
  
  gmpls optical-uni
  controller dwdm 0/1/0/0
  tunnel-properties
  logging events lsp-status state
  tunnel-id 1
  destination ipv4 unicast 10.0.0.2
  path-option 10 no-ero xro-attribute-set exclude-tun2
  !
  
  controller dwdm 0/1/0/1
  tunnel-properties
  logging events lsp-status state
  tunnel-id 2
  destination ipv4 unicast 10.0.1.2
  path-option 10 no-ero xro-attribute-set exclude-tun1
  !
  
```

Additional References

For additional information related to implementing GMPLS UNI, refer to the following references:

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMPLS UNI commands</td>
<td>GMPLS UNI Commands module in Cisco IOS XR MPLS Command Reference for the Cisco CRS Router</td>
</tr>
<tr>
<td>MPLS Traffic Engineering commands</td>
<td>MPLS Traffic Engineering commands module in Cisco IOS XR MPLS Command Reference for the Cisco CRS Router</td>
</tr>
<tr>
<td>RSVP commands</td>
<td>RSVP commands module in Cisco IOS XR MPLS Command Reference for the Cisco CRS Router</td>
</tr>
<tr>
<td>Getting started material</td>
<td>Cisco IOS XR Getting Started Guide for the Cisco CRS Router</td>
</tr>
</tbody>
</table>
### Related Topic

| Information about user groups and task IDs | Configuring AAA Services module in Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router |

### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>OIF UNI 1.0</td>
<td>User Network Interface (UNI) 1.0 Signaling Specification</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 3471</td>
<td>Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description</td>
</tr>
<tr>
<td>RFC 4872</td>
<td>RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery</td>
</tr>
</tbody>
</table>
### RFCs

| RFC 6205 | Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers |

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
Implementing MPLS OAM

This module describes Multiprotocol Label Switching (MPLS) P2MP Ping and Traceroute features. These feature provide a means to check connectivity, isolate failure point, thus providing the MPLS Operations, Administration, and Maintenance (OAM) solution.

For detailed information about MPLS commands and examples, see Cisco IOS XR MPLS Command Reference for the Cisco CRS Router.

Feature History for Implementing MPLS OAM

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.0.0</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for MPLS LSP Ping and Traceroute for P2MP, page 305
- MPLS Network Management with MPLS LSP Ping and MPLS LSP Traceroute, page 306
- Roles of Various Routers, page 306
- P2MP Ping, page 307
- P2MP Traceroute, page 308
- MPLS OAM Support for BGP 3107, page 309
- Configure the Ping and Traceroute: Example, page 309

Prerequisites for MPLS LSP Ping and Traceroute for P2MP

Before you use the MPLS LSP Ping and Traceroute for P2MP feature, you should have the support for following:

- Cisco IOS XR software Release 4.0.0 or a later release
- Configure Resource Reservation Protocol (RSVP) features on the headend, midpoint, and tailend routers in the MPLS network
Configure traffic engineering features on the headend, midpoint, and tailend routers in the MPLS network

Enable MPLS OAM using the `mpls oam` command on all routers in the MPLS network

**MPLS Network Management with MPLS LSP Ping and MPLS LSP Traceroute**

To manage an MPLS network, you must have the ability to monitor LSPs and quickly isolate MPLS forwarding problems. You need ways to characterize the liveliness of an LSP and reliably detect when an LSP fails to deliver user traffic.

You can use MPLS LSP ping to verify the LSP that is used to transport packets. You can use MPLS LSP traceroute to trace LSPs that are used to carry packets destined for P2MP LSP.

An MPLS echo request is sent through an LSP to validate it. A TTL expiration or LSP breakage causes the transit router to process the echo request before it gets to the intended destination. The router returns an MPLS echo reply that contains an explanatory reply code to the originator of the echo request.

The successful echo request is processed at the egress of the LSP. The echo reply is sent through an IP path, an MPLS path, or a combination of both, back to the originator of the echo request.

**Roles of Various Routers**

A P2MP TE network contains the following elements:

- **Headend Router**
  The headend router, also called the source or ingress router, is responsible for initiating the signaling messages that set up the P2MP TE LSP. The headend router can also be a branch point, which means the router performs packet replication and the sub-LSPs split into different directions.

- **Midpoint Router**
  The midpoint router is where the sub-LSP signaling is processed. The midpoint router can be a branch point.

- **Tailend Router**
  The tailend router, also called the destination, egress, or leaf-node router, is where sub-LSP signaling ends. The router which is one of potentially many destinations of the P2MP TE LSP.

- **Bud Router**
  A bud router is a midpoint and tailend router at the same time. An LSR that is an egress LSR, but also has one or more directly connected downstream LSRs.

- **Branch Router**
  A branch router is either a midpoint or tailend router at any given time.

- **Transit Router**
  A transit router is an LSR that is not an egress router, but also has one or more directly connected downstream routers.

- A P2MP tunnel consists of one or more sub-LSPs. All sub-LSPs belonging to the same P2MP tunnel employ the same constraints, protection policies, and so on, which are configured at the headend router.
Figure 19: Elements of P2MP TE Network illustrates the elements of P2MP TE network.

Figure 19: Elements of P2MP TE Network

P2MP TE tunnels build on the features that exist in basic point-to-point TE tunnels. The P2MP TE tunnels have the following characteristics:

- There is one source (headend) but more than one destination (tailend).
- They are unidirectional.
- They are explicitly routed.
- Multiple sub-LSPs connect the headend router to various tailend routers.

**P2MP Ping**

The P2MP ping feature is used to check the connectivity between Ingress LSR and egress LSR, along a P2MP LSP. The Ingress LSR sends the P2MP echo request message along the specified P2MP LSP. All egress LSRs which receive the P2MP echo request message from the ingress LSR must send a P2MP echo reply message to the ingress LSR, according to the reply mode specified in the P2MP echo request message.

MPLS LSP ping uses MPLS echo request and reply packets to validate an LSP. You can use MPLS LSP ping to validate RSVP P2MP IPv4 FECs by using appropriate keywords and arguments with the `ping mpls` command.
The MPLS echo request packet is sent to a target router through the use of the appropriate label stack associated with the LSP to be validated. Use of the label stack causes the packet to be forwarded over the LSP itself.

The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.x.y.z/8 address. The 127.x.y.z/8 address prevents the IP packet from being IP switched to its destination, if the LSP is broken.

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

The MPLS echo reply destination port is set to the echo request source port.

---

**Note**

Only P2MP TE LSP IPv4 is supported. If the Responder Identifier TLV is missing, the *echo request* requests information from all responder-ids.

---

**Jitter**

Jitter is used to reduce the load on the LSR where the ping is performed. By adding a jitter, the replying routers will space their reply time based on a random number between 0 and the jitter value, Jitter TLV, specified in the packet.

---

**P2MP Traceroute**

The P2MP traceroute feature is used to isolate the failure point of a P2MP LSP. It is used for hop-by-hop fault localization and path tracing. The traceroute feature relies on the expiration of the TTL of the packet that carries the echo request. When the P2MP echo request message hits a transit node, it checks the TTL and if it is expired, the packet is punted to the control plane, else the message is forwarded or replicated. If punted to the control plane, a reply message is build based on the contents of the request message.

Traceroute can be applied to all nodes in the P2MP tree. However, you can select a specific traceroute target through the P2MP Responder Identifier TLV. An entry in this TLV represents an responder-id or a transit node. This is only the case for P2MP TE LSPs.

---

**Note**

Only P2MP TE LSP IPv4 is supported. If the Responder Identifier TLV is missing, the *echo request* requests information from all responder-ids.

---

**Jitter**

Jitter is used to reduce the load on the LSR where the traceroute is performed. By adding a jitter, the replying routers will space their reply time based on a random number between 0 and the jitter value, , Jitter TLV, specified in the packet.

For more information about ping and traceroute commands, see *MPLS OAM commands* chapter in the *Cisco IOS XR MPLS Command Reference for the Cisco CRS Router*.
MPLS OAM Support for BGP 3107

The MPLS OAM Support for BGP 3107 feature provides support for ping, traceroute and treetrace (traceroute multipath) operations for LSPs signaled via BGP for the IPv4 unicast prefix FECs in the default VRF, according to the RFC 3107 - Carrying Label Information in BGP-4. This feature adds support for MPLS OAM operations in the seamless MPLS architecture deployments, i.e., combinations of BGP and LDP signaled LSPs.

Configure the Ping and Traceroute: Example

This section contains examples of the ping and traceroute commands, based on this topology.

![Topology Diagram]

This example shows multiple destinations set on the assigned LSP path.

```
RP/0/RP0/CPU0:router# show run int tunnel-mte 10
interface tunnel-mte10
 ipv4 unnumbered Loopback0
destination 11.0.0.1
  path-option 1 dynamic
! destination 12.0.0.1
  path-option 1 dynamic
! destination 13.0.0.1
  path-option 1 dynamic
!
```

This example shows an extract of the ping command.

```
# ping mpls traffic-eng tunnel-mte 10
Sending 1, 100-byte MPLS Echos to tunnel-mte10,
  timeout is 2.2 seconds, send interval is 0 msec, jitter value is 200 msec:

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

Type escape sequence to abort.
Request #1
! reply addr 192.168.222.2
! reply addr 192.168.140.2
! reply addr 192.168.170.1
```

Cisco IOS XR MPLS Configuration Guide for the Cisco CRS Router, Release 4.3.x
Success rate is 100 percent (3 received replies/3 expected replies),
round-trip min/avg/max = 154/232/302 ms

This example shows an extract of the ping command with the jitter option.

RP/0/RP0/CPU0:router# ping mpls traffic-eng tunnel-mte 10 jitter 300

Sending 1, 100-byte MPLS Echos to tunnel-mte10,
timeout is 2.3 seconds, send interval is 0 msec, jitter value is 300 msec:

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

Type escape sequence to abort.

Request #1
! reply addr 192.168.222.2
! reply addr 192.168.140.2
! reply addr 192.168.170.1

Success rate is 100 percent (3 received replies/3 expected replies),
round-trip min/avg/max = 148/191/256 ms

This example shows an extract of the ping command with the ddmap option.

RP/0/RP0/CPU0:router# ping mpls traffic-eng tunnel-mte 10 ddmap

Sending 1, 100-byte MPLS Echos to tunnel-mte10,
timeout is 2.2 seconds, send interval is 0 msec, jitter value is 200 msec:

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

Type escape sequence to abort.

Request #1
! reply addr 192.168.222.2
! reply addr 192.168.140.2
! reply addr 192.168.170.1

Success rate is 100 percent (3 received replies/3 expected replies),
round-trip min/avg/max = 105/178/237 ms

RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels p2mp 10
Mon Apr 12 12:13:55.075 EST
Signalling Summary:
  LSP Tunnels Process: running
  RSVP Process: running
  Forwarding: enabled
  Periodic reoptimization: every 3600 seconds, next in 654 seconds
  Periodic FRR Promotion: every 300 seconds, next in 70 seconds
  Auto-bw enabled tunnels: 0 (disabled)

Name: tunnel-mte10
Status:
  Admin: up  Oper: up (Up for 12w4d)

Config Parameters:
Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
Metric Type: TE (default)
Fast Reroute: Not Enabled, Protection Desired: None
Record Route: Not Enabled

Destination summary: (3 up, 0 down, 0 disabled) Affinity: 0x0/0xffff
Auto-bw: disabled
Destination: 11.0.0.1
State: Up for 12w4d
Path options:
  path-option 1 dynamic [active]
Destination: 12.0.0.1
State: Up for 12w4d
Path options:
  path-option 1 dynamic [active]
Destination: 13.0.0.1
State: Up for 12w4d
Path options:
  path-option 1 dynamic [active]

History:
  Reopt. LSP:
    Last Failure:
      LSP not signalled, identical to the [CURRENT] LSP
    Date/Time: Thu Jan 14 02:49:22 EST 2010 [12w4d ago]

Current LSP:
  lsp-id: 10002 p2mp-id: 10 tun-id: 10 src: 10.0.0.1 extid: 10.0.0.1
  LSP up for: 12w4d
  Reroute Pending: No
  Inuse Bandwidth: 0 kbps (CT0)
  Number of S2Ls: 3 connected, 0 signaling proceeding, 0 down

S2L Sub LSP: Destination 11.0.0.1 Signaling Status: connected
  S2L up for: 12w4d
  Sub Group ID: 1 Sub Group Originator ID: 10.0.0.1
  Path option path-option 1 dynamic (path weight 1)
  Path info (OSPF 1 area 0)
  192.168.222.2
  11.0.0.1

S2L Sub LSP: Destination 12.0.0.1 Signaling Status: connected
  S2L up for: 12w4d
  Sub Group ID: 2 Sub Group Originator ID: 10.0.0.1
  Path option path-option 1 dynamic (path weight 2)
  Path info (OSPF 1 area 0)
  192.168.222.2
  192.168.140.3
  192.168.140.2
  12.0.0.1

S2L Sub LSP: Destination 13.0.0.1 Signaling Status: connected
  S2L up for: 12w4d
  Sub Group ID: 3 Sub Group Originator ID: 10.0.0.1
  Path option path-option 1 dynamic (path weight 2)
  Path info (OSPF 1 area 0)
  192.168.222.2
  192.168.170.3
  192.168.170.1
  13.0.0.1

Reoptimized LSP (Install Timer Remaining 0 Seconds):
  None
Cleaned LSP (Cleanup Timer Remaining 0 Seconds):
  None
Displayed 1 (of 16) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

RP/0/RP0/CPU0:router# ping mpls traffic-eng tunnel-mte 10 lsp id 10002
Mon Apr 12 12:14:04.532 EST
Sending 1, 100-byte MPLS Echos to tunnel-mte10,
timeout is 2.2 seconds, send interval is 0 msec, jitter value is 200 msec:
This example shows an extract of the ping command with the responder-id of R3.

```
RP/0/RP0/CPU0:router# ping mpls traffic-eng tunnel-mte 10 responder-id 13.0.0.1
Mon Apr 12 12:15:34.205 EST
Sending 1, 100-byte MPLS Echos to tunnel-mte10,
  timeout is 2.2 seconds, send interval is 0 msec, jitter value is 200 msec:
Codes: '!1' - success, 'Q' - request not sent, '.1' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0, 'd' - DDMAP
Type escape sequence to abort.
Request #1
! reply addr 192.168.222.2
! reply addr 192.168.170.1
! reply addr 192.168.140.2
Success rate is 100 percent (3 received replies/3 expected replies),
  round-trip min/avg/max = 128/153/167 ms
```

This example shows an extract of the traceroute command with the ttl option.

```
RP/0/RP0/CPU0:router# traceroute mpls traffic-eng tunnel-mte 10 ttl 4
Mon Apr 12 12:16:50.095 EST
Tracing MPLS MTE Label Switched Path on tunnel-mte10, timeout is 2.2 seconds
Codes: '!1' - success, 'Q' - request not sent, '.1' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0, 'd' - DDMAP
Type escape sequence to abort.
! 1 192.168.222.2 186 ms [Estimated Role: Bud]
  [L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]
! 2 192.168.222.2 115 ms [Estimated Role: Egress]
! 2 192.168.140.2 213 ms [Estimated Role: Egress]
! 3 192.168.170.1 254 ms [Estimated Role: Egress]
```

Implementing MPLS OAM Configure the Ping and Traceroute: Example

Cisco IOS XR MPLS Configuration Guide for the Cisco CRS Router, Release 4.3.x
This example shows an extract of the traceroute command with the responder-id option.

RP/0/RP0/CPU0:router# traceroute mpls traffic-eng tunnel-mte 10 responder-id 13.0.0.1

Tracing MPLS MTE Label Switched Path on tunnel-mte10, timeout is 2.2 seconds

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

Type escape sequence to abort.

d 1 192.168.222.2 113 ms [Estimated Role: Branch]
  [L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]

This example shows an extract of the traceroute command with the jitter option.

RP/0/RP0/CPU0:router# traceroute mpls traffic-eng tunnel-mte 10 responder-id 13.0.0.1 ttl 5

Tracing MPLS MTE Label Switched Path on tunnel-mte10, timeout is 2.5 seconds

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

Type escape sequence to abort.

d 1 192.168.222.2 238 ms [Estimated Role: Branch]
  [L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]
Configure the Ping and Traceroute: Example

! 2 192.168.170.1 290 ms [Estimated Role: Egress]

d 3 192.168.222.2 115 ms [Estimated Role: Branch]
[L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]
! 3 192.168.170.1 428 ms [Estimated Role: Egress]

d 4 192.168.222.2 127 ms [Estimated Role: Branch]
[L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]
! 4 192.168.170.1 327 ms [Estimated Role: Egress]