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

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 5
Prerequisites for Implementing Cisco MPLS LDP 6
Information About Implementing Cisco MPLS LDP 6
Overview of Label Distribution Protocol 6
Label Switched Paths 7
LDP Control Plane 7
Exchanging Label Bindings 7
LDP Forwarding 8
LDP Graceful Restart 9
Control Plane Failure 10
Phases in Graceful Restart 11
Recovery with Graceful-Restart 12
Label Advertisement Control (Outbound Filtering) 13
Label Acceptance Control (Inbound Filtering) 13
Local Label Allocation Control 14
Session Protection 14
IGP Synchronization 15
IGP Auto-configuration 16
LDP Nonstop Routing 16
Contents

IP LDP Fast Reroute Loop Free Alternate 17
Downstream on Demand 18
Explicit-Null and Implicit-Null Labels 19
How to Implement MPLS LDP 19
  Configuring LDP Discovery Parameters 20
  Configuring LDP Discovery Over a Link 21
  Configuring LDP Discovery for Active Targeted Hellos 23
  Configuring LDP Discovery for Passive Targeted Hellos 25
  Configuring Label Advertisement Control (Outbound Filtering) 28
Setting Up LDP Neighbors 29
Setting Up LDP Forwarding 31
Setting Up LDP NSF Using Graceful Restart 32
Configuring Label Acceptance Control (Inbound Filtering) 35
Configuring Local Label Allocation Control 36
Configuring Session Protection 37
Configuring LDP IGP Synchronization: OSPF 38
Configuring LDP IGP Synchronization: ISIS 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
Setting Up Implicit-Null-Override Label 46
Redistributing MPLS LDP Routes into BGP 47
Configuration Examples for Implementing MPLS LDP 48
  Configuring LDP with Graceful Restart: Example 48
  Configuring LDP Discovery: Example 49
  Configuring LDP Link: Example 49
  Configuring LDP Discovery for Targeted Hellos: Example 49
  Configuring Label Advertisement (Outbound Filtering): Example 50
  Configuring LDP Neighbors: Example 50
  Configuring LDP Forwarding: Example 51
  Configuring LDP Nonstop Forwarding with Graceful Restart: Example 51
  Configuring Label Acceptance (Inbound Filtering): Example 51
  Configuring Local Label Allocation Control: Example 52
CHAPTER 3 Implementing RSVP for MPLS-TE 59

Prerequisites for Implementing RSVP for MPLS-TE 60
Information About Implementing RSVP for MPLS-TE 60
  Overview of RSVP for MPLS-TE 60
  LSP Setup 61
  High Availability 61
  Graceful Restart 61
    Graceful Restart: Standard and Interface-Based 62
    Graceful Restart: Figure 63
  ACL-based Prefix Filtering 64
  RSVP MIB 64
  Bandwidth Reservation Percentage 65
Information About Implementing RSVP Authentication 65
  RSVP Authentication Functions 65
  RSVP Authentication Design 65
  Global, Interface, and Neighbor Authentication Modes 66
  Security Association 67
  Key-source Key-chain 68
  Guidelines for Window-Size and Out-of-Sequence Messages 69
  Caveats for Out-of-Sequence 69
How to Implement RSVP 70
  Configuring Traffic Engineering Tunnel Bandwidth 70
  Confirming DiffServ-TE Bandwidth 70
  Enabling Graceful Restart 71
  Configuring ACL-based Prefix Filtering 72
    Configuring ACLs for Prefix Filtering 72
    Configuring RSVP Packet Dropping 73
Verifying RSVP Configuration  74

Enabling RSVP Traps  77

How to Implement RSVP Authentication  78

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  80

Configuring an Interface for RSVP Authentication  81

  Specifying the RSVP Authentication Keychain in Interface Mode  81
  Configuring a Lifetime for an Interface for RSVP Authentication  82
  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  88

Configuration Examples for RSVP  88

  Bandwidth Configuration (Prestandard): Example  89
  Bandwidth Configuration (MAM): Example  89
  Bandwidth Configuration (RDM): Example  89
  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  90
    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  91
    Change the Hello Interval: Example  92
  Configure ACL-based Prefix Filtering: Example  92
  Set DSCP for RSVP Packets: Example  92
<table>
<thead>
<tr>
<th>Backup AutoTunnel Assignment</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit Paths</td>
<td>112</td>
</tr>
<tr>
<td>Periodic Backup Promotion</td>
<td>112</td>
</tr>
<tr>
<td>Protocol-Based CLI</td>
<td>113</td>
</tr>
<tr>
<td>Differentiated Services Traffic Engineering</td>
<td>113</td>
</tr>
<tr>
<td>Prestandard DS-TE Mode</td>
<td>113</td>
</tr>
<tr>
<td>IETF DS-TE Mode</td>
<td>113</td>
</tr>
<tr>
<td>Bandwidth Constraint Models</td>
<td>114</td>
</tr>
<tr>
<td>Maximum Allocation Bandwidth Constraint Model</td>
<td>114</td>
</tr>
<tr>
<td>Russian Doll Bandwidth Constraint Model</td>
<td>114</td>
</tr>
<tr>
<td>TE Class Mapping</td>
<td>115</td>
</tr>
<tr>
<td>Flooding</td>
<td>115</td>
</tr>
<tr>
<td>Flooding Triggers</td>
<td>115</td>
</tr>
<tr>
<td>Flooding Thresholds</td>
<td>116</td>
</tr>
<tr>
<td>Fast Reroute</td>
<td>116</td>
</tr>
<tr>
<td>MPLS-TE and Fast Reroute over Link Bundles</td>
<td>117</td>
</tr>
<tr>
<td>Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE</td>
<td>117</td>
</tr>
<tr>
<td>Flexible Name-based Tunnel Constraints</td>
<td>118</td>
</tr>
<tr>
<td>MPLS Traffic Engineering Interarea Tunneling</td>
<td>119</td>
</tr>
<tr>
<td>Interarea Support</td>
<td>119</td>
</tr>
<tr>
<td>Multiarea Support</td>
<td>120</td>
</tr>
<tr>
<td>Loose Hop Expansion</td>
<td>120</td>
</tr>
<tr>
<td>Loose Hop Reoptimization</td>
<td>121</td>
</tr>
<tr>
<td>ABR Node Protection</td>
<td>121</td>
</tr>
<tr>
<td>Fast Reroute Node Protection</td>
<td>121</td>
</tr>
<tr>
<td>MPLS-TE Forwarding Adjacency</td>
<td>121</td>
</tr>
<tr>
<td>MPLS-TE Forwarding Adjacency Benefits</td>
<td>122</td>
</tr>
<tr>
<td>MPLS-TE Forwarding Adjacency Restrictions</td>
<td>122</td>
</tr>
<tr>
<td>MPLS-TE Forwarding Adjacency Prerequisites</td>
<td>122</td>
</tr>
<tr>
<td>Path Computation Element</td>
<td>122</td>
</tr>
<tr>
<td>Policy-Based Tunnel Selection</td>
<td>124</td>
</tr>
<tr>
<td>Policy-Based Tunnel Selection</td>
<td>124</td>
</tr>
<tr>
<td>Policy-Based Tunnel Selection Functions</td>
<td>124</td>
</tr>
<tr>
<td>PBTS Restrictions</td>
<td>124</td>
</tr>
<tr>
<td>Path Protection</td>
<td>125</td>
</tr>
</tbody>
</table>
Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link  189
Creating an Explicit Path With Exclude SRLG  191
Using Explicit Path With Exclude SRLG  192
Creating a Link Protection on Backup Tunnel with SRLG Constraint  194
Creating a Node Protection on Backup Tunnel with SRLG Constraint  197
Configuring Default Admin Weight  200
Configuring Static SRLG Value to Topology Link  200
Configuring Admin-Weight Associated with an SRLG Value  201
Configuring Point-to-Multipoint TE  202
   Enabling Multicast Routing on the Router  202
   Configuring the Static Group for the Point-to-Multipoint Interface  204
   Configuring Destinations for the Tunnel Interface  206
   Disabling Destinations  209
   Logging Per Destinations for Point-to-Multipoint  210
Enabling Soft-Preemption on a Node  212
Enabling Soft-Preemption on a Tunnel  213
Configuring Attributes within a Path-Option Attribute  214
Configuring Auto-Tunnel Mesh Tunnel ID  216
Configuring Auto-tunnel Mesh Unused Timeout  217
Configuring Auto-Tunnel Mesh Group  218
Configuring Tunnel Attribute-Set Templates  219
Enabling LDP on Auto-Tunnel Mesh  221
Implementing Associated Bidirectional Label Switched Paths  222
   Signaling Methods and Object Association for Bidirectional LSPs  223
   Associated Bidirectional Non Co-routed and Co-routed LSPs  224
   Configure Associated Bidirectional Co-routed LSPs  225
   Path Protection  227
       Configure Path Protection for Associated Bidirectional LSPs  228
   OAM Support for Associated Bidirectional LSPs  229
       Generate Fault OAM Messages at Mid-point  230
       Generate Fault OAM Messages at End-point  231
Pseudowire Call Admission Control  231
Configuration Examples for Cisco MPLS-TE  231
   Build MPLS-TE Topology and Tunnels: Example  231
   Configure IETF DS-TE Tunnels: Example  233
Configure MPLS-TE and Fast-Reroute on OSPF: Example 233
Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example 234
Configure Flexible Name-based Tunnel Constraints: Example 234
Configure an Interarea Tunnel: Example 236
Configure Forwarding Adjacency: Example 237
Configure PCE: Example 237
Configure Tunnels for Path Protection: Example 238
Configure Tunnels for Explicit Path Protection: Example 238
Configure Tunnels for Co-existence of Path Protection with Fast Reroute: Example 239
Configure Automatic Bandwidth: Example 239
Configure the MPLS-TE Shared Risk Link Groups: Example 239
Configure the MPLS-TE Auto-Tunnel Backup: Example 242
Configure Point-to-Multipoint TE: Examples 248
  P2MP Topology Scenario: Example 249
  Configure Point-to-Multipoint for the Source: Example 250
  Configure the Point-to-Multipoint Tunnel: Example 251
  Disable a Destination: Example 251
  Configure the Point-to-Multipoint Solution: Example 252
Additional References 255

CHAPTER 6

Implementing GMPLS UNI 257
  Prerequisites for Implementing GMPLS UNI 257
  Restrictions for Implementing GMPLS UNI 258
  Information About Implementing GMPLS UNI 258
    GMPLS UNI vs GMPLS NNI 258
    GMPLS LSP Signaling 258
      Path Message without an ERO 259
      XRO Attribute-set 259
      Connection Diversity 259
    DWDM Transponder Integration 259
  How to Implement GMPLS UNI 260
    Configuring TE for GMPLS UNI 260
      Enabling GMPLS UNI Submode 260
      Configuring GMPLS UNI Controller 261
      Configuring GMPLS UNI Controller as a Tunnel Head 262
Configuring Other Tunnel Properties for a GMPLS UNI Tunnel 263
Configuring LSP Diversity 265
  Configuring XRO Attribute-set 265
  Configuring Connection Diversity 266
Configuring LMP for GMPLS UNI 267
  Configuring Optical Router ID 267
  Configuring an LMP Neighbor 268
  Configuring an LMP Controller 270
Configuring RSVP Optical Refresh Interval and Missed Count 271
Configuration Examples for GMPLS UNI 272
  Configuring Head UNI-C for a GMPLS Tunnel: Example 272
  Configuring Tail UNI-C for a GMPLS Tunnel: Example 273
  Configuring LSP Diversity: Example 274
Additional References 274

CHAPTER 7  Implementing MPLS OAM 277
  Prerequisites for MPLS LSP Ping and Traceroute for P2MP 277
  MPLS Network Management with MPLS LSP Ping and MPLS LSP Traceroute 278
  Roles of Various Routers 278
  P2MP Ping 279
  P2MP Traceroute 280
  MPLS OAM Support for BGP 3107 281
  IP-Less MPLS-TP Ping and MPLS-TP Traceroute 281
  Configure the Ping and Traceroute: Example 281

CHAPTER 8  Implementing MPLS Transport Profile 287
  Restrictions for MPLS-TP 287
  Information About Implementing MPLS Transport Profile 288
    MPLS Transport Profile 288
    Bidirectional LSPs 289
    MPLS-TP Path Protection 289
    Fault OAM Support 289
    MPLS-TP Links and Physical Interfaces 291
    Tunnel LSPs 291
    MPLS-TP IP-less support 292
Preface

From Release 6.1.1 onwards, Cisco introduces support for the 64-bit Linux-based IOS XR operating system. Extensive feature parity is maintained between the 32-bit and 64-bit environments. Unless explicitly marked otherwise, the contents of this document are applicable for both the environments. For more details on Cisco IOS XR 64 bit, refer to the Release Notes for Cisco ASR 9000 Series Routers, Release 6.1.1 document.

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-28381-02</td>
<td>May 2013</td>
<td>Republished with documentation updates for Cisco IOS XR Release 4.3.1 features.</td>
</tr>
<tr>
<td>OL-28381-01</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 ASR 9000 Series Aggregation Services Router MPLS Configuration Guide, 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 ASR 9000 Series Aggregation Services 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 222</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>
</table>
| Bandwidth Reservation Percentage     | This feature was introduced.           | Release 4.3.1                 | *Implementing MPLS OAM* chapter:  
  Bandwidth Reservation Percentage, on page 65  
  Refer *RSVP Infrastructure Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference* for information on the commands used for configuring Bandwidth Reservation Percentage. |
| IP-Less MPLS-TP Ping and MPLS-TP Traceroute | This feature was introduced.           | Release 4.3.1                 | *Implementing MPLS OAM* chapter:  
  IP-Less MPLS-TP Ping and MPLS-TP Traceroute, on page 281  
  Refer *MPLS OAM Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference* for information on the commands used for configuring IP-Less MPLS-TP Ping and MPLS-TP Traceroute. |
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
</table>
| Label Security for BGP Inter-AS Option-B | This feature was introduced. | Release 4.3.1 | *Implementing MPLS Forwarding* chapter:  
Label Security for BGP Inter-AS Option-B, on page 102  
Refer *MPLS Forwarding Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference* for information on the commands used for configuring Label Security for BGP Inter-AS Option-B. |
| MPLS OAM Support for BGP 3107 | This feature was introduced. | Release 4.3.1 | *Implementing MPLS OAM* chapter:  
MPLS OAM Support for BGP 3107, on page 281  
Refer *MPLS Transport Profile Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference* for information on the commands used for configuring Label Security for BGP Inter-AS Option-B. |
| MPLS-TP LSP Wrapping | This feature was introduced. | Release 4.3.1 | *Implementing MPLS Transport Profile* chapter:  
MPLS-TP LSP Wrapping, on page 292  
Refer *MPLS Transport Profile Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference* for information on the commands used for configuring Label Security for BGP Inter-AS Option-B. |
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Introduced/Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
</table>
| Weighted-SRLG Auto-backup Path Computation | This feature was introduced. | Release 4.3.1 | *Implementing MPLS Traffic Engineering* chapter:  
Weighted-SRLG Auto-backup Path Computation, *on page 139*  
Refer *MPLS Traffic Engineering Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference* for information on the commands used for configuring Label Security for BGP Inter-AS Option-B. |
| GMPLS UNI               | This feature was introduced. | Release 4.3.0 | *Implementing GMPLS UNI* chapter:  
*Implementing GMPLS UNI, on page 257*  
Refer *GMPLS UNI Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference* for information on the commands used for configuring GMPLS UNI. |
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 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 4.0.1</td>
<td>Support was added for these features:</td>
</tr>
<tr>
<td></td>
<td>• IP LDP Fast Reroute Loop Free Alternate</td>
</tr>
<tr>
<td></td>
<td>• Downstream on Demand</td>
</tr>
<tr>
<td>Release 4.2.1</td>
<td>Support was added for LDP Implicit Null for IGP Routes.</td>
</tr>
<tr>
<td>Release 5.1.1</td>
<td>The feature MPLS LDP Carrier Supporting Carrier for Multiple VRFs was introduced.</td>
</tr>
<tr>
<td>Release 5.3.0</td>
<td>IPv6 Support in MPLS LDP was introduced.</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 20
- Configuring LDP Discovery Over a Link, on page 21
- Configuring LDP Link: Example, on page 49
- Configuring LDP Discovery for Active Targeted Hellos, on page 23
- Configuring LDP Discovery for Passive Targeted Hellos, on page 25
- Configuring LDP Discovery for Targeted Hellos: Example, on page 49

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 29
Configuring LDP Neighbors: Example, on page 50

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 31
- Configuring LDP Forwarding: Example, on page 51

LDP Graceful Restart

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

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

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

Reconnect time

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

Recovery time

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

FT flag

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

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

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

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

For more information about LDP commands, see *MPLS Label Distribution Protocol Commands* module of the *Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference*.

**Related Topics**
- Setting Up LDP NSF Using Graceful Restart, on page 32
- Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 51

**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 32
Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 51
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 32
- Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 51

**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 28
- Configuring Label Advertisement (Outbound Filtering): Example, on page 50

**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 35
- Configuring Label Acceptance (Inbound Filtering): Example, on page 51
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.

Tip

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 36
Configuring Local Label Allocation Control: Example, on page 52

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

---

1 For L3VPN Inter-AS option C, LDP may also be required to assign local labels for some BGP prefixes.
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.

Related Topics
- Configuring Session Protection, on page 37
- Configuring LDP Session Protection: Example, on page 52

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 38
- Configuring LDP IGP Synchronization—OSPF: Example, on page 52
- Configuring LDP IGP Synchronization: ISIS, on page 39
- Configuring LDP IGP Synchronization—ISIS: Example, on page 53

**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 53

**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)

**Note**

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 ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*.

**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 offailure.

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 Switching Router (LSR) running LDP must distribute its labels for the Forwarding Equivalence Classes (FECs) it can provide to all its neighbors, regardless of whether they are upstream, or not.

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.
- Cisco ASR 9000 Series SPA Interface Processor-700 POS line card on Cisco ASR 9000 Series Router is not supported as primary link. It can be used as LFA backup only on main interface.
- 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 ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

**Related Topics**

- Configure IP LDP Fast Reroute Loop Free Alternate: Examples, on page 54
- Verify IP LDP Fast Reroute Loop Free Alternate: Example, on page 55

**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

Explicit-Null and Implicit-Null Labels

Cisco MPLS LDP uses null label, implicit or explicit, as local label for routes or prefixes that terminate on the given LSR. These routes include all local, connected, and attached networks. By default, the null label is implicit-null that allows LDP control plane to implement penultimate hop popping (PHOP) mechanism. When this is not desirable, you can configure explicit-null that allows LDP control plane to implement ultimate hop popping (UHOP) mechanism. You can configure this explicit-null feature on the ultimate hop LSR. This configuration knob includes an access-list to specify the IP prefixes for which PHOP is desired.

This new enhancement allows you to configure implicit-null local label for non-egress (ultimate hop LSR) prefixes by using the implicit-null-override command. This enforces implicit-null local label for a specific prefix even if the prefix requires a non-null label to be allocated by default. For example, by default, an LSR allocates and advertises a non-null label for an IGP route. If you wish to terminate LSP for this route on penultimate hop of the LSR, you can enforce implicit-null label allocation and advertisement for this prefix using implicit-null-override feature.

Note
If a given prefix is permitted in both explicit-null and implicit-null-override feature, then implicit-null-override supercedes and an implicit-null label is allocated and advertised for the prefix.

In order to enable implicit-null-override mode, this configuration must be applied at MPLS LDP label configuration mode:

mpls ldp
label
    implicit-null-override for <prefix><ACL>

This feature works with any prefix including static, IGP, and BGP, when specified in the ACL.

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</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>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>mpls ldp</code></td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# <code>mpls ldp</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>[vrf vrf-name] router-id ip-address lsr-id</code></td>
<td>Specifies the router ID of the local node.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# <code>router-id 192.168.70.1</code></td>
<td>(Optional) Specifies a non-default VRF.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>`discovery { hello</td>
<td>targeted-hello } holdtime seconds`</td>
</tr>
</tbody>
</table>
| **Example:** | RP/0/RSP0/CPU0:router(config-ldp)# `discovery hello holdtime 30`  
RP/0/RSP0/CPU0:router(config-ldp)# `discovery targeted-hello holdtime 180` | |
### Purpose

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:**

```
RP/0/RSP0/CPU0# discovery hello interval 15
RP/0/RSP0/CPU0# discovery targeted-hello interval 20
```

### Step 6

**commit**

### Step 7

**show mpls ldp [vrf vrf-name] parameters**

(Optional)

Displays all the current MPLS LDP parameters.

Displays the LDP parameters for the specified VRF.

**Example:**

```
RP/0/RSP0/CPU0# show mpls ldp parameters
RP/0/RSP0/CPU0# show mpls ldp vrf red parameters
```

### Related Topics

- LDP Control Plane, on page 7

### 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>Step 1</td>
<td>configure</td>
<td>Enters MPLS LDP configuration mode.</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><code>RP/0/RSP0/CPU0:router(config)# mpls ldp</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>[vrf vrf-name] router-id ip-address lsr-id</code></td>
<td>(Optional) Specifies a non-default VRF.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Specifies the router ID of the local node.</td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-idp)# router-id 192.168.70.1</code></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>Step 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><code>RP/0/RSP0/CPU0:router(config-idp)# interface tunnel-te 12001</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-idp-if)#</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>show mpls ldp discovery</code></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/recv 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><code>RP/0/RSP0/CPU0:router# show mpls ldp discovery</code></td>
<td></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>7</td>
<td><code>show mpls ldp vrf vrf-name discovery</code></td>
<td>(Optional) Displays the status of the LDP discovery process for the specified VRF.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# <code>show mpls ldp vrf red discovery</code></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>show mpls ldp vrf all discovery summary</code></td>
<td>(Optional) Displays the summarized status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# <code>show mpls ldp vrf all discovery summary</code></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><code>show mpls ldp vrf all discovery brief</code></td>
<td>(Optional) Displays the brief status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# <code>show mpls ldp vrf all discovery brief</code></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><code>show mpls ldp vrf all ipv4 discovery summary</code></td>
<td>(Optional) Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# <code>show mpls ldp vrf all ipv4 discovery summary</code></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><code>show mpls ldp discovery summary all</code></td>
<td>(Optional) Displays the aggregate summary across all the LDP discovery processes.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# <code>show mpls ldp discovery summary all</code></td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

- LDP Control Plane, on page 7
- Configuring LDP Link: Example, on page 49

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.

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>Step 1</td>
<td>configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls ldp</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td><strong>RP/0/RSP0/CPU0:router(config)# mpls ldp</strong></td>
</tr>
<tr>
<td>Step 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>Example:</td>
<td></td>
<td><strong>RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.168.70.1</strong></td>
</tr>
<tr>
<td>Step 4</td>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode for the LDP protocol.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td><strong>RP/0/RSP0/CPU0:router(config-ldp)# interface tunnel-te 12001</strong></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 6**

**show mpls ldp discovery**

*Example:*

```
RP/0/RSP0/CPU0:router# show mpls ldp discovery
```

**Purpose**

(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.

**Step 7**

**show mpls ldp vrf vrf-name discovery**

*Example:*

```
RP/0/RSP0/CPU0:router# show mpls ldp vrf red discovery
```

**Purpose**

(Optional)

Displays the status of the LDP discovery process for the specified VRF.

**Step 8**

**show mpls ldp vrf all discovery summary**

*Example:*

```
RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery summary
```

**Purpose**

(Optional)

Displays the summarized status of the LDP discovery process for all VRFs.

**Step 9**

**show mpls ldp vrf all discovery brief**

*Example:*

```
RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery brief
```

**Purpose**

(Optional)

Displays the brief status of the LDP discovery process for all VRFs.

**Step 10**

**show mpls ldp vrf all ipv4 discovery summary**

*Example:*

```
RP/0/RSP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary
```

**Purpose**

(Optional)

Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.

**Step 11**

**show mpls ldp discovery summary all**

*Example:*

```
RP/0/RSP0/CPU0:router# show mpls ldp discovery summary all
```

**Purpose**

(Optional)

Displays the aggregate summary across all the LDP discovery processes.

### Related Topics

- LDP Control Plane, on page 7
- Configuring LDP Discovery for Targeted Hellos: Example, on page 49

### 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.

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>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 MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>[vrf vrf-name] router-id ip-address lsr-id</td>
</tr>
<tr>
<td>Example:</td>
<td>(Optional) Specifies a non-default VRF. Specifies the router ID of the local node.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.168.70.1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>discovery targeted-hello accept</td>
</tr>
<tr>
<td>Example:</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>RP/0/RSP0/CPU0:router(config-ldp)# discovery targeted-hello accept</td>
<td></td>
</tr>
</tbody>
</table>

- In Cisco IOS XR software, the router ID is specified as an interface IP address or LSR ID. By default, LDP uses the global router ID (configured by global router ID process).

- This command is executed on the receiver node (with respect to a given MPLS TE tunnel).
### Command or Action | Purpose
---|---
| | • You can control the targeted-hello acceptance using the **discovery targeted-hello accept** command.

**Step 5**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>commit</strong></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>
</tbody>
</table>

**Step 6**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show mpls ldp discovery</strong></td>
<td>(Optional) Displays the status of the LDP discovery process for the specified VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# show mpls ldp discovery</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 7**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show mpls ldp vrf vrf-name discovery</strong></td>
<td>(Optional) Displays the status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# show mpls ldp vrf red discovery</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 8**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show mpls ldp vrf all discovery summary</strong></td>
<td>(Optional) Displays the summarized status of the LDP discovery process for all VRFs.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery summary</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 9**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show mpls ldp vrf all discovery brief</strong></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><code>RP/0/RSP0/CPU0:router# show mpls ldp vrf all discovery brief</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 10**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show mpls ldp vrf all ipv4 discovery summary</strong></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><code>RP/0/RSP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 11**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show mpls ldp discovery summary all</strong></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><code>RP/0/RSP0/CPU0:router# show mpls ldp discovery summary all</code></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- LDP Control Plane, on page 7
- Configuring LDP Discovery for Targeted Hellos: Example, on page 49
Configuring Label Advertisement Control (Outbound Filtering)

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

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>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><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# mpls ldp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>label advertise { disable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# label advertise interface POS 0/1/0/0</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# for pfx_acl1 to peer_acl1</td>
</tr>
</tbody>
</table>

Configures label advertisement by specifying one of the following options:

- **disable**
  Disables label advertisement to all peers for all prefixes (if there are no other conflicting rules).

- **interface**
  Specifies an interface for label advertisement of an interface address.
### Command or Action | Purpose
--- | ---
| **for** prefix-acl **to** peer-acl | Specifies neighbors to advertise and receive label advertisements.

**Step 4**: commit

### Related Topics
- Label Advertisement Control (Outbound Filtering), on page 13
- Configuring Label Advertisement (Outbound Filtering): Example, on page 50

### 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

| Command or Action | Purpose |
--- | --- |
<p>| <strong>Step 1</strong>: configure |  |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>mpls ldp</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# mpls ldp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode for the LDP protocol.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# interface POS 0/1/0/0</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>discovery transport-address [ ip-address</td>
<td>Provides an alternative transport address for a TCP connection.</td>
</tr>
<tr>
<td>interface ]</td>
<td>- Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID.</td>
</tr>
<tr>
<td>Example:</td>
<td>- Transport address configuration is applied for a given LDP-enabled interface.</td>
</tr>
<tr>
<td>or</td>
<td>- 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.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp-if-if-af)# discovery transport-address interface</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ldp-if)# exit</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td>holdtime seconds</td>
<td>Changes the time for which an LDP session is maintained in the absence of LDP messages from the peer.</td>
</tr>
<tr>
<td>Example:</td>
<td>- Outgoing keepalive interval is adjusted accordingly (to make three keepalives in a given holdtime) with a change in session holdtime value.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp)# holdtime 30</td>
<td>- Session holdtime is also exchanged when the session is established.</td>
</tr>
<tr>
<td></td>
<td>- 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.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td>neighbor ip-address password [ encryption ] password</td>
<td>Configures password authentication (using the TCP MD5 option) for a given neighbor.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# neighbor 192.168.2.44 password secretpasswd</td>
</tr>
</tbody>
</table>
### Purpose

**Step 8**

**backoff**  *initial maximum*

**Example:**

```
RP/0/RSP0/CPU0:router(config-ldp)# backoff 10 20
```

**Purpose**

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.

**Step 9**

**commit**

(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 10**

**show mpls ldp neighbor**

**Example:**

```
RP/0/RSP0/CPU0:router# show mpls ldp neighbor
```

### Related Topics

*Configuring LDP Neighbors: Example*, on page 50

### 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 MPLS 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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls ldp</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></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# mpls ldp</td>
</tr>
<tr>
<td><strong>Step 3</strong> explicit-null</td>
<td>(Optional) Displays the MPLS LDP view of installed forwarding states (rewrites).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ldp-af)# explicit-null</td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</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><strong>Step 5</strong> show mpls ldp forwarding</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# show mpls ldp forwarding</td>
</tr>
<tr>
<td><strong>Step 6</strong> show mpls forwarding</td>
<td>(Optional) Checks for connectivity to a particular IP address (going through MPLS LSP as shown in the show mpls forwarding command).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# show mpls forwarding</td>
</tr>
<tr>
<td><strong>Step 7</strong> ping ip-address</td>
<td>(Optional) Checks for connectivity to a particular IP address (going through MPLS LSP as shown in the show mpls forwarding command).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# ping 192.168.2.55</td>
</tr>
</tbody>
</table>

**Related Topics**

- LDP Forwarding, on page 8
- Configuring LDP Forwarding: Example, on page 51

### 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><strong>Step 1</strong> configure</td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls ldp</td>
<td>Enters interface configuration mode for the LDP protocol.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface type interface-path-id</td>
<td>Enlists the current configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp)# interface POS 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Enables the LDP graceful restart feature.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> graceful-restart</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp)# graceful-restart</td>
<td></td>
</tr>
</tbody>
</table>
### Setting Up LDP NSF Using Graceful Restart

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>graceful-restart forwarding-state-holdtime</strong> seconds</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# graceful-restart forwarding-state-holdtime 180</td>
</tr>
<tr>
<td></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></td>
<td>• After restart of the control plane, when the forwarding state holdtime expires, any previously installed LDP forwarding state or rewrite that is not yet refreshed is deleted from the forwarding.</td>
</tr>
<tr>
<td></td>
<td>• Recovery time sent after restart is computed as the current remaining value of the forwarding state hold timer.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>graceful-restart reconnect-timeout</strong> seconds</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# graceful-restart reconnect-timeout 169</td>
</tr>
<tr>
<td></td>
<td>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 <strong>Neighbor Liveness</strong> timer.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>commit</strong></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>show mpls ldp parameters</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# show mpls ldp parameters</td>
</tr>
<tr>
<td></td>
<td>(Optional) Displays all the current MPLS LDP parameters.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>show mpls ldp neighbor</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# show mpls ldp neighbor</td>
</tr>
<tr>
<td></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>show mpls ldp graceful-restart</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# show mpls ldp graceful-restart</td>
</tr>
<tr>
<td></td>
<td>(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.</td>
</tr>
</tbody>
</table>

**Related Topics**

- LDP Graceful Restart, on page 9
- Phases in Graceful Restart, on page 11
- Recovery with Graceful-Restart, on page 12
- Configuring LDP Nonstop Forwarding with Graceful Restart: Example, on page 51
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>Step 1configure</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls ldp</td>
<td>Configures inbound label acceptance for prefixes specified by prefix-acl from neighbor (as specified by its IP address).</td>
</tr>
<tr>
<td>Example: RP/O/RSP0/CPU0:router(config)# mpls ldp</td>
<td>(Optional) Specifies a non-default VRF. Enables the LDP IPv4 or IPv6 address family.</td>
</tr>
<tr>
<td>Step 3 label accept for prefix-acl from ip-address</td>
<td>Enables the LDP IPv4 or IPv6 address family.</td>
</tr>
<tr>
<td>Example: RP/O/RSP0/CPU0:router(config-ldp)# label accept for pfx_acl_1 from 192.168.1.1 RP/O/RSP0/CPU0:router(config-ldp)# label accept for pfx_acl_2 from 192.168.2.2</td>
<td>Enables the LDP IPv4 or IPv6 address family.</td>
</tr>
</tbody>
</table>
### 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>
</tbody>
</table>

### Configuring Inbound Label Acceptance Control

Configures inbound label acceptance control for prefixes specified by prefix-acl from neighbor (as specified by its LDP ID).

**Example:**

```plaintext
RP/0/RSP0/CPU0:router(config-ldp-af)# label remote accept from 192.168.1.1:0 for pfx_acl_1
```

### Step 6 commit

**Related Topics**

- Label Acceptance Control (Inbound Filtering), on page 13
- Configuring Label Acceptance (Inbound Filtering): 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><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>mpls ldp</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config) # mpls ldp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>session protection [ for peer-acl ] [ duration seconds ]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-ldp) # session protection for peer_acl_1 duration 60</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>
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/RSP0/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>
</tbody>
</table>
### Configuring LDP IGP Synchronization: ISIS

Perform this task to configure LDP IGP Synchronization under ISIS.

**Note**

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>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>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# mpls ldp sync
```

<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>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# router isis 100
RP/0/RSP0/CPU0:router(config-isis)#
```

By default, there is no synchronization between LDP and ISIS.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>interface type interface-path-id</td>
<td>Configures the IS-IS protocol on an interface and enters ISIS interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis)#</td>
<td>interface POS 0/2/0/0</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-if)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>address-family {ipv4} unicast</td>
<td>Enters address family configuration mode for configuring IS-IS routing for a standard IP version 4 (IPv4) address prefix.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-if)#</td>
<td>address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-if-af)#</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>mpls ldp sync</td>
<td>Enables LDP IGP synchronization.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-isis-if-af)#</td>
<td>mpls ldp sync</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- IGP Synchronization, on page 15
- Configuring LDP IGP Synchronization—ISIS: Example, on page 53

### 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

DETAIL 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  process-name</td>
<td>Enables LDP auto-configuration.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# router ospf 190 RP/0/RSP0/CPU0:router(config-ospf)#</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: RP/0/RSP0/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: RP/0/RSP0/CPU0:router(config-ospf)# area 8</td>
<td>area-id 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: RP/0/RSP0/CPU0:router(config-ospf-ar)# interface pos 0/6/0/0</td>
<td>Note 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>

Related Topics

IGP Auto-configuration, on page 16
Configuring LDP Auto-Configuration: Example, on page 53
Disabling LDP Auto-Configuration, on page 43
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></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><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# router ospf 100 RP/0/RSP0/CPU0:router(config-ospf)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> area area-id</td>
<td>Configures an OSPF area and identifier.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf)# area 8 RP/0/RSP0/CPU0:router(config-ospf-ar)#</td>
<td>area-id Either a decimal value or an IP address.</td>
</tr>
<tr>
<td><strong>Step 4</strong> mpls ldp auto-config</td>
<td>Enables LDP auto-configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# mpls ldp auto-config</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> interface type interface-path-id</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-ospf-ar)# interface</td>
<td></td>
</tr>
</tbody>
</table>
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>Step 1 configure</td>
<td>Enters the MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls ldp</td>
<td>Enters interface configuration mode and configures an interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 interface type interface-path-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-ldp)# interface pos 0/6/0/0</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><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>mpls ldp</code></td>
</tr>
<tr>
<td>Example:</td>
<td><code>RP/0/RSP0/CPU0:router(config)# mpls ldp</code></td>
</tr>
</tbody>
</table>

**Related Topics**

- IGPAuto-configuration, on page 16
- Configuring LDP Auto-Configuration: Example, on page 53
### 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><code>configure</code></td>
<td></td>
</tr>
<tr>
<td><code>mpls ldp</code></td>
<td></td>
</tr>
<tr>
<td><code>[vrf vrf-name session] downstream-on-demand</code></td>
<td></td>
</tr>
<tr>
<td><code>commit</code></td>
<td></td>
</tr>
</tbody>
</table>
### Setting Up Implicit-Null-Override Label

Perform this task to configure implicit-null label for non-egress prefixes.

#### SUMMARY STEPS

1. configure
2. mpls ldp
3. label
4. **implicit-null-override for** *access-list*
5. commit

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <strong>configure</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <strong>mpls ldp</strong></td>
<td>Enters MPLS LDP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls ldp</td>
<td></td>
</tr>
</tbody>
</table>

(Optional) Enters downstream on demand label advertisement mode under the specified non-default VRF.

Enters downstream on demand label advertisement mode. The ACL contains the list of peer IDs that are configured for downstream-on-demand mode. When the ACL is changed or configured, the list of established neighbor is traversed.

### Related Topics

- Downstream on Demand, on page 18
### Purpose

Configure the allocation, advertisement, and acceptance of labels.

#### Command or Action

**Step 3**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>Configures the allocation, advertisement, and acceptance of labels.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-ldp-af)# label

#### Step 4

**Command or Action**

**Step 4**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>implicit-null-override for access-list</td>
<td>Configures implicit-null local label for non-egress prefixes.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-ldp-af-lbl)# implicit-null-override for 70

**Note:** This feature works with any prefix including static, IGP, and BGP, when specified in the ACL.

#### Step 5

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

---

### 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>Enters Global Configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** | Enters MPLS LDP configuration mode. |
| mpls ldp | |
| **Example:** | |
| RP/0/RSP0/CPU0:router(config)# mpls ldp | |

| **Step 3** | Allows the redistribution of BGP routes into an MPLS LDP processes. |
| redistribute bgp | |
### 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.

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

---

### Autonomous System Numbers (ASNs)

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.

**Example:**

```
RP/0/RSP0/CPU0:router(config-ldp)#
redistribute bgp advertise-to acl_1
```
Configuring LDP Discovery: Example

The example shows how to configure LDP discovery parameters.

```bash
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.

```bash
mpls ldp
  interface pos 0/1/0/0
!
!
show mpls ldp discovery
```

Related Topics

- Configuring LDP Discovery Over a Link, on page 21
- LDP Control Plane, on page 7

Configuring LDP Discovery for Targeted Hellos: Example

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

Active (tunnel head)

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

Passive (tunnel tail)

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

Related Topics

- Configuring LDP Discovery for Active Targeted Hellos, on page 23
- Configuring LDP Discovery for Passive Targeted Hellos, on page 25
LDP Control Plane, on page 7

Configuring Label Advertisement (Outbound Filtering): Example

The example shows how to configure LDP label advertisement control.

```plaintext
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

show mpls ldp binding
```

Related Topics

Configuring Label Advertisement Control (Outbound Filtering), on page 28
Label Advertisement Control (Outbound Filtering), on page 13

Configuring LDP Neighbors: Example

The example shows how to disable label advertisement.

```plaintext
mpls ldp
    router-id 192.168.70.1
    neighbor 1.1.1.1 password encrypted 110A1016141E
    neighbor 2.2.2.2 implicit-withdraw
```

Related Topics

Setting Up LDP Neighbors, on page 29
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 Forwarding, on page 31
- LDP Forwarding, on page 8

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 32
- LDP Graceful Restart, on page 9
- Phases in Graceful Restart, on page 11
- Recovery with Graceful-Restart, on page 12

Configuring Label Acceptance (Inbound Filtering): Example

The example shows how to configure inbound label filtering.

```
mpls ldp
  label
  accept
    for prefix acl 2 from 192.168.2.2
!
!
```

Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide, Release 4.3.x
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 35
  Label Acceptance Control (Inbound Filtering), on page 13

Configuring Local Label Allocation Control: Example

The example shows how to configure local label allocation control.

mpls ldp
  label
  allocate for pfx_acl_1
  !

Related Topics
  Configuring Local Label Allocation Control, on page 36
  Local Label Allocation Control, on page 14

Configuring LDP Session Protection: Example

The example shows how to configure session protection.

mpls ldp
  session protection duration 60 for peer_acl_1
  !

Related Topics
  Configuring Session Protection, on page 37
  Session Protection, on page 14

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 38
- IGP Synchronization, on page 15

## 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 39
- IGP Synchronization, on page 15

## 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 16
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 10

Sample configuration:

cisco asr 9000 series aggregation services router mpls configuration guide, release 4.3.x

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
downstream index 10
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 17

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/RSP0/CPU0:router#show isis fast-reroute summary

**IS-IS 1 IPv4 Unicast FRR summary**

<table>
<thead>
<tr>
<th>Prefixes reachable in L1</th>
<th>Critical Priority</th>
<th>High Priority</th>
<th>Medium Priority</th>
<th>Low Priority</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>All paths protected</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1008</td>
<td>1012</td>
</tr>
<tr>
<td>Some paths protected</td>
<td>0</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>
<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>
<td>100.00%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prefixes reachable in L2</th>
<th>Critical Priority</th>
<th>High Priority</th>
<th>Medium Priority</th>
<th>Low Priority</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>All paths protected</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Some paths protected</td>
<td>0</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>
<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>
<td>100.00%</td>
</tr>
</tbody>
</table>

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

RP/0/RSP0/CPU0:router#show isis fast-reroute 211.1.1.1/24
The following examples show how to verify the IGP route 211.1.1.1/24 in RIB output:

```
RP/0/RSP0/RSP0/CPU0#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/RSP0/RSP0/CPU0#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
    [387 type 5 flags 0x101001 (0x9d132398) 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 0x0]
      path-idx 0, bkup-idx 1 [0x9e5b71b4 0x0]
      next 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
        next hop 14.0.2.2
        remote adjacency
          local label 16080 labels imposed {16079}
```

```
RP/0/RSP0/RSP0/CPU0#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
    [387 type 5 flags 0x101001 (0x9d132398) 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 0x0]
      path-idx 0, bkup-idx 1 [0x9e5b71b4 0x0]
      next 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
        next hop 14.0.2.2
        remote adjacency
          local label 16080 labels imposed {16079}
```
The following examples show how to verify the IGP route 211.1.1.1/24 in MPLS LDP output:

RP/0/RSP0/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 Interface</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>16079</td>
<td></td>
<td></td>
<td>Gi0/6/0/0.3</td>
<td>14.0.2.2</td>
<td>(!)</td>
<td>Y</td>
</tr>
</tbody>
</table>

RP/0/RSP0/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 Interface</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>
</tbody>
</table>
  [ Protected; path-id 1 backup-path-id 33; peer 20.20.20.20:0 ]
| 16079       |       |       | Gi0/6/0/0.3        | 14.0.2.2 | (!) | Y     |
  [ Backup; path-id 33; peer 40.40.40.40:0 ]

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 17

Additional References

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

**Related Documents**

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

This module describes how to implement Resource Reservation Protocol (RSVP) for MPLS Traffic Engineering (MPLS-TE) on Cisco ASR 9000 Series Aggregation Services Routers.

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) uses RSVP to signal label switched paths (LSPs).

Feature History for Implementing RSVP for MPLS-TE

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>This feature was introduced.</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, page 60
- Information About Implementing RSVP for MPLS-TE, page 60
- Information About Implementing RSVP Authentication, page 65
- How to Implement RSVP, page 70
- How to Implement RSVP Authentication, page 78
- Configuration Examples for RSVP, page 88
- Configuration Examples for RSVP Authentication, page 93
- Additional References, page 95
Prerequisites for Implementing RSVP for MPLS-TE

These prerequisites are required to implement RSVP for 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.
- Either a composite mini-image plus an MPLS package, or a full image, must be installed.

Information About Implementing RSVP for MPLS-TE

To implement MPLS RSVP, you must understand these concepts:

Related Topics
- How to Implement RSVP Authentication, on page 78

Overview of RSVP for MPLS-TE

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

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.

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 73
- Set DSCP for RSVP Packets: Example, on page 92
- Verifying RSVP Configuration, on page 74
LSP Setup

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

**Figure 6: RSVP Operation**

![RSVP Operation Diagram]

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 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 71
- 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 send 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 72
- 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 77
- Enable RSVP Traps: Example, on page 92
Bandwidth Reservation Percentage

The Bandwidth Reservation Percentage allows the RSVP interface bandwidth to be specified as percentages of the link’s physical bandwidth.

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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide.

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.
- key-source key-chain command is set to none or disabled.

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 81
- RSVP Authentication by Using All the Modes: Example, on page 94

## 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 form the key-source) that was last used.</td>
</tr>
<tr>
<td>digest</td>
<td>Algorithm last used (returned from the key-source).</td>
</tr>
<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 window size 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>

**Related Topics**

- [Specifying the Keychain for RSVP Neighbor Authentication](#), on page 85
- [RSVP Neighbor Authentication: Example](#), on page 94
- [Configuring a Lifetime for RSVP Neighbor Authentication](#), on page 86
- [RSVP Authentication Global Configuration Mode: Example](#), on page 93

**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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide*.

**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 re-order 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 159
Configuring an IETF DS-TE Tunnel Using RDM, on page 161
Configuring an IETF DS-TE Tunnel Using MAM, on page 163

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>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</td>
<td>Enters RSVP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# rsvp</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
<td>Enters interface configuration mode for the RSVP protocol.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-rsvp)# interface pos 0/2/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>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>RP/0/RSP0/CPU0:router(config-rsvp-if)# bandwidth 1000 100 sub-pool 150</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- Differentiated Services Traffic Engineering, on page 113
- Bandwidth Configuration (MAM): Example, on page 89
- Bandwidth Configuration (RDM): Example, on page 89

### 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><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> rsvp</td>
<td>Enters the RSVP configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# rsvp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> signalling graceful-restart</td>
<td>Enables the graceful restart process on the node.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-rsvp)# signalling graceful-restart</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> signalling graceful-restart interface-based</td>
<td>Enables interface-based graceful restart process on the node.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-rsvp)# signalling graceful-restart interface-based</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> 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 72
- Configuring RSVP Packet Dropping, on page 73

### 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.
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>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</td>
<td>Enters the RSVP configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# rsvp</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>signalling prefix-filtering access-list</td>
<td>Enter an extended access list name as a string.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp)# signalling</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prefix-filtering access-list banks</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- ACL-based Prefix Filtering, on page 64
- Configure ACL-based Prefix Filtering: Example, on page 92

**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>Enters the RSVP configuration mode.</td>
</tr>
<tr>
<td>Step 2 rsvp</td>
<td>Example: RP/0/RSP0/CPU0:router(config)# rsvp</td>
</tr>
<tr>
<td>Step 3 signalling prefix-filtering default-deny-action</td>
<td>Example: RP/0/RSP0/CPU0:router(config-rsvp)# signalling prefix-filtering default-deny-action</td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Overview of RSVP for MPLS-TE, 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
deaf
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/RSP0/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) 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/RSP0/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/RSP0/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
Path states 0 Expired Path states 0 Expired Resv states 0 Expired Resv states 0 NACKs received 0 POS0/3/0/0 Expired Path states 0 Expired Path states 0 Expired Resv states 0 Expired Resv states 1 NACKs received 0 NACKs received 0 POS0/3/0/2
```

Step 4 show rsvp interface type interface-path-id [detail]
Verifies that refresh reduction is working on a particular interface.

Example:

```
RP/0/RSP0/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/RSP0/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/RSP0/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 19:01:49 0
RP/0/RSP0/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/RSP0/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%) 0 PO0/3/0/1 1000M 1000M 0 ( 0%) 0 PO0/3/0/2 1000M 1000M 0 ( 0%) 0 PO0/3/0/3 1000M 1000M 1K ( 0%) 0
```

Step 8  show rsvp neighbor
Verifies the RSVP neighbors.

Example:

```
RP/0/RSP0/CPU0:router# show rsvp neighbor detail
Global Neighbor: 40.40.40.40 Interface Neighbor: 1.1.1.1
Interface: POS0/0/0/0 Refresh Reduction: "Enabled" or "Disabled". Remote epoch: 0xXXXXXXXX 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: 0xXXXXXXXX Out of order messages: 0 Retransmitted messages: 0
```

Related Topics
Overview of RSVP for MPLS-TE, on page 60

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>Sends RSVP notifications to enable RSVP LostFlow traps.</td>
</tr>
<tr>
<td>Step 2</td>
<td>snmp-server traps rsvp lost-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/RSP0/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 all RSVP MIB traps.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/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></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/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>

**Related Topics**

- RSVP MIB, on page 64
- Enable RSVP Traps: Example, on page 92

**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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide).

SUMMARY STEPS

1. configure
2. rsvp authentication
3. key-source key-chain key-chain-name
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 RSVP authentication configuration mode.</td>
</tr>
<tr>
<td>Step 2 rsvp authentication</td>
<td></td>
</tr>
<tr>
<td>Example: RP/0/RSP0:CPU0:router(config)# rsvp authentication RP/0/RSP0:CPU0:router(config-rsvp-auth)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 key-source key-chain key-chain-name</td>
<td>Specifies the source of the key information to authenticate RSVP signaling messages. key-chain-name Name of the keychain. The maximum number of characters is 32.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0:CPU0:router(config-rsvp-auth)# key-source key-chain mpls-keys</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
Key-source Key-chain, on page 68
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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 rsvp authentication</td>
<td>Enters RSVP authentication configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# rsvp</td>
<td></td>
</tr>
<tr>
<td>authentication</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-auth)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 life-time seconds</td>
<td>Controls how long RSVP maintains security associations with other trusted RSVP neighbors. seconds</td>
</tr>
<tr>
<td>Example:</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>RP/0/RSP0/CPU0:router(config-rsvp-auth)# life-time 2000</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

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>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# rsvp authentication RP/0/RSP0/CPU0:router(config-rsvp-auth)#</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>window-size $N$</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-auth)# window-size 33</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</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 94
RSVP Authentication for an Interface: Example, on page 94

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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide).
### 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>Enters RSVP interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>rsvp interface type interface-path-id</code></td>
<td>Enters RSVP authentication configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# <code>rsvp interface POS 0/2/1/0</code></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>authentication</code></td>
<td>Specifies the source of the key information to authenticate RSVP signaling messages.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if)# <code>authentication</code></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if-auth)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>key-source key-chain key-chain-name</code></td>
<td>Name of the keychain. The maximum number of characters is 32.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if-auth)# <code>key-source key-chain mpls-keys</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>commit</code></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 94

### 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>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>rsvp interface type interface-path-id</td>
</tr>
<tr>
<td></td>
<td>Enters RSVP interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config)# rsvp interface</td>
</tr>
<tr>
<td></td>
<td>POS 0/2/1/0</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if)#</td>
</tr>
<tr>
<td>Step 3</td>
<td>authentication</td>
</tr>
<tr>
<td></td>
<td>Enters RSVP authentication configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-rsvp-if)#</td>
</tr>
<tr>
<td></td>
<td>authentication</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if-auth)#</td>
</tr>
<tr>
<td>Step 4</td>
<td>life-time seconds</td>
</tr>
<tr>
<td></td>
<td>Controls how long RSVP maintains security associations with other trusted RSVP neighbors. seconds</td>
</tr>
<tr>
<td></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></td>
<td>Example: RP/0/RSP0/CPU0:router(config-rsvp-if-auth)#</td>
</tr>
<tr>
<td></td>
<td>life-time 2000</td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
</tr>
</tbody>
</table>

Related Topics

RSVP Authentication Design, on page 65
RSVP Authentication by Using All the Modes: Example, on page 94
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>Step 2 rsvp interface type interface-path-d</td>
<td>Enters RSVP interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>POS 0/2/1/0</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)</td>
<td>rsvp interface #</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if)</td>
<td></td>
</tr>
<tr>
<td>Step 3 authentication</td>
<td>Enters RSVP interface authentication configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if-auth)</td>
<td>authentication #</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if-auth)</td>
<td></td>
</tr>
<tr>
<td>Step 4 window-size $N$</td>
<td>Specifies the maximum number of RSVP authenticated messages that can be received out-of-sequence.</td>
</tr>
<tr>
<td>Example:</td>
<td>$N$</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if-auth)</td>
<td>window-size 33</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if-auth)</td>
<td></td>
</tr>
<tr>
<td>Step 5 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 94
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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide).

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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters neighbor authentication configuration mode. Use the rsvp neighbor command to activate RSVP cryptographic authentication for a neighbor.</td>
</tr>
<tr>
<td><strong>Step 2</strong> rsvp neighbor IP-address authentication</td>
<td>IP address</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RSP0/CP00:router(config)# rsvp neighbor 1.1.1.1 authentication
RP/0/RSP0/CP00:router(config-rsvp-nbor-auth)#
```

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.
Configuring RSVP Neighbor Authentication

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 3 | key-source key-chain *key-chain-name*  
Example:  
RP/0/RSP0/CPU0:router(config-rsvp-nbor-auth)# key-source key-chain mpls-keys | Specifies the source of the key information to authenticate RSVP signaling messages.  
*key-chain-name*  
Name of the keychain. The maximum number of characters is 32. |
| Step 4 | commit |

Related Topics
- Key-source Key-chain, on page 68
- 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>Step 1</td>
<td>configure</td>
</tr>
</tbody>
</table>
| Step 2 | rsvp neighbor *IP-address authentication*  
Example:  
RP/0/RSP0/CPU0:router(config)# rsvp neighbor 1.1.1.1 authentication  
RP/0/RSP0/CPU0:router(config-rsvp-nbor-auth)# | Enters RSVP neighbor authentication configuration mode. Use the rsvp neighbor command to specify a neighbor under RSVP.  
*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. |
### Configuring 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 rsvp neighbor command to specify a neighbor under RSVP.</td>
</tr>
<tr>
<td><strong>Step 2</strong> rsvp neighbor IP address 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>
</tr>
<tr>
<td></td>
<td>authentication</td>
</tr>
<tr>
<td></td>
<td>Configures the RSVP authentication parameters.</td>
</tr>
</tbody>
</table>

---

**Related Topics**

- Security Association, on page 67
- RSVP Authentication Global Configuration Mode: Example, on page 93
### 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 89
- 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 92
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
```

The following example shows how to allocate a percentage of total bandwidth to bc0 and bc1 pools:

```
  rsvp interface pos 0/3/0/0
  bandwidth mam percentage 100 bc0 100 bc1 50
```

Related Topics

- Confirming DiffServ-TE Bandwidth, on page 70
- Differentiated Services Traffic Engineering, on page 113

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
```

The following example shows how to allocate a percentage of total bandwidth to bc0 and bc1 pools:

```
  rsvp interface pos 0/3/0/0
  bandwidth rdm percentage 100 bc0 100 bc1 50
```

Related Topics

- Confirming DiffServ-TE Bandwidth, on page 70
- Differentiated Services Traffic Engineering, on page 113
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
   signalling refresh reduction reliable ack-max-size 1000
```

Note

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.

```
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 0/0/0/1
  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 71
- 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/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config-rsvp)#interface bundle-ether 17
RP/0/RSP0/CPU0:router(config-rsvp-if)#signalling hello graceful-restart ?
  interface-based Configure Interface-based Hello
RP/0/RSP0/CPU0:router(config-rsvp-if)#signalling hello graceful-restart interface-based
RP/0/RSP0/CPU0:router(config-rsvp-if)#
```

Related Topics

- Enabling Graceful Restart, on page 71
- 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.

```bash
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 72
- 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 73
- Overview of RSVP for MPLS-TE, 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 77
- RSVP MIB, on page 64

## 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 94

### RSVP Authentication Global Configuration Mode: Example

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

```
rsvp
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 68
- Configuring a Lifetime for RSVP Authentication in Global Configuration Mode, on page 80
- Global, Interface, and Neighbor Authentication Modes, on page 66
- Configuring a Lifetime for RSVP Neighbor Authentication, on page 86
- Security Association, on page 67
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.

```bash
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 80
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.

```bash
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 68
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.

```bash
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
    !
    !
```

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 80
- 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 81
- Global, Interface, and Neighbor Authentication Modes, on page 66
- Configuring a Lifetime for an Interface for RSVP Authentication, on page 82
- 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 Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference</td>
</tr>
<tr>
<td>MPLS Traffic Engineering commands</td>
<td>MPLS Traffic Engineering commands module in Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference</td>
</tr>
</tbody>
</table>
## Related Topic

<table>
<thead>
<tr>
<th>RSVP commands</th>
<th><strong>RSVP commands</strong> module in <em>Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting started material</td>
<td><em>Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide</em></td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td><em>Configuring AAA Services</em> module in <em>Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide</em></td>
</tr>
</tbody>
</table>

## Standards

<table>
<thead>
<tr>
<th>Standard</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>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 3471</td>
<td>Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description</td>
</tr>
</tbody>
</table>
### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<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

This module describes how to implement MPLS Forwarding on Cisco ASR 9000 Series Aggregation Services Routers.

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 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for Implementing Cisco MPLS Forwarding, page 99
- Restrictions for Implementing Cisco MPLS Forwarding, page 100
- Information About Implementing MPLS Forwarding, page 100
- How to Implement MPLS Forwarding, page 102
- Additional References, page 103

**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

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:
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 * 4 bytes are added to the size of the packet when the packet is labeled. The MPLS MTU parameter pertains to labeled packets.

Label Security for BGP Inter-AS Option-B

Option-B is a method to exchange VPNv4/VPNv6 routes between Autonomous Systems (AS), as described in RFC-4364. When a router configured with Option-B, peers with a router from another confederation, or an autonomous system, and receives a labeled packet from such an external peer, the router ensures the following:

- the top label is advertised to the source of traffic
- label stack on the packet received from the external peer contains at least one label (explicit null label is not included)

How to Implement MPLS Forwarding

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

Configuring MPLS Label Security

Perform this task to configure the MPLS label security on the interface.

SUMMARY STEPS

1. `configure`
2. `interface type interface-path-id`
3. `mpls label-security rpf`
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 type interface-path-id</td>
<td>Enters the interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)#interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>mpls label-security rpf</td>
<td>Configures the MPLS label security on the specified interface and checks for RPF label on incoming packets.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)#mpls label-security rpf</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

Additional References

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

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS Forwarding commands</td>
<td>MLS Forwarding Commands on Cisco ASR 9000 Series Router module in Cisco ASR 9000 Series Aggregation Services Routers MPLS Command Reference</td>
</tr>
<tr>
<td>Getting started material</td>
<td>Cisco ASR 9000 Series Aggregation Services Routers Getting Started Guide</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS Forwarding commands</td>
<td>MLS Forwarding Commands on Cisco IOS XR Software module in Cisco IOS XR MPLS Command Reference for the Cisco ASR 9000 Series Router</td>
</tr>
<tr>
<td>Getting started material</td>
<td>Cisco IOS XR Getting Started Guide for the Cisco ASR 9000 Series Router</td>
</tr>
<tr>
<td>Related Topic</td>
<td>Document Title</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td><em>Configuring AAA Services on Cisco IOS XR Software</em> module of <em>Cisco IOS XR System Security Configuration Guide for the Cisco ASR 9000 Series Router</em></td>
</tr>
</tbody>
</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. Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</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>
Implementing MPLS Traffic Engineering

This module describes how to implement MPLS Traffic Engineering on Cisco ASR 9000 Series Router. 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 3.7.2</td>
<td>This feature was introduced.</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>The MPLS Traffic Engineering (TE): Path Protection feature was added.</td>
</tr>
<tr>
<td>Release 3.9.1</td>
<td>The MPLS-TE automatic bandwidth feature is supported.</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</td>
</tr>
<tr>
<td></td>
<td>MPLS-TE</td>
</tr>
<tr>
<td></td>
<td>• Point-to-Multipoint Traffic-Engineering</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.
- 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.

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<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</td>
</tr>
<tr>
<td></td>
<td>Co-existence of Path Protection with Fast Reroute</td>
</tr>
<tr>
<td></td>
<td>• P2MP-TE Inter-area Enhancements</td>
</tr>
<tr>
<td>Release 6.0.1</td>
<td>Point-to-Multipoint Implicit Null feature was added.</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, page 106
- Restrictions for Implementing GMPLS UNI, page 107
- Information About Implementing MPLS Traffic Engineering, page 107
- How to Implement Traffic Engineering, page 145
- Configuration Examples for Cisco MPLS-TE, page 231
- Additional References, page 255
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 ASR 9000 Enhanced Ethernet Line Cards:
  - A9K-MOD80-SE: 80G Modular Line Card, Service Edge Optimized
  - A9K-MOD80-TR: 80G Modular Line Card, Packet Transport Optimized

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.

Related Topics

- Configuring Forwarding over the MPLS-TE Tunnel, on page 149

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 145
- Creating an MPLS-TE Tunnel, on page 147
- Build MPLS-TE Topology and Tunnels: Example, on page 231

**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 155
- Removing an AutoTunnel Backup, on page 156
- Establishing MPLS Backup AutoTunnels to Protect Fast Reroutable TE LSPs, on page 157
- Establishing Next-Hop Tunnels with Link Protection, on page 158
- Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 242
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
```
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.

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 89

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

- Configuring a Prestandard DS-TE Tunnel, on page 159
- Configure IETF DS-TE Tunnels: Example, on page 233

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.

**Related Topics**

Configuring an IETF DS-TE Tunnel Using MAM, on page 163

**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 161

**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 to 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>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.

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.

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.
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.

Note

Related Topics

Protecting MPLS Tunnels with Fast Reroute, on page 151

MPLS-TE and Fast Reroute over Link Bundles

MPLS Traffic Engineering (TE) and Fast Reroute (FRR) are supported over bundle interfaces and virtual local area network (VLAN) interfaces. 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 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 and TenGigE).

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 ASR 9000 Series Aggregation Services Router MPLS Command Reference.

Related Topics

- Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE, on page 167
- Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 234

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.

Note

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.
Related Topics

Assigning Color Names to Numeric Values, on page 168
Associating Affinity-Names with TE Links, on page 169
Associating Affinity Constraints for TE Tunnels, on page 170
Configure Flexible Name-based Tunnel Constraints: Example, on page 234

MPLS Traffic Engineering Interarea Tunneling

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

- Interarea Support, on page 119
- Multiarea Support, on page 120
- Loose Hop Expansion, on page 120
- Loose Hop Reoptimization, on page 121
- Fast Reroute Node Protection, on page 121

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 the 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 151](#)

**MPLS-TE Forwarding Adjacency**

The MPLS-TE Forwarding Adjacency feature allows a network administrator to handle a 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 175
- Configure Forwarding Adjacency: Example, on page 237

**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)
- OSPF

**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 176
Configuring a Path Computation Element Address, on page 177
Configuring PCE Parameters, on page 178
Configure PCE: Example, on page 237
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.

This figure illustrates a PBTS implementation.

Figure 14: Policy-Based Tunnel Selection Implementation

PBTS is supported on the ingress interface and any of the L3 interfaces (physical, sub-interface, and bundle interface).

PBTS supports modification of the class-map and forward-group to TE association.

Policy-Based Tunnel Selection Functions

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.
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

- Enabling Path Protection for an Interface, on page 181
- Assigning a Dynamic Path Option to a Tunnel, on page 182
- Forcing a Manual Switchover on a Path-Protected Tunnel, on page 183
- Configuring the Delay the Tunnel Takes Before Reoptimization, on page 183
- Configure Tunnels for Path Protection: Example, on page 238

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.
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.
• Do not use link and node protection with path protection on the headend router.
• 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.

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
- Enabling Path Protection for an Interface, on page 181
- Assigning a Dynamic Path Option to a Tunnel, on page 182
- Forcing a Manual Switchover on a Path-Protected Tunnel, on page 183
- Configuring the Delay the Tunnel Takes Before Reoptimization, on page 183
- Configure Tunnels for Path Protection: Example, on page 238

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-FRRLSP 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.
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 184
- Configuring the Automatic Bandwidth Functions, on page 186
- Configure Automatic Bandwidth: Example, on page 239

**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 10kbps, the tunnel is signalled with 30 Mbps when the application timer expires. This is because 10% of 45Mbit/s is 4.5 Mbit/s, which is smaller than 15 Mbit/s. The absolute threshold, which by default is 10kbps, 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.

**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 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-TED tunnel, IP multicast traffic is encapsulated with a unique MPLS label, which is associated with the P2MP-TED 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-TED 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-TED 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.

Bud Node

 Specifies the node that not only acts as a transit for some S2Ls but also acts as a termination point for a S2L of a P2MP TE tunnel.

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) link 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 set up and signaling of P2MP-TE paths in the MPLS network.

Related Topics
Configure Point-to-Multipoint for the Source: Example, on page 250
Configure the Point-to-Multipoint Solution: Example, on page 252
Disable a Destination: Example, on page 251
Configure the Point-to-Multipoint Tunnel: Example, on page 251
Configure the Point-to-Multipoint Solution: Example, on page 252
Point-to-Multipoint RSVP-TE, on page 132

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
Configure Point-to-Multipoint for the Source: Example, on page 250
Configure the Point-to-Multipoint Solution: Example, on page 252
Point-to-Multipoint Fast Reroute, on page 133
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 130
Point-to-Multipoint RSVP-TE, on page 132

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.

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 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**

Configure the Point-to-Multipoint Tunnel: Example, on page 251
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.

**Figure 15: Shared Risk Link Group**

![Shared Risk Link Group Diagram]

**Related Topics**
- Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 189
- Creating an Explicit Path With Exclude SRLG, on page 191
- Using Explicit Path With Exclude SRLG, on page 192
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

**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 189
- Creating an Explicit Path With Exclude SRLG, on page 191
- Using Explicit Path With Exclude SRLG, on page 192
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

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.

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 189
- Creating an Explicit Path With Exclude SRLG, on page 191
- Using Explicit Path With Exclude SRLG, on page 192
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

**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 189
Creating an Explicit Path With Exclude SRLG, on page 191
Using Explicit Path With Exclude SRLG, on page 192
Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

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.

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 189
Creating an Explicit Path With Exclude SRLG, on page 191
Using Explicit Path With Exclude SRLG, on page 192
Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

Weighted-SRLG Auto-backup Path Computation

In shared-risk link groups (SRLG) fate-sharing, links are assigned one or more numbers to represent risks. When two links are assigned a common number then this indicates that these two links are sharing fate. In the weighted-SRLG auto-backup path computation mode, the links that share SRLG numbers with the protected link are not excluded from the topology. The admin-weight of these links is set to reflect the sharing of SRLG
with the protected link. Setting the admin weight consists of adding a penalty metric to make using the link less desirable.

For more information about Weighted-SRLG auto-backup path computation, see Implementing MPLS Traffic Engineering chapter in the Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide. For more information about Weighted-SRLG auto-backup path computation, see MPLS Traffic Engineering Commands chapter in the Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference.

Related Topics

- Configuring the SRLG Values of Each Link that has a Shared Risk with Another Link, on page 189
- Creating an Explicit Path With Exclude SRLG, on page 191
- Using Explicit Path With Exclude SRLG, on page 192
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

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 189
- Creating an Explicit Path With Exclude SRLG, on page 191
- Using Explicit Path With Exclude SRLG, on page 192
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

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 212
Enabling Soft-Preemption on a Tunnel, on page 213

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 214

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 0xC0C0 mask 0xC0C0
  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 214

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 214

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
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 214

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 214

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 216
- Configuring Auto-tunnel Mesh Unused Timeout, on page 217
- Configuring Auto-Tunnel Mesh Group, on page 218
- Configuring Tunnel Attribute-Set Templates, on page 219
- Enabling LDP on Auto-Tunnel Mesh, on page 221

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)
IGP Id: 0001.0000.0011.00, MPLS TE Id: 100.2.2.2 Router Node (ISIS 1 level-2)
IGP Id: 0001.0000.0012.00, MPLS TE Id: 100.3.3.3 Router Node (ISIS 1 level-2)

A prefix-list may be configured on each TE router to match a desired set of router IDs (MPLS TE ID as shown in the above output). For example, if a prefix-list is configured to match addresses of 100.0.0.0 with wildcard 0.255.255.255, then all 100.x.x.x router IDs are included in the auto-mesh group.

When a new TE router is added in the network and its router ID is also in the block of addresses described by the prefix-list, for example, 100.x.x.x, then it is added in the auto-mesh group on each existing TE router without having to explicitly modify the prefix-list or perform any additional configuration.

Auto-mesh does not create tunnels to its own (local) TE router IDs.

---

**Note**

When prefix-list configurations on all routers are not identical, it can result in non-symmetrical mesh of tunnels between those routers.

---

Related Topics

- Configuring Auto-Tunnel Mesh Tunnel ID, on page 216
- Configuring Auto-tunnel Mesh Unused Timeout, on page 217
- Configuring Auto-Tunnel Mesh Group, on page 218
- Configuring Tunnel Attribute-Set Templates, on page 219
- Enabling LDP on Auto-Tunnel Mesh, on page 221
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.

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 1</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
</tbody>
</table>

### Step 2

Enters MPLS-TE configuration mode.

**Example:**
```
RP/0/RSP0/CPU0:router(config)# mpls traffic-eng
RP/0/RSP0/CPU0:router(config-mpls-te)#
```

### Step 3

Enters interface configuration mode.

**Example:**
```
RP/0/RSP0/CPU0:router(config-mpls-te)# interface POS0/6/0/0
RP/0/RSP0/CPU0:router(config-mpls-te-if)#
```

### Step 4

Exits the current configuration mode.

**Example:**
```
RP/0/RSP0/CPU0:router(config-mpls-te-if)# exit
RP/0/RSP0/CPU0:router(config-mpls-te)#
```

### Step 5

Exits the current configuration mode.

**Example:**
```
RP/0/RSP0/CPU0:router(config-mpls-te)# exit
RP/0/RSP0/CPU0:router(config)#
```

### Step 6

Enters a name for the OSPF process.

**Example:**
```
RP/0/RSP0/CPU0:router(config)# router ospf 1
```

### Step 7

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

**Example:**
```
RP/0/RSP0/CPU0:router(config-router)# area 0
```

### Step 8

Exits the current configuration mode.

**Example:**
```
RP/0/RSP0/CPU0:router(config-ospf-ar)# exit
RP/0/RSP0/CPU0:router(config-ospf)#
```
## 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.

### Related Topics

- How MPLS-TE Works, on page 108
- Build MPLS-TE Topology and Tunnels: Example, on page 231
### 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>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> interface tunnel-te tunnel-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> destination ip-address</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# destination 192.168.92.125</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv4 unnumbered type interface-path-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> path-option preference - priority dynamic</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# path-option 1 dynamic</td>
<td></td>
</tr>
</tbody>
</table>
Purpose
Command or Action | Purpose
--- | ---
Step 6 | signalled-bandwidth \{bandwidth [class-type ct] | sub-pool bandwidth\}
Example: RP/0/RSP0/CPU0:router(config-if)# signalled-bandwidth 100
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).
Step 7 | commit
Step 8 | show mpls traffic-eng tunnels
Example: RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels
(Optional) Verifies that the tunnel is connected (in the UP state) and displays all configured TE tunnels.
Step 9 | show ipv4 interface brief
Example: RP/0/RSP0/CPU0:router# show ipv4 interface brief
(Optional) Displays all TE tunnel interfaces.
Step 10 | show mpls traffic-eng link-management admission-control
Example: RP/0/RSP0/CPU0:router# show mpls traffic-eng link-management admission-control
(Optional) Displays all the tunnels on this node.

Related Topics
- How MPLS-TE Works, on page 108
- Build MPLS-TE Topology and Tunnels: Example, on page 231
- Building MPLS-TE Topology, on page 145

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</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters MPLS-TE interface configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-te tunnel-id</td>
<td>Enters MPLS-TE interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>ipv4 unnumbered type interface-path-id</td>
<td>Assigns a source address so that forwarding can be performed on the new tunnel.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>autoroute announce</td>
<td>Enables messages that notify the neighbor nodes about the routes that are forwarding.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# autoroute announce</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# exit</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>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>
<td></td>
</tr>
<tr>
<td><code>router static address-family ipv4 unicast prefix mask ip-address interface type</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# router static address-family ipv4 unicast 2.2.2.2/32 tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>(Optional) Checks for connectivity to a particular IP address or host name.</td>
<td></td>
</tr>
<tr>
<td>`ping {ip-address</td>
<td>hostname}`</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# ping 192.168.12.52</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>(Optional) Verifies forwarding by displaying what is advertised to IGP for the TE tunnel.</td>
<td></td>
</tr>
<tr>
<td><code>show mpls traffic-eng autoroute</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# show mpls traffic-eng autoroute</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**
- Overview of MPLS Traffic Engineering, on page 107
- Creating an MPLS-TE Tunnel, on page 147

## 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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Purpose</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface. Example: RP/0/RSP0/CPU0:router# interface tunnel-te 1</td>
</tr>
<tr>
<td><strong>Step 3</strong> fast-reroute</td>
<td>Enables fast reroute. Example: RP/0/RSP0/CPU0:router(config-if)# fast-reroute</td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits the current configuration mode. Example: RP/0/RSP0/CPU0:router(config-if)# exit</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
</tr>
<tr>
<td>5</td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
</tr>
<tr>
<td>6</td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# interface pos0/6/0/0</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if)#</td>
</tr>
<tr>
<td>7</td>
<td>backup-path tunnel-te tunnel-number</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if)# backup-path tunnel-te 2</td>
</tr>
<tr>
<td>8</td>
<td>exit</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if)# exit</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
</tr>
<tr>
<td>9</td>
<td>exit</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# exit</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)#</td>
</tr>
<tr>
<td>10</td>
<td>interface tunnel-te tunnel-id</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te 2</td>
</tr>
<tr>
<td>11</td>
<td>backup-bw {backup bandwidth</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)#backup-bw</td>
</tr>
<tr>
<td></td>
<td>global-pool 5000</td>
</tr>
</tbody>
</table>

Note: Because the default tunnel priority is 7, tunnels use the default TE class map.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>ipv4 unnumbered</strong> type interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
</tr>
<tr>
<td></td>
<td>Assigns a source address to set up forwarding on the new tunnel.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>path-option</strong> preference-priority {explicit name explicit-path-name}</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# path-option 1 explicit name backup-path</td>
</tr>
<tr>
<td></td>
<td>Sets the path option to explicit with a given name (previously configured) and assigns the path ID.</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>destination</strong> ip-address</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# destination 192.168.92.125</td>
</tr>
<tr>
<td></td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td></td>
<td>• Destination address is the remote node's MPLS-TE router ID.</td>
</tr>
<tr>
<td></td>
<td>• Destination address is the merge point between backup and protected tunnels.</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><strong>Step 15</strong></td>
<td>commit</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>show mpls traffic-eng tunnels backup</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels backup</td>
</tr>
<tr>
<td>(Optional)</td>
<td>Displays the backup tunnel information.</td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td>show mpls traffic-eng tunnels protection frr</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels protection frr</td>
</tr>
<tr>
<td>(Optional)</td>
<td>Displays the tunnel protection information for Fast-Reroute (FRR).</td>
</tr>
<tr>
<td><strong>Step 18</strong></td>
<td>show mpls traffic-eng fast-reroute database</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show mpls traffic-eng fast-reroute database</td>
</tr>
<tr>
<td>(Optional)</td>
<td>Displays the protected tunnel state (for example, the tunnel’s current ready or active state).</td>
</tr>
</tbody>
</table>
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 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><strong>Step 1</strong> configure</td>
<td>Configures the globally configured IPv4 address that can be used by the AutoTunnel Backup Tunnels.</td>
</tr>
<tr>
<td><strong>Step 2</strong> 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><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)#ipv4 unnumbered mpls traffic-eng Loopback 0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> 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><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-mpls-te)# auto-tunnel backup timers removal unused 20</td>
<td></td>
</tr>
</tbody>
</table>
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>clears all MPLS-TE automatic backup tunnels from the EXEC mode. You can also remove the automatic backup tunnel marked with specific tunnel-te, provided it is currently unused.</td>
</tr>
<tr>
<td>clear mpls traffic-eng auto-tunnel backup unused { all</td>
<td>tunnel-te number }</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/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>displays information about MPLS-TE autotunnels including the ones removed.</td>
</tr>
<tr>
<td>show mpls traffic-eng auto-tunnel summary</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show mpls traffic-eng auto-tunnel summary</td>
</tr>
</tbody>
</table>
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><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>Enables traffic engineering on a specific interface on the originating node.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface type interface-path-id</td>
<td>Enables an auto-tunnel backup feature for the specified interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> auto-tunnel backup</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> attribute-set attribute-set-name</td>
<td>Configures attribute-set template for auto-tunnel backup tunnels.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if-auto-backup)#attribute-set ab</td>
<td></td>
</tr>
</tbody>
</table>
### 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> <code>configure</code></td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>mpls traffic-eng</code></td>
<td>Enables traffic engineering on a specific interface on the originating node.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# mpls traffic-eng
```

**Step 3** `interface type interface-path-id`

**Example:**

```
RP/0/RSP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0
```
### Command or Action | Purpose
---|---
**Step 4** | auto-tunnel backup nhop-only
**Example:**
RP/0/RSP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup nhop-only

EnablesthecreationofdynamicNHOP backup tunnels. Bydefault, bothNHOP andNNHOP protection are enabled.

**Note** Using this nhop-only option, only link protection is provided.

### Step 5 | auto-tunnel backup exclude srlg [preferred]
**Example:**
RP/0/RSP0/CPU0:router(config-mpls-te-if)# auto-tunnel backup exclude srlg preferred

EnablestheexclusionofSRLGvalues onagivenlinkfortheAutoTunnel backupassociatedwithagiveninterface.

The preferred option allows the AutoTunnel Backup tunnels to come up even if no path excluding all SRLG is found.

### Step 6 | attribute-set attribute-set-name
**Example:**
RP/0/RSP0/CPU0:router(config-mpls-te-if-auto-backup)#attribute-set ab

Configures attribute-set template for auto-tunnel backup tunnels.

### Step 7 | commit

### Step 8 | show mpls traffic-eng tunnels number detail
**Example:**
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels 1 detail

Displays information about configured NHOP tunnels and SRLG information.

---

**Related Topics**
- Backup AutoTunnels, on page 109
- Configure the MPLS-TE Auto-Tunnel Backup: Example, on page 242

### 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>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 prestandard DS-TE mode. The range for the total reserve bandwidth argument is 0 to 4294967295. Physical interface bandwidth is not used by MPLS-TE.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-rsvp-if)# bandwidth 100 150 sub-pool 50</td>
<td></td>
</tr>
<tr>
<td>Step 3 bandwidth [total reservable bandwidth] [bc0 bandwidth] [global-pool bandwidth] [sub-pool reservable-bw]</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-rsvp-if)# exit RP/0/RSP0/CPU0:router(config-rsvp)</td>
<td></td>
</tr>
<tr>
<td>Step 4 exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-rsvp)# exit RP/0/RSP0/CPU0:router(config)</td>
<td></td>
</tr>
</tbody>
</table>
### 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.

### Command or Action

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></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:router(config)# interface tunnel-te 2</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>signalled-bandwidth {bandwidth [class-type ct]</td>
<td>sub-pool bandwidth}</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# signalled-bandwidth sub-pool 10</td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics

- Configuring Traffic Engineering Tunnel Bandwidth, on page 70
- Prestandard DS-TE Mode, on page 113
- Configure IETF DS-TE Tunnels: Example, on page 233
### 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 ct] | sub-pool bandwidth}`
11. **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><strong>configure</strong></td>
<td>Enters RSVP configuration mode and selects an RSVP interface.</td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>rsvp interface</strong> <code>type interface-path-id</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# <code>rsvp interface pos0/6/0/0</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>bandwidth rdm</strong> `{total-reservable-bw</td>
<td>bc0</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if)# <code>bandwidth rdm 100 150</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>Physical interface bandwidth is not used by MPLS-TE.</td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>exit</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if)# <strong>exit</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp)</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><strong>exit</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp) <strong>exit</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/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>mpls traffic-eng</strong></td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>ds-te mode ietf</strong></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>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# ds-te mode ietf</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>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>interface tunnel-te tunnel-id</strong></td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te 4 RP/0/RSP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>**signalled-bandwidth {bandwidth [class-type ct]</td>
<td>sub-pool bandwidth}**</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# signalled-bandwidth 10 class-type 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>commit</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**
- Configuring Traffic Engineering Tunnel Bandwidth, on page 70
- Russian Doll Bandwidth Constraint Model, on page 114

**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 reservable bandwidth] [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

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 RSVP configuration mode.</td>
</tr>
<tr>
<td>Step 2</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></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# rsvp interface pos0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 3</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></td>
<td>Physical interface bandwidth is not used by MPLS-TE.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-if)# bandwidth mam max-reservable-bw 400 bc0 300 bc1 200</td>
<td></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>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/RSP0/CPU0:router(config-rsvp-if)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp)# exit</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/RSP0/CPU0:router(config-rsvp)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# exit</td>
<td></td>
</tr>
<tr>
<td>6</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/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td>7</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></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# ds-te mode ietf</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ds-te bc-model mam</td>
<td>Enables the MAM bandwidth constraint model globally.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# ds-te bc-model mam</td>
<td></td>
</tr>
<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/RSP0/CPU0:router(config-mpls-te)# exit</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/RSP0/CPU0:router(config)# interface tunnel-te 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>signalled-bandwidth {bandwidth [class-type ct]</td>
<td>sub-pool bandwidth}</td>
</tr>
</tbody>
</table>
### Configuring MPLS -TE and Fast-Reroute on OSPF

Perform this task to configure MPLS-TE and Fast Reroute (FRR) on OSPF.

#### Before You Begin

**Note**

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>Step 1 configure</td>
<td></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>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/RSP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>path-selection ignore overload {head</td>
</tr>
<tr>
<td>Example:</td>
<td>Ignores the Intermediate System-to-Intermediate System (IS-IS) overload bit setting for MPLS-TE.</td>
</tr>
<tr>
<td></td>
<td>If set-overload-bit is set by IS-IS on the head router, the tunnels stay up.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
</tr>
<tr>
<td></td>
<td>path-selection ignore overload head</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

## Related Topics

- Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE, on page 117
- Configure the Ignore IS-IS Overload Bit Setting in MPLS-TE: Example, on page 234

## 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 168
2. Associating Affinity-Names with TE Links, on page 169
3. Associating Affinity Constraints for TE Tunnels, on page 170

### 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).

**Note**

An affinity color name cannot exceed 64 characters. An affinity value cannot exceed a single digit. For example, magenta.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. affinity-map affinity name {affinity value | bit-position 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>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enters 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. The value you assign to a color name must be a single digit.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td>Step 3 affinity-map affinity name</td>
<td>Enters 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. The value you assign to a color name must be a single digit.</td>
</tr>
<tr>
<td>bit-position value</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
<td>affinity-map red 1</td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Flexible Name-based Tunnel Constraints, on page 118
Configure Flexible Name-based Tunnel Constraints: Example, on page 234

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.
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>Example:</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td>Enables MPLS-TE on an interface and enters MPLS-TE interface configuration mode.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# interface tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>attribute-names attribute name</td>
</tr>
<tr>
<td>Example:</td>
<td>Assigns colors to TE links over the selected interface.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if)# attribute-names red</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

Related Topics

Flexible Name-based Tunnel Constraints, on page 118
Configure Flexible Name-based Tunnel Constraints: Example, on page 234
Assigning Color Names to Numeric Values, on page 168

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

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>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> interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> affinity {affinity-value mask mask-value</td>
<td>exclude name</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# affinity include red</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Flexible Name-based Tunnel Constraints, on page 118
- Configure Flexible Name-based Tunnel Constraints: Example, on page 234
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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters an IS-IS instance.</td>
</tr>
<tr>
<td>Step 2</td>
<td>router isis instance-id</td>
<td>Enters an IS-IS network entity title (NET) for the routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# router isis 1</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>net network-entity-title</td>
<td>Enters address family configuration mode for configuring IS-IS routing that uses IPv4 and IPv6 address prefixes.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-isis)# net 47.0001.0000.0000.0002.00</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>address-family {ipv4</td>
<td>ipv6} {unicast}</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-isis)# address-family ipv4 unicast</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>metric-style wide</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-isis-af)# metric-style wide</td>
<td></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><code>configure</code></td>
</tr>
</tbody>
</table>
| **Step 2** | `router ospf process-name`  
Example:  
`RP/0/RSP0/CPU0:router(config)# router ospf 100`  
*process-name*  
Any alphanumeric string no longer than 40 characters without spaces. |
| **Step 3** | `mpls traffic-eng router-id ip-address`  
Example:  
`RP/0/RSP0/CPU0:router(config-ospf)# mpls traffic-eng router-id 192.168.70.1`  
Enters the MPLS interface type. For more information, use the question mark (?) online help function. |
## 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><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>explicit-path name name</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# explicit-path name interarea1</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>index index-id next-address [loose] ipv4 unicast ip-address</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-expl-path)# index 1 next-address loose ipv4 unicast 10.10.10.10</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>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>configure</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>interface tunnel-te tunnel-id</code></td>
<td>Enters MPLS-TE interface configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface tunnel-te 1
```

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>forwarding-adjacency holdtime value</code></td>
<td>Configures forwarding adjacency using an optional specific holdtime value. By default, this value is 0 (milliseconds).</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# forwarding-adjacency holdtime 60
```

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>commit</code></td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- MPLS-TE Forwarding Adjacency Benefits, on page 122
- Configure Forwarding Adjacency: Example, on page 237
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 176
- Configuring a Path Computation Element Address, on page 177
- Configuring PCE Parameters, on page 178

Configuring a Path Computation Client

Perform this task to configure a TE tunnel as a PCC.

Note

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>Step 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>Step 2</td>
<td>interface tunnel-te tunnel-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te 6</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>path-option preference-priority dynamic pce</td>
<td>Configures a TE tunnel as a PCC.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# path-option 1 dynamic pce</td>
<td></td>
</tr>
</tbody>
</table>
Related Topics
Path Computation Element, on page 122
Configure PCE: Example, on page 237

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>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td>Enters the MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3 pce address ipv4 address</td>
<td>Configures a PCE IPv4 address.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# pce address ipv4 10.1.1.1</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
Path Computation Element, on page 122
Configure PCE: Example, on page 237
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>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td>Purpose</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>pce address ipv4 address</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# pce address ipv4 10.1.1.1</td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures a PCE IPv4 address.</td>
</tr>
<tr>
<td>Step 4</td>
<td>pce peer ipv4 address</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# pce peer address ipv4 10.1.1.1</td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures a static PCE peer address. PCE peers are also discovered dynamically through OSPF or ISIS.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>5</td>
<td><code>pce keepalive interval</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>6</td>
<td><code>pce deadtimer value</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>7</td>
<td><code>pce reoptimize value</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>8</td>
<td><code>pce request-timeout value</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>9</td>
<td><code>pce tolerance keepalive value</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>10</td>
<td><code>commit</code></td>
</tr>
<tr>
<td>11</td>
<td>`show mpls traffic-eng pce peer [address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>12</td>
<td><code>show mpls traffic-eng pce tunnels</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
</tbody>
</table>
Related Topics
Path Computation Element, on page 122
Configure PCE: Example, on page 237

Configuring Forwarding Path

Perform this task to configure forwarding path in the MPLS-TE interface.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. forward-class forward-class
4. exit
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>interface tunnel-te tunnel-id</td>
<td>Defines forwarding path in the MPLS-TE interface.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config)# interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>forward-class forward-class</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-if)# forward-class 1</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-if)# exit RP/0/RSP0/CPU0:router(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Path Protection on MPLS-TE

These tasks show how to configure path protection on MPLS-TE:

Enabling Path Protection for an Interface

Perform this task to enable path protection for a given tunnel interface.

SUMMARY STEPS

1. configure
2. interface tunnel-te tunnel-id
3. path-protection
4. commit
5. show mpls traffic-eng tunnels [tunnel-number]

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 and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-te tunnel-id</td>
<td>Enables path protection on the tunnel-te interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te 6</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>path-protection</td>
<td>Displays information that path protection is enabled on the tunnel-te interface for tunnel number 6.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# path-protection</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>show mpls traffic-eng tunnels [tunnel-number]</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels 6</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Path Protection, on page 125
Assigning a Dynamic Path Option to a Tunnel

Perform this task to assign a secondary path option in case there is a link or node failure along a path and all interfaces in your network are not protected.

SUMMARY STEPS

1. configure
2. interface tunnel-te <tunnel-id>
3. path-option preference-priority dynamic
4. commit
5. 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>Step 1 configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 interface tunnel-te &lt;tunnel-id&gt;</td>
<td>Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# interface tunnel-te 6</td>
<td></td>
</tr>
<tr>
<td>Step 3 path-option preference-priority dynamic</td>
<td>Configures a secondary path option for an MPLS-TE tunnel.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-if)# path-option 10 dynamic</td>
<td></td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
<tr>
<td>Step 5 show mpls traffic-eng tunnels [tunnel-number]</td>
<td>Displays information about the secondary path option that on the tunnel-te interface for tunnel number 6.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels 6</td>
<td></td>
</tr>
</tbody>
</table>
Related Topics

Path Protection, on page 125
Pre-requisites for Path Protection, on page 125
Restrictions for Path Protection, on page 126
Restrictions for Explicit Path Protection, on page 126
Configure Tunnels for Path Protection: Example, on page 238

Forcing a Manual Switchover on a Path-Protected Tunnel

Perform this task to force a manual switchover on a path-protected tunnel.

SUMMARY STEPS

1. `mpls traffic-eng path-protection switchover tunnel-te tunnel-ID`
2. `commit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Forces the path protection switchover of the Point-to-Point (P2P) tunnel on the tunnel-te interface.</td>
</tr>
<tr>
<td><code>mpls traffic-eng path-protection switchover tunnel-te tunnel-ID</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router# mpls traffic-eng path-protection switchover tunnel-te 6</code></td>
<td></td>
</tr>
</tbody>
</table>

| Step 2 | commit |

Related Topics

Path Protection, on page 125
Pre-requisites for Path Protection, on page 125
Restrictions for Path Protection, on page 126
Restrictions for Explicit Path Protection, on page 126
Configure Tunnels for Path Protection: Example, on page 238

Configuring the Delay the Tunnel Takes Before Reoptimization

Perform this task to configure the time between when a path-protection switchover event is effected on a tunnel head to when a reoptimization is performed on that tunnel. This timer affects only the required reoptimization that is attempted due to a switchover and does not override the global reoptimization timer.
### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. reoptimize timers delay path-protection 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 MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Adjusts the number of seconds that the tunnel takes before triggering reoptimization after switchover has happened. <strong>Note</strong>: The restriction is that at least one dynamic path-option must be configured for a standby LSP to come up. The strict (explicit) path option is not supported for the standby LSP.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> reoptimize timers delay path-protection seconds</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-mpls-te)# reoptimize timers delay path-protection 180</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Path Protection, on page 125
- Pre-requisites for Path Protection, on page 125
- Restrictions for Path Protection, on page 126
- Restrictions for Explicit Path Protection, on page 126
- Configure Tunnels for Path Protection: Example, on page 238

**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>1</td>
<td><code>configure</code></td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td><code>mpls traffic-eng</code></td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-mpls-te)#</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>auto-bw collect frequency minutes</code></td>
<td>Configures the automatic bandwidth collection frequency, and controls</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the manner in which the bandwidth for a tunnel collects output rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information; but does not adjust the tunnel bandwidth.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-mpls-te)# auto-bw collect frequency 1</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>commit</code></td>
<td>Displays information about MPLS-TE tunnels for the automatic bandwidth.</td>
</tr>
<tr>
<td>5</td>
<td><code>show mpls traffic-eng tunnels [auto-bw]</code></td>
<td>Displays information about MPLS-TE tunnels for the automatic bandwidth.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# show mpls traffic tunnels auto-bw</code></td>
<td></td>
</tr>
</tbody>
</table>

### Related Topics
- [MPLS-TE Automatic Bandwidth Overview](#), on page 127
- [Configure Automatic Bandwidth: Example](#), on page 239
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>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`mpls traffic-eng auto-bw apply {all</td>
<td>tunnel-te tunnel-number}`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/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></td>
<td></td>
<td>Configures the highest bandwidth instantly to the specified tunnel. Range is from 0 to 65535.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>commit</code></td>
<td>Displays information about MPLS-TE tunnels for the automatic bandwidth.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>show mpls traffic-eng tunnels [auto-bw]</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels auto-bw</code></td>
<td></td>
</tr>
</tbody>
</table>

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>Step 1  configure</td>
<td></td>
</tr>
<tr>
<td>Step 2  interface tunnel-te  tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface and enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-te 6 RP/0/RSP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td>Step 3  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></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# auto-bw RP/0/RSP0/CPU0:router(config-if-tunte-autobw)#</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>application minutes</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/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 {min bandwidth} {max bandwidth}</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/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 percentage {min minimum-bandwidth}</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/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 percentage [min bandwidth] limit limit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/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>
</tbody>
</table>
## Implementing MPLS Traffic Engineering

### 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.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<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/RSP0/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 127
- [Configure Automatic Bandwidth: Example](#), on page 239

---

**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>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>srlg</td>
<td>Configures SRLG configuration commands on a specific interface configuration mode and assigns this SRLG a value.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# srlg</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
<td>Configures an interface type and path ID to be associated with an SRLG and enters SRLG interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-srlg)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>value value</td>
<td>Configures SRLG network values for a specific interface. Range is 0 to 4294967295.</td>
</tr>
<tr>
<td>Note</td>
<td>You can also set SRLG values on multiple interfaces including bundle interface.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-srlg-if)# value 100 RP/0/RSP0/CPU0:router (config-srlg-if)# value 200 RP/0/RSP0/CPU0:router(config-srlg-if)# value 300</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>show srlg interface type interface-path-id</td>
<td>(Optional) Displays the SRLG values configured for a specific interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show srlg interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>show srlg</td>
<td>(Optional) Displays the SRLG values for all the configured interfaces.</td>
</tr>
<tr>
<td>Note</td>
<td>You can configure up to 250 interfaces.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# show srlg</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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 2** explicit-path {identifier number [disable | index]} { name explicit-path-name}  
**Example:**  
RP/0/RSP0/CPU0:router(config)# explicit-path name backup-srlg | Enters the explicit path configuration mode. Identifier range is 1 to 65535. |
| **Step 3** index 1 exclude-address 192.168.92.1  
**Example:**  
RP/0/RSP0/CPU0:router(config-expl-path)# index 1 exclude-address 192.168.92.1 | Specifies the IP address to be excluded from the explicit path. |
| **Step 4** index 2 exclude-srlg 192.168.92.2  
**Example:**  
RP/0/RSP0/CPU0:router(config-expl-path)# index 2 exclude-srlg 192.168.192.2 | Specifies the IP address to extract SRLGs to be excluded from the explicit path. |
| **Step 5** commit | |
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-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 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>mpls traffic-eng</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>3</td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>4</td>
<td>backup-path tunnel-te tunnel-number</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>5</td>
<td>exit</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>6</td>
<td>exit</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>7</td>
<td>interface tunnel-te tunnel-id</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>8</td>
<td>ipv4 unnumbered type interface-path-id</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>9</td>
<td>path-option preference-priority { dynamic</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>10</td>
<td>destination ip-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
</tbody>
</table>

- Destination address is the remote node’s MPLS-TE router ID.
- Destination address is the merge point between backup and protected tunnels.
### Configuring the Shared Risk Link Groups

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong> exits</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Step 12</strong> commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> show run explicit-path name name</td>
<td>Displays the SRLG values that are configured for the link.</td>
</tr>
<tr>
<td><strong>Step 14</strong> show mpls traffic-eng topology path destination name explicit-path name</td>
<td>Displays the SRLG values that are configured for the link.</td>
</tr>
</tbody>
</table>

**Related Topics**
- MPLS Traffic Engineering Shared Risk Link Groups, on page 135
- Explicit Path, on page 135
- Fast ReRoute with SRLG Constraints, on page 136
- Importance of Protection, on page 138
- Delivery of Packets During a Failure, on page 139
- Multiple Backup Tunnels Protecting the Same Interface, on page 139
- Weighted-SRLG Auto-backup Path Computation, on page 139
- SRLG Limitations, on page 140
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

### 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>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></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
<td>Enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>backup-path tunnel-te tunnel-number</td>
<td>Sets the backup path to the primary tunnel outgoing interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>Step 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/RSP0/CPU0:router(config-mpls-te-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 6</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/RSP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring the Shared Risk Link Groups

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>interface tunnel-te</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config)# interface tunnel-te 2</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>ipv4 unnumbered</strong>&lt;br&gt;<strong>type</strong>&lt;br&gt;<strong>interface-path-id</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>path-option</strong>&lt;br&gt;<strong>preference-priority</strong>&lt;br&gt;**{ dynamic</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>destination</strong>&lt;br&gt;<strong>ip-address</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# destination 192.168.92.125</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>exit</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>explicit-path</strong>&lt;br&gt;**{identifier number [disable</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>index 1 exclude-srlg 192.168.92.2</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router:router(config-if)# index 1 exclude-srlg 192.168.192.2</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>commit</strong></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>show mpls traffic-eng tunnels (tunnel-number) detail</td>
<td>Display the tunnel details with SRLG values that are configured for the link.</td>
</tr>
</tbody>
</table>

**Example:**
```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels 2 detail
```

**Related Topics**
- MPLS Traffic Engineering Shared Risk Link Groups, on page 135
- Explicit Path, on page 135
- Fast ReRoute with SRLG Constraints, on page 136
- Importance of Protection, on page 138
- Delivery of Packets During a Failure, on page 139
- Multiple Backup Tunnels Protecting the Same Interface, on page 139
- Weighted-SRLG Auto-backup Path Computation, on page 139
- SRLG Limitations, on page 140
- Configure the MPLS-TE Shared Risk Link Groups: Example, on page 239

**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>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 traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>interface type interface-path-id</td>
<td>Enables traffic engineering on a particular interface on the originating node.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>backup-path tunnel-te tunnel-number</td>
<td>Sets the backup path for the primary tunnel outgoing interface.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-mpls-te)# backup-path tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-mpls-te-if)# exit</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-mpls-te)# exit</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>interface tunnel-te tunnel-id</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config)# interface tunnel-te 2</td>
<td></td>
</tr>
<tr>
<td>8</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: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>path-option preference-priority{ dynamic</td>
<td>explicit</td>
</tr>
<tr>
<td></td>
<td>{identifier</td>
<td>name explicit-path-name}</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# path-option 1 explicit name backup-srlg</td>
<td>Note You can use the dynamic option to dynamically assign path.</td>
</tr>
<tr>
<td>10</td>
<td>destination ip-address</td>
<td>Assigns a destination address on the new tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# destination 192.168.92.125</td>
<td>• Destination address is the remote node’s MPLS-TE router ID.</td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Destination address is the merge point between backup and protected tunnels.</td>
<td></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>
</tbody>
</table>

### Step 11

**exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# exit
```

Exits the current configuration mode.

### Step 12

**explicit-path {identifier number [disable | index]}{ name explicit-path-name}**

**Example:**

```
RP/0/RSP0/CPU0:router(config)# explicit-path name backup-srlg-nodep
```

Enters the explicit path configuration mode. Identifier range is 1 to 65535.

### Step 13

**index 1 exclude-address 192.168.92.1**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# index 1 exclude-address 192.168.92.1
```

Specifies the protected node IP address to be excluded from the explicit path.

### Step 14

**index 2 exclude-srlg 192.168.92.2**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# index 2 exclude-srlg 192.168.192.2
```

Specifies the protected link IP address to get SRLGs to be excluded from the explicit path.

### Step 16

**show mpls traffic-eng tunnels topology path destination ip-address explicit-path-name name**

**Example:**

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels topology path destination 192.168.92.125 explicit-path-name backup-srlg-nodep
```

Displays the path to the destination with the constraint specified in the explicit path.

### Related Topics

- MPLS Traffic Engineering Shared Risk Link Groups, on page 135
- Explicit Path, on page 135
- Fast ReRoute with SRLG Constraints, on page 136
- Importance of Protection, on page 138
- Delivery of Packets During a Failure, on page 139
- Multiple Backup Tunnels Protecting the Same Interface, on page 139
- Weighted-SRLG Auto-backup Path Computation, on page 139
Configuring Default Admin Weight

Perform this task to configure a default admin weight to apply to all SRLG values if a specific admin weight is not configured under the SRLG value configuration mode.

SUMMARY STEPS

1. `configure`
2. `mpls traffic-eng srlg`
3. `admin-weight weight`
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><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>mpls traffic-eng srlg</code></td>
<td>Enters MPLS TE SRLG configuration mode.</td>
</tr>
<tr>
<td></td>
<td>`Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng srlg</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>admin-weight weight</code></td>
<td>Configures default admin-weight for all the SRLG values. Range is from 0-4294967295. Default is 1. The example shows how to configure an admin-weight of 10 for all the SRLG values.</td>
</tr>
<tr>
<td></td>
<td>`Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-srlg)# admin-weight 10</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>commit</code></td>
<td></td>
</tr>
</tbody>
</table>

Configuring Static SRLG Value to Topology Link

Perform this task to assign static SRLG value to a topology link based on its IP address. Use this command for platforms that do not support SRLG flooding, so that the local node auto-tunnel backup diverse path calculation is based on static SRLG.
SUMMARY STEPS

1. configure
2. mpls traffic-eng srlg
3. value srlg-value
4. static ipv4 address ip-address next-hop ipv4 address next-hop-ip-address
5. 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>mpls traffic-eng srlg</td>
<td>Enters MPLS TE SRLG configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config)# mpls traffic-eng srlg</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>value srlg-value</td>
<td>Enters MPLS TE SRLG value configuration mode and configures SRLG value. The example shows how to enter MPLS TE SRLG value configuration mode and configure a SRLG value of 5.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-mpls-te-srlg)# value 5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>static ipv4 address ip-address next-hop ipv4 address next-hop-ip-address</td>
<td>Configures static SRLG value to a topology link. The example shows how to configure static SRLG value for a topology link with source IP address of 1.1.1.1 and next-hop IP address of 1.1.1.2.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-mpls-te-srlg)# static ipv4 address 1.1.1.1 next-hop ipv4 address 1.1.1.2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Admin-Weight Associated with an SRLG Value

Perform this task to configure admin-weight associated with an SRLG value. This admin-weight will be added to the link admin weight during SRLG aware path calculation when the link matches the SRLG value of the protected link. The admin-weight configured in the MPLS TE SRLG value configuration mode overwrites the admin-weight configured in the MPLS TE SRLG configuration mode.
**SUMMARY STEPS**

1. configure
2. mpls traffic-eng srlg
3. value srlg-value
4. admin-weight weight
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 SRLG configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng srlg</td>
<td>Enters MPLS TE SRLG value configuration mode and configures SRLG value. The example shows how to enter MPLS TE SRLG value configuration mode and configure a SRLG value of 150.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config)# mpls traffic-eng srlg</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>value srlg-value</td>
<td>Configures admin-weight for SRLG value. Range is from 0-4294967295. Default is 1. The example shows how to configure an admin-weight of 100 for the SRLG value of 150.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-mpls-te-srlg)# value 150</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>admin-weight weight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-mpls-te-srlg)# admin-weight 100</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters multicast routing configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> 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><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# multicast-routing RP/0/RSP0/CPU0:router(config-mcast)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> address-family {ipv4</td>
<td>ipv6 }</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-mcast)# address-family ipv6 RP/0/RSP0/CPU0:router(config-mcast-default-ipv6)#</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>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-mcast-default-ipv6-if)# interface tunnel-mte 1 RP/0/RSP0/CPU0:router(config-mcast-default-ipv6-if)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> enable</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

#### Purpose

**Step 6**
exit

Example:
RP/0/RSP0/CPU0:router(config-mcast-default-ipv6-if)#
exit
RP/0/RSP0/CPU0:router(config-mcast-default-ipv6)#

Exits the current configuration mode.

**Step 7**
interface type interface-path-id

Example:
RP/0/RSP0/CPU0:router(config-mcast-default-ipv6)#
interface GigabitEthernet0/2/0/3
RP/0/RSP0/CPU0:router(config-mcast-default-ipv6-if)#

Configures multicast routing on the GigabitEthernet interface.

**Step 8**
enable

Example:
RP/0/RSP0/CPU0:router(config-mcast-default-ipv6-if)#
enable

Enables multicast routing on the GigabitEthernet interface.

**Step 9**
commit

**Step 10**
show pim ipv6 interface type interface-path-id

Example:
RP/0/RSP0/CPU0:router# show pim ipv6 interface
tunnel-mte 1

Displays the output for the P2MP-TE tunnel interface that has IPv6 multicast enabled.

---

**Related Topics**

Configuring the Static Group for the Point-to-Multipoint Interface, on page 204

### 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> configure</td>
<td>Enters router MLD configuration mode.</td>
</tr>
</tbody>
</table>

**Step 2** router mld  
Example:  
RP/0/RSP0/CPU0:router(config)# router mld  
RP/0/RSP0/CPU0:router(config-mld)#

**Step 3** vrf vrf-name  
Example:  
RP/0/RSP0/CPU0:router(config-mld)#vrf default  
RP/0/RSP0/CPU0:router(config-mld-default)#

**Step 4** interface tunnel-mte tunnel-id  
Example:  
RP/0/RSP0/CPU0:router(config-mld-default)#interface tunnel-mte 1  
RP/0/RSP0/CPU0:router(config-mld-default-if)#

**Step 5** static-group group-address  
Example:  
RP/0/RSP0/CPU0:router(config-mld-default-if)#static-group ff35::1 2000::1

**Step 6** commit
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> show mrib ipv6 route source-address</td>
<td>Verifies the multicast static mapping.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CP0:router# show mrib ipv6 route ff35::1</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

Enabling Multicast Routing on the Router, on page 202

**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>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-mte tunnel-id</td>
<td>Configures an MPLS-TE P2MP tunnel interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>destination ip-address</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>4</td>
<td>path-option preference-priority explicit identifier path-number</td>
<td>Configures the path number of the IP explicit path.</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>5</td>
<td><strong>path-option preference-priority dynamic</strong></td>
<td>Specifies that label switched paths (LSP) are dynamically calculated.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# path-option 2 dynamic</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>exit</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# <strong>exit</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>destination ip-address</strong></td>
<td>Sets the destination address for tunnel-mte 10 to 172.16.255.2.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>destination 172.16.255.2</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)#</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><strong>path-option preference-priority explicit name pathname</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></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# <strong>path-option 1 explicit name</strong></td>
<td>how-to-get-to-172.16.255.2</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)#</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><strong>path-option preference-priority dynamic</strong></td>
<td>Sets the fall back path option as dynamic when the destination cannot come to the explicit path.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# <strong>path-option 2 dynamic</strong></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>exit</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# <strong>exit</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>destination ip-address</strong></td>
<td>Specifies that destination 172.16.255.3 uses only the dynamically computed path.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)#</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>destination 172.16.255.3</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Step 12</th>
<th>path-option preference-priority explicit name pathname [verbatim]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>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>
</tbody>
</table>

**Example:**

```bash
RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# path-option 1 explicit name how-to-get-to-172.16.255.3 verbatim
```

<table>
<thead>
<tr>
<th>Step 13</th>
<th>commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Displays the brief summary of the P2MP tunnel status and configuration.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 14</th>
<th>show mpls traffic-eng tunnels [brief] [p2mp tunnel-number]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Displays the brief summary of the P2MP tunnel status and configuration.</td>
</tr>
</tbody>
</table>

**Example:**

```bash
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels brief p2mp 10
```

### Related Topics

- Enabling Multicast Routing on the Router, on page 202
- Configuring the Static Group for the Point-to-Multipoint Interface, on page 204

### 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>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td><code>interface tunnel-mte tunnel-id</code></td>
</tr>
</tbody>
</table>
| **Example:** | ```
RP/0/RSP0/CPU0:router(config)# interface tunnel-mte 101
RP/0/RSP0/CPU0:router(config-if)#
``` | |
| **3** | `ipv4 unnumbered type interface-path-id` | Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is commonly used as the interface type. |
| **Example:** | ```
RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
``` | |
| **4** | `destination ip-address` | Sets the destination address for tunnel-mte 10 to 140.140.140.140. |
| **Example:** | ```
RP/0/RSP0/CPU0:router(config-if)# destination 140.140.140.140
RP/0/RSP0/CPU0:router(config-if-p2mp-dest)#
``` | |
| **5** | `disable` | Disables destination 140.140.140.140 for tunnel-mte 10. |
| **Example:** | ```
RP/0/RSP0/CPU0:router(config-if-p2mp-dest)#disable
``` | |
| **6** | `path-option preference-priority dynamic` | Specifies that label switched paths (LSP) are dynamically calculated. |
| **Example:** | ```
RP/0/RSP0/CPU0:router(config-if-p2mp-dest)#path-option 1 dynamic
``` | |
| **7** | `path-option preference-priority explicit name pathname` | Specifies that destination 140.140.140.140 uses the explicit path identified by the explicit path name "to4." |
| **Example:** | ```
RP/0/RSP0/CPU0:router(config-if-p2mp-dest)#path-option 2 explicit name to4
``` | |
| **8** | `commit` | |
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></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-mte 1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)#</td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>destination ip-address</td>
<td>Sends out the log message when the tunnel LSP goes up or down when the software is enabled.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# destination 100.0.0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)#</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Step 5 | logging events lsp-status state | | }

Implementing MPLS Traffic Engineering
Configuring Point-to-Multipoint TE
<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>logging events lsp-status reroute</td>
<td>Sends out the log message when the tunnel LSP is rerouted due to an FRR event when the software is enabled.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# logging events lsp-status reroute</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>path-option preference-priority explicit name pathname</td>
<td>Specifies the path name of the IP explicit path. Destination 100.0.0.3 uses the explicit path that is identified by the explicit path name &quot;path123.&quot;</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if-p2mp-dest)# path-option 1 explicit name path123</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>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/RSP0/CPU0:router(config-if-p2mp-dest)# exit</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>fast-reroute</td>
<td>Enables fast-reroute (FRR) protection for a P2MP TE tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-if)# fast-reroute</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>commit</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 traffic-eng tunnels [p2mp]</td>
<td>Displays the information for all P2MP tunnels.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels p2mp</td>
<td></td>
</tr>
</tbody>
</table>

### 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>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 soft-preemption on a node.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 3 soft-preemption</td>
<td>Enables soft-preemption on a node.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# soft-preemption</td>
<td>Enables soft-preemption on a node.</td>
</tr>
<tr>
<td>Note</td>
<td>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 timeout seconds</td>
<td>Specifies the timeout for the soft-preempted LSP, in seconds. The range is from 1 to 300.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-soft-preemption)# timeout 20</td>
<td>Specifies the timeout for the soft-preempted LSP, in seconds. The range is from 1 to 300.</td>
</tr>
<tr>
<td>Step 5 commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

Soft-Preemption, on page 140

**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 tunnel-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# interface tunnel-te 10</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>soft-preemption</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# soft-preemption</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

Enables soft-preemption on a tunnel.

When soft preemption is enabled on a tunnel, these actions occur:

- A path-modify message is sent for the current LSP with the `soft preemption desired` property.
- A path-modify message is sent for the reopt LSP with the `soft preemption desired` property.
- A path-modify message is sent for the path protection LSP with the `soft preemption desired` property.
- A path-modify message is sent for the current LSP in FRR active state with the `soft preemption desired` property.

**Note**
The soft-preemption is not available in the interface tunnel-mte and interface tunnel-gte configuration modes.

Related Topics

Soft-Preemption, on page 140

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>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 attribute-set path option configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>attribute-set path-option attribute-set-name</td>
<td>Enters attribute-set path option configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-mpls-te)# attribute-set path-option myset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>The configuration at the path-option level takes precedence over the values configured at the level of the tunnel, and therefore is applied.</td>
</tr>
<tr>
<td>Step 4</td>
<td>affinity affinity-value mask mask-value</td>
<td>Configures affinity attribute under a path option attribute-set. The attribute values that are required for links to carry this tunnel.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-te-attribute-set)# affinity 0xBEEF mask 0xBEEF</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>signalled-bandwidth kbps class-type class-type number</td>
<td>Configures the bandwidth attribute required for an MPLS-TE tunnel under a path option attribute-set.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-te-attribute-set)# signalled-bandwidth 1000 class-type 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>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.</td>
</tr>
<tr>
<td>Step 6</td>
<td>commit</td>
<td>Displays the attributes that are defined in the attribute-set for the link.</td>
</tr>
<tr>
<td>Step 7</td>
<td>show mpls traffic-eng attribute-set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router# show mpls traffic-eng attribute-set</td>
<td></td>
</tr>
</tbody>
</table>
### 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><code>mpls traffic-eng</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>auto-tunnel mesh</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config-mpls-te)# auto-tunnel mesh</code></td>
</tr>
</tbody>
</table>
### Purpose

Command or Action | Purpose
--- | ---
**Step 4** tunnel-id min value max value | Specifies the minimum and maximum number of auto-tunnel mesh tunnels that can be created on this router. The range of tunnel ID is from 0 to 65535.

Example:

```
RP/0/RSP0/CPU0:router(config-te-auto-mesh)#
tunnel-id min 10 max 50
```

**Step 5** commit

---

### Related Topics

- Auto-Tunnel Mesh, on page 143
- Destination List (Prefix-List), on page 144

### 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> configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> mpls traffic-eng</td>
<td>Enables auto-tunnel mesh groups globally.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RSP0/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> auto-tunnel mesh</td>
<td></td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RSP0/CPU0:router(config-mpls-te)#
auto-tunnel mesh
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 4   timer removal unused timeout | 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  
  Note The unused timer runs per tunnel because the same destination in different mesh-groups may have different tunnels created. |
| Step 5   commit                     |                                                                                                                                        |

**Related Topics**

- Auto-Tunnel Mesh, on page 143
- Destination List (Prefix-List), on page 144

**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

**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>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables auto-tunnel mesh groups globally.</td>
</tr>
<tr>
<td>auto-tunnel mesh</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# auto-tunnel mesh</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the membership of auto-tunnel mesh. The range is from 0 to 4294967295.</td>
</tr>
<tr>
<td>group value</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-auto-mesh)# group 65</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>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></td>
<td>Disables the meshgroup and deletes all tunnels created for this meshgroup.</td>
</tr>
<tr>
<td>disable</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-auto-mesh-group)# disable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></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>attribute-setname</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-auto-mesh-group)# attribute-set am-65</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></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>destination-list</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-auto-mesh-group)# destination-list dl-65</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**
- Auto-Tunnel Mesh, on page 143
- Destination List (Prefix-List), on page 144

**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>
<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/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> attribute-set auto-mesh attribute-set-name</td>
<td>Specifies name of the attribute-set of auto-mesh type.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-te)# attribute-set</td>
<td></td>
</tr>
<tr>
<td>auto-mesh attribute-set-mesh</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> affinity value mask mask-value</td>
<td>Configures the affinity properties the tunnel requires in its links for an MPLS-TE tunnel under an auto-mesh attribute-set.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-te)# affinity 0101</td>
<td></td>
</tr>
<tr>
<td>mask 320</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> signalled-bandwidth kbps class-type class-type number</td>
<td>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).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-te-attribute-set)#</td>
<td></td>
</tr>
<tr>
<td>signalled-bandwidth 1000 class-type 0</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>You can configure the class type of the tunnel bandwidth request. The class-type 0 is strictly equivalent to <strong>global-pool</strong> and class-type 1 is strictly equivalent to <strong>subpool</strong>.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>autoroute announce</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>fast-reroute protect bandwidth node</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>auto-bw collect-bw-only</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>logging events lsp-status {state</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>commit</td>
</tr>
</tbody>
</table>
**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><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ldp)# mpls ldp</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>traffic-eng auto-tunnel mesh</td>
<td>Enters auto-tunnel mesh configuration mode. You can configure TE auto-tunnel mesh groups from this mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ldp-te-auto-mesh)# traffic-eng auto-tunnel mesh</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>group id all</td>
<td>Configures an auto-tunnel mesh group of interfaces in LDP. You can enable LDP on all TE meshgroup 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></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-ldp-te-auto-mesh)# group all</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Auto-Tunnel Mesh, on page 143
- Destination List (Prefix-List), on page 144

**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 223, **Associated Bidirectional Non Co-routed and Co-routed LSPs**, on page 224 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 227 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 225 describes the procedure to create associated bidirectional co-routed LSPs.

## Associated Bidirectional Non-Corouted and Corouted LSPs

This section provides an overview of associated bidirectional non-corouted and corouted 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 corouted 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 225* 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.

### SUMMARY STEPS

1. configure
2. interface tunnel-te *tunnel-id*
3. bidirectional
4. association {id <0-65535> | source-address <IP address>} [global-id <0-4294967295>]
5. association type co-routed
6. commit
7. show mpls traffic-eng tunnels bidirectional-associated co-routed

### 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.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-te <em>tunnel-id</em></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>Example:</td>
<td>RP/0/RSP0/CPU0:router# interface tunnel-te 1</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>bidirectional</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>Example:</td>
<td>RP/0/0/CPU0:router(config-if)# bidirectional</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>association {id &lt;0-65535&gt;</td>
<td>source-address &lt;IP address&gt;} [global-id &lt;0-4294967295&gt;]</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/0/CPU0:router(config-if-bidir)# association id 1 source-address 11.0.0.1</td>
<td>Association ID, association source and global ID must be configured identically on both the endpoints.</td>
</tr>
<tr>
<td>Step 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>Example:</td>
<td>RP/0/0/CPU0:router(config-if-bidir)#association type co-routed</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>Step 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>Example:</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-te1 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 Parameters: Min-interval 100 ms (default), Multiplier 3 (default)
  BFD Bringsup Timeout: Interval 60 seconds (default)
  BFD Initial Dampening: 160000 ms (default)
  BFD Maximum Dampening: 6000000 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 228 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>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>
</tbody>
</table>

**Step 2** `interface tunnel-te tunnel-id`

*Example*:
```
RP/0/RSP0/CPU0:router# interface tunnel-te 1
```

**Step 3** `ipv4 unnumbered type interface-path-id`

*Example*:
```
RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
```

**Step 4** `bfd {fast-detect | encap-mode}`

*Example*:
```
RP/0/RSP0/CPU0:IMC0(config-if-bfd)#fast-detect
RP/0/RSP0/CPU0:IMC0(config-if-tunte-bfd)#encap-mode gal
```

**Step 5** `destination ip-address`

*Example*:
```
RP/0/RSP0/CPU0:router(config-if)# destination 49.49.49.2
```

**Step 6** `bidirectional`

*Example*:
```
Router(config-if)# bidirectional
```
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>bidirectional association {id &lt;0-65535&gt;</td>
<td>source-address &lt;IP address&gt;} {global-id &lt;0-4294967295&gt;</td>
</tr>
<tr>
<td></td>
<td>Example: | 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 a associated co-routed bidirectional LSP.</td>
</tr>
<tr>
<td></td>
<td>Example: | Router(config-if-bidir)#association type co-routed</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>path-protection</td>
<td>Enable path protection.</td>
</tr>
<tr>
<td></td>
<td>Example: | RP/0/RSP0/CPU0:IMC0(config-if-bidir-co-routed)#path-protection</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>path-option preference - priority {dynamic</td>
<td>explicit}</td>
</tr>
<tr>
<td></td>
<td>Example: | RP/0/RSP0/CPU0:router(config-if)# path-option 1 dynamic</td>
<td>Note Both sides of the co-routed bidirectional LSPs must use dynamic or matching co-routed strict-hop explicit path-option.</td>
</tr>
<tr>
<td>11</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

Here is a sample configuration with path protection defined for the Associated Bidirectional LSP.

```
RP/0/RSP0/CPU0: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)#path-option 1 dynamic
RP/0/RSP0/CPU0:IMC0(config-if)#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 230 and *Generate Fault OAM Messages at End-point*, on page 231 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>configure</td>
<td></td>
</tr>
<tr>
<td>mpls traffic-eng</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td>fault-oam</td>
<td>Enable fault OAM for an associated bidirectional LSP.</td>
</tr>
<tr>
<td>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>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><code>interface tunnel-te tunnel-id</code></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:IMC0(config)#interface tunnel-te 1</td>
<td>Configures an MPLS-TE tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>bidirectional association type co-routed fault-oam</code></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:IMC0(config-if)#bidirectional association type co-routed fault-oam</td>
<td>Enable fault OAM for an associated co-routed bidirectional LSP.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>commit</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:

**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
  !
  interface POS0/0/0/0
  address-family ipv4 unicast
  !

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
  !
  rsvp
  interface pos 0/6/0/0
  bandwidth 100 150 sub-pool 50
  interface tunnel-te1
  bandwidth sub-pool 10
Configure IETF DS-TE Tunnels: Example

The following example shows how to configure DS-TE:

```
commit

rsvp
  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
  rsvp 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

rsvp
  interface pos 0/6/0/0
  bandwidth rdm percentage bc0 100 bc1 50
  bandwidth 10 class-type 1
  commit

configure
  rsvp interface 0/6/0/0
  bandwidth mam percentage bc0 100 bc1 50
  ds-te mode ietf
  ds-te model mam
  interface tunnel-te 1 bandwidth 10 class-type 1
  commit
```

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
```
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

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
```

Related Topics
- Configuring the Ignore Integrated IS-IS Overload Bit Setting in MPLS-TE, on page 167
- Ignore Intermediate System-to-Intermediate System Overload Bit Setting in MPLS-TE, on page 117

Configure Flexible Name-based Tunnel Constraints: Example

The following configuration shows the three-step process used to configure flexible name-based tunnel constraints.
line console
exe-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 47.0001.0000.0000.0001.00
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
!
mlps traffic-eng
interface GigabitEthernet0/1/0/0
attribute-names red purple
interface GigabitEthernet0/1/0/1
attribute-names red orange
Configure an Interarea Tunnel: Example

The following configuration example shows how to configure a traffic engineering interarea tunnel.

```
! 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
```

**Related Topics**

Assigning Color Names to Numeric Values, on page 168  
Associating Affinity-Names with TE Links, on page 169  
Associating Affinity Constraints for TE Tunnels, on page 170  
Flexible Name-based Tunnel Constraints, on page 118

---

**Configure an Interarea Tunnel: Example**

The following configuration example shows how to configure a traffic engineering interarea tunnel.

```
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 175
- MPLS-TE Forwarding Adjacency Benefits, on page 122

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
```
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.

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

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.

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.

```
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:

```
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 184
- Configuring the Automatic Bandwidth Functions, on page 186
- MPLS-TE Automatic Bandwidth Overview, on page 127

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:

```
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.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng topology brief
My_System_id: 100.0.0.2 (OSPF 0 area 0)
```
Implementing MPLS Traffic Engineering

The following examples show the configured tunnels and associated SRLG values.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels
<snip>
Signalling Summary:
LSF 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
```

The following example shows all the interfaces associated with SRLG.

```
RP/0/RSP0/CPU0:router# show mpls traffic-eng topo srlg
```

```
<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>

Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide, Release 4.3.x
The following example shows the NHOP and NNHOP backup tunnels with excluded SRLG values.

```
RP/0/RSP0/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
Hop1:50.5.1.1
Hop2:50.1.3.1
Hop3:50.1.3.3
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/RSP0/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
 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 189
- Creating an Explicit Path With Exclude SRLG, on page 191
- Using Explicit Path With Exclude SRLG, on page 192
- Creating a Link Protection on Backup Tunnel with SRLG Constraint, on page 194
- Creating a Node Protection on Backup Tunnel with SRLG Constraint, on page 197
- MPLS Traffic Engineering Shared Risk Link Groups, on page 135
- Explicit Path, on page 135
- Fast ReRoute with SRLG Constraints, on page 136
- Importance of Protection, on page 138
- Delivery of Packets During a Failure, on page 139
- Multiple Backup Tunnels Protecting the Same Interface, on page 139
- Weighted-SRLG Auto-backup Path Computation, on page 139
- SRLG Limitations, on page 140
Configure the MPLS-TE Auto-Tunnel Backup: Example

The following example shows the auto-tunnel backup configuration for core or edge routers.

```
RP/0/RSP0/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/RSP0/CPU0:router# show mpls traffic-eng tunnels 1
Signalling 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: 0x0/0xffff
  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 LSPs: 4
    Protected S2L Sharing Families: 0
    Protected S2Ls: 0
    Protected i/f: Gi0/1/0/0  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/RSP0/CP0:router# show mpls traffic-eng tunnels 1 detail
Signalling 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 ? 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: Gi0/1/0/0 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/RSP0/CP0: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>
</tbody>
</table>
```
This example shows the auto-tunnel backup details.

RP/0/RSP0/CP00# 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: 0x0/0xffffff
  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: G10/1/0/3  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:
Record Route:
IPv4 24.9.0.2, flags 0x0
IPv4 12.9.1.1, flags 0x0
Tspec: avg rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
Displayed 1 (of 104) heads, 0 (of 0) midpoints, 0 (of 201) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

This example shows the automatically created backup tunnels.

RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels auto-tunnel backup tabular

<table>
<thead>
<tr>
<th>Tunnel Name</th>
<th>LSP ID</th>
<th>Destination Address</th>
<th>Source Address</th>
<th>Tun</th>
<th>FRR</th>
<th>LSP Path</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>*tunnel-te400</td>
<td>2</td>
<td>1.1.1.1</td>
<td>4.4.4.4</td>
<td>up</td>
<td>Inact</td>
<td>Head</td>
<td>Inact</td>
</tr>
<tr>
<td>*tunnel-te401</td>
<td>2</td>
<td>3.3.3.3</td>
<td>4.4.4.4</td>
<td>up</td>
<td>Inact</td>
<td>Head</td>
<td>Inact</td>
</tr>
<tr>
<td>* - automatically created backup tunnel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RP/0/RSP0/CPU0:router# show mpls traffic-eng tunnels auto-tunnel backup brief

<table>
<thead>
<tr>
<th>TUNNEL NAME</th>
<th>DESTINATION ADDRESS</th>
<th>STATUS</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>*tunnel-te400</td>
<td>1.1.1.1</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>*tunnel-te401</td>
<td>3.3.3.3</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>* - automatically created backup tunnel</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Displayed 2 (of 104) heads, 0 (of 0) midpoints, 0 (of 201) tails
Displayed 2 up, 0 down, 0 recovering, 0 recovered heads

This example shows the attribute-set for auto-backup tunnels.

RP/0/RSP0/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)
POS0_3_0_1
List of tunnel IDs (count 1)
3000

This example shows the attribute-set for auto-mesh tunnels.

RP/0/RSP0/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-bw: 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/RSP0/CPU0# 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_nhopper_te3000) (Basis for Setup, path weight 2)
  path option 10, type explicit (autob_nhopper_srlg_te3000) [disabled]
  G-PID: 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: P00/3/0/1
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/RSP0/CPU0# show mpls traffic-eng tunnels backup protected-interface

Interface: Gi0/2/0/1 (auto-tunnel backup)
SRLAG: 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: P00/3/0/2 Admin: up Oper: up

Interface: P00/3/0/1 (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/RSP0/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
   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)  
Current LSP:  
  Uptime: 00:06:13 (since Tue Aug 14 23:25:41 EST 2012)  

Path info (OSPF 100 area 16909060):  
Node hop count: 1  
Hop0: 23.9.0.2  
Hop1: 2.2.2.2  

Name: tunnel-te3503 Destination: 4.4.4.4 (auto-tunnel mesh)  
Status:  
  Admin: up Oper: down Path: not valid Signalling: Down  
  path option 10, type dynamic  
  Last PCALC Error: Tue Aug 14 23:31:26 2012  
  Info: No path to destination, 4.4.4.4 (affinity)  
  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: 2 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  

Displayed 3 (of 7) heads, 0 (of 3) midpoints, 0 (of 0) tails Displayed 2 up, 1 down, 0 recovering, 0 recovered heads

Related Topics

- Enabling an AutoTunnel Backup, on page 155  
- Removing an AutoTunnel Backup, on page 156  
- Establishing MPLS Backup AutoTunnels to Protect Fast Reroutable TE LSPs, on page 157  
- Establishing Next-Hop Tunnels with Link Protection, on page 158  
- Backup AutoTunnels, on page 109

**Configure Point-to-Multipoint TE: Examples**

These configuration examples show how to configure Point-to-Multipoint TE:
P2MP Topology Scenario: Example

This section describes a typical scenario of point-to-multipoint traffic engineering topology. This figure illustrates the P2MP topology.

Figure 19: P2MP Topology

This head router describes the configuration at head node. This router does the imposition of MPLS at head node.

```
interface tunnel-mte1
   ipv4 unnumbered Loopback0
   destination 1.1.1.1
   path-option 1 explicit name path-to-tail1
   destination 2.2.2.2
   path-option 1 explicit name path-to-tail2
   fast-reroute
mpls traffic-eng
   interface GigabitEthernet0/1/3/0
   interface GigabitEthernet0/1/3/7
multicast-routing
   address-family ipv4
tsf
   interface all enable
   address-family ipv6
tsf
   interface all enable
! router igmp
   vrf default
   interface tunnel-mte1
      static-group 232.0.0.1 192.168.10.1
!```

This mid router describes the configuration at mid node. This router performs the role of MPLS label replication at mid node.

```
mpls traffic-eng
   interface POS0/2/0/0
   interface POS0/2/0/1
      backup-path tunnel-te 1000
!```
Configure Point-to-Multipoint TE: Examples

This tail router describes the configuration at tail node. This router performs the role of MPLS disposition at tail node.

```plaintext
interface TenGigE0/3/0/3  
!  interface GigabitEthernet0/2/5/0  
!  
This tail router describes the configuration at tail node. This router performs the role of MPLS disposition at tail node.
mpls traffic-eng
  interface POS0/0/3/0
  !
  !
  multicast-routing
    address-family ipv4
      interface all enable
    !
    core-tree-protocol rsvp-te group-list lsm
      static-rpf 192.168.10.1 32 mpls 5.5.5.5
    !
    !
    This configuration describes the Fast Reroute configuration in the MPLS network.
    explicit-path name backup-path-to-tail1
      index 1 next-address strict 198.1.1.2
      index 2 next-address strict 198.1.2.2
    !
    interface tunnel-te1000 <<< backup p2p tunnel
      ipv4 unnumbered Loopback0
      destination 140.140.140.140
      path-option 1 explicit name backup-path-to-tail1
    
mpls traffic-eng
    interface POS0/2/0/0
    !
    interface POS0/2/0/1
      backup-path tunnel-te 1000
    !
    interface TenGigE0/5/0/4
    !
```

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.

```
!  
Note
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
    !

```

Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide, Release 4.3.x
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
index 4 next-address strict ipv4 unicast 192.168.16.2
!
```

Related Topics
- Path Option for Point-to-Multipoint RSVP-TE, on page 133
- Point-to-Multipoint Traffic-Engineering Overview, on page 130

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
!
```
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 GigabitEthernet0/0/0/1.2
  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
  0.0.0.0/0 156.0.1
  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
!
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 GigabitEthernet0/0/0/0/1
  bandwidth 20000
!
interface GigabitEthernet0/0/0/0/2
  bandwidth 20000
!
interface GigabitEthernet0/0/0/0/1.2
  bandwidth 20000
!
interface GigabitEthernet0/0/0/0/2.2
  bandwidth 20000
!

mpls traffic-eng
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 ldp
router-id 192.168.1.2
nsr
graceful-restart
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/0/2
!
interface GigabitEthernet0/0/0/1.2
!
interface GigabitEthernet0/0/0/0/2.2
!

mpls ldp
router-id 192.168.1.2
nsr
graceful-restart
interface GigabitEthernet0/0/0/0
!
interface GigabitEthernet0/0/0/1
!
interface GigabitEthernet0/0/0/0/2
!
interface GigabitEthernet0/0/0/1.2
!
interface GigabitEthernet0/0/0/0/2.2
!

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
  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

  Point-to-Multipoint Traffic-Engineering Overview, on page 130
  Point-to-Multipoint RSVP-TE, on page 132
  Path Option for Point-to-Multipoint RSVP-TE, on page 133
  Point-to-Multipoint Traffic-Engineering Overview, on page 130

Additional References

For additional information related to implementing MPLS-TE, refer to the following references:

<table>
<thead>
<tr>
<th>Related Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Related Topic</strong></td>
</tr>
<tr>
<td>MPLS-TE commands</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standards</strong></td>
</tr>
<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>
</tr>
</tbody>
</table>

---
Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide, Release 4.3.x
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>

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>
<tr>
<td>Release 6.0</td>
<td>nLight enhancements were introduced.</td>
</tr>
</tbody>
</table>

Prerequisites for Implementing GMPLS UNI

The following prerequisites are required to implement GMPLS 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.

• 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 ASR 9000 Enhanced Ethernet Line Cards:
  ◦ A9K-MOD80-SE: 80G Modular Line Card, Service Edge Optimized
  ◦ A9K-MOD80-TR: 80G Modular Line Card, Packet Transport Optimized

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.

Note

The optical router-ids and interface addresses may or may not be the same as the packet ones.

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.

Note

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>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>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>gmpls optical-uni</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# gmpls optical-uni</td>
</tr>
</tbody>
</table>
### 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>Step 1 configure</td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Step 2 mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3 gmpls optical-uni</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# gmpls optical-uni</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. gmpls 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>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></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
</tbody>
</table>

---

**Purpose**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4 controller dwdm interface</td>
<td>Enters GMPLS UNI controller submode.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-te-gmpls)# controller dwdm 0/1/0/1

RP/0/RSP0/CPU0:router(config-te-gmpls-cntl)#

| Step 5 commit | |
### 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.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>gmpls optical-uni</strong>&lt;br&gt;Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-mpls-te)# gmpls optical-uni</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>controller dwdm interface</strong>&lt;br&gt;Enters GMPLS UNI controller submode.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-te-gmpls)# controller dwdm 0/1/0/1&lt;br&gt;RP/0/RSP0/CPU0:router(config-te-gmpls-cntl)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>tunnel-properties</strong>&lt;br&gt;Enters the submode to configure tunnel-specific information for a GMPLS UNI controller.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-te-gmpls-cntl)# tunnel-properties&lt;br&gt;RP/0/RSP0/CPU0:router(config-te-gmpls-tun)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>tunnel-id number</strong>&lt;br&gt;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><strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-te-gmpls-tun)# tunnel-id 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>destination ipv4 unicast address</strong>&lt;br&gt;Specifies a tunnel destination for a headend router of a GMPLS tunnel. The destination argument is an IPv4 address.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-te-gmpls-tun)# destination ipv4 unicast 10.10.3.4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>path-option 10 no-ero lockdown</strong>&lt;br&gt;Specifies the path-option for a headend router of a GMPLS tunnel.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>An XRO attribute-set can be specified as part of the path-option, if required.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-te-gmpls-tun)# path-option 10 no-ero lockdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>commit</strong></td>
</tr>
</tbody>
</table>
### 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>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 GMPLS UNI configuration submode.</td>
</tr>
</tbody>
</table>

```
Example:
RP/0/RSP0/CPU0:router(config)# mpls traffic-eng
```

| Step 3 | gmpls optical-uni | Enters GMPLS UNI controller submode. |

```
Example:
RP/0/RSP0/CPU0:router(config-mpls-te)# gmpls optical-uni
```

| Step 4 | controller dwdm interface | Enters the submode to configure tunnel-specific information for a GMPLS UNI controller. |

```
Example:
RP/0/RSP0/CPU0:router(config-te-gmpls)# controller dwdm 0/1/0/1
```

| Step 5 | tunnel-properties | Specifies the priority for a GMPLS tunnel. The default priority value is 7 for both setup and hold priorities. |

```
Example:
RP/0/RSP0/CPU0:router(config-te-gmpls-cntl)# tunnel-properties
```

| Step 6 | priority setup-priority hold-priority | |

```
Example:
RP/0/RSP0/CPU0:router(config-te-gmpls-tun)# priority 3 2
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>record-route</td>
<td>Enables record-route functionality for a GMPLS tunnel.</td>
</tr>
<tr>
<td>signalled-name</td>
<td>Configures signalled-name for a GMPLS tunnel.</td>
</tr>
<tr>
<td>logging events</td>
<td>Configure events to generate system log messages when state changes occur on the GMPLS tunnel.</td>
</tr>
<tr>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

### 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
### 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></td>
</tr>
<tr>
<td>Step 2 <code>mpls traffic-eng</code></td>
<td>Enters MPLS-TE configuration mode.</td>
</tr>
<tr>
<td>Example: <code>mpls traffic-eng</code></td>
<td></td>
</tr>
<tr>
<td>Step 3 <code>attribute-set xro name</code></td>
<td>Configures an XRO attribute-set for a GMPLS tunnel.</td>
</tr>
<tr>
<td>Example: <code>attribute-set xro attrset01</code></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: <code>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</code></td>
<td></td>
</tr>
<tr>
<td>Step 5 <code>commit</code></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>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>lmp</td>
<td>Enters LMP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# lmp</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>gmpls optical-uni</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-lmp)# gmpls optical-uni</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>router-id ipv4 unicast address</td>
<td>Configures the LMP unicast router ID for GMPLS.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni)# router-id ipv4 unicast 10.10.4.4</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
<td></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><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> lmp</td>
<td>Enters LMP configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# lmp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> gmpls optical-uni</td>
<td>Enters GMPLS UNI configuration submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-lmp)# gmpls optical-uni</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> neighbor name</td>
<td>Specifies an LMP neighbor for GMPLS and enters LMP GMPLS UNI neighbor configuration submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni)# neighbor nbr1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ipcc routed</td>
<td>Specifies the LMP neighbor IPCC configuration for GMPLS UNI.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni-nbr-nbr1)# ipcc routed</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> router-id ipv4 unicast address</td>
<td>Configures the LMP unicast router ID for GMPLS.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni-nbr-nbr1)# router-id ipv4 unicast 10.10.4.4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> commit</td>
<td></td>
</tr>
</tbody>
</table>
### 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></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>lmp&lt;br&gt;Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config)# lmp</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>gmpls optical-uni&lt;br&gt;Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-lmp)# gmpls optical-uni</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>controller dwdm controller&lt;br&gt;Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni)# controller dwdm 0/4/0/0</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>neighbor name&lt;br&gt;Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni-cntl)# neighbor nbr1</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>

---

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6</td>
<td>link-id ipv4 unicast address</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni-cntl)# link-id ipv4 unicast 10.2.2.4</td>
</tr>
<tr>
<td>Step 7</td>
<td>neighbor link-id ipv4 unicast address</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni-cntl)# neighbor link-id ipv4 unicast 10.10.4.4</td>
</tr>
<tr>
<td>Step 8</td>
<td>neighbor interface-id unnumbered interface-id</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-lmp-gmpls-uni-cntl)# neighbor interface-id unnumbered 17</td>
</tr>
<tr>
<td>Step 9</td>
<td>commit</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</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# <code>rsvp</code></td>
</tr>
<tr>
<td>3</td>
<td><code>controller dwdm interface</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp)# <code>controller dwdm 0/1/0/1</code></td>
</tr>
<tr>
<td>4</td>
<td><code>signalling refresh out-of-band interval interval</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-cntl)# <code>signalling refresh out-of-band interval 200</code></td>
</tr>
<tr>
<td>5</td>
<td><code>signalling refresh out-of-band missed miss-count</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-rsvp-cntl)# <code>signalling refresh out-of-band missed 30</code></td>
</tr>
<tr>
<td>6</td>
<td><code>commit</code></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
gmpls optical-uni
controller dwdm 0/1/0/1
tunnel-properties
tunnel-id 100
```
Configuring Tail UNI-C for a GMPLS Tunnel: Example

This example shows the minimal tail UNI-C configuration require to establish a GMPLS tunnel:

```
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>destination 100.20.20.20</td>
<td>path-option 10 no-ero</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>lmp</td>
<td>gmpls optical-uni</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>router-id 100.11.11.11</td>
<td>neighbor nbr_A ipcc routed</td>
</tr>
<tr>
<td>neighbor router-id ipv4 unicast 100.12.12.12</td>
<td></td>
</tr>
<tr>
<td>controller dwdm 0/1/0/1</td>
<td>neighbor nbr_A link-id ipv4 unicast 192.168.100.1</td>
</tr>
<tr>
<td>neighbor router-id ipv4 unicast 192.168.100.2</td>
<td>neighbor link-id ipv4 unicast 192.168.100.1 unnumbered 13</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>controller dwdm 0/1/0/1</td>
<td>neighbor_nbr_B link-id ipv4 unicast 192.168.103.2</td>
</tr>
<tr>
<td>neighbor link-id ipv4 unicast 192.168.103.1</td>
<td>neighbor interface-id unnumbered 22</td>
</tr>
</tbody>
</table>
```

**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.
Configuring LSP Diversity: Example

This example shows the configuration for two diverse LSPs:

```plaintext
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

mpls 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:

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMPLS UNI commands</td>
<td><strong>GMPLS UNI Commands</strong> module in <em>Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference</em></td>
</tr>
<tr>
<td>MPLS Traffic Engineering commands</td>
<td><strong>MPLS Traffic Engineering commands</strong> module in <em>Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference</em></td>
</tr>
<tr>
<td>RSVP commands</td>
<td><strong>RSVP commands</strong> module in <em>Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference</em></td>
</tr>
<tr>
<td>Getting started material</td>
<td><em>Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide</em></td>
</tr>
<tr>
<td>Related Topic</td>
<td>Document Title</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Configuring AAA Services module in Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standard</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>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

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<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 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 ASR 9000 Series Aggregation Services Router MPLS Command Reference.

Feature History for Implementing MPLS OAM

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.1.0</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Prerequisites for MPLS LSP Ping and Traceroute for P2MP, page 277
- MPLS Network Management with MPLS LSP Ping and MPLS LSP Traceroute, page 278
- Roles of Various Routers, page 278
- P2MP Ping, page 279
- P2MP Traceroute, page 280
- MPLS OAM Support for BGP 3107, page 281
- IP-Less MPLS-TP Ping and MPLS-TP Traceroute, page 281
- Configure the Ping and Traceroute: Example, page 281

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 4.1.0 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 20: Elements of P2MP TE Network illustrates the elements of P2MP TE network.

Figure 20: 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 ASR 9000 Series Aggregation Services Router MPLS Command Reference*. 
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.

IP-Less MPLS-TP Ping and MPLS-TP Traceroute

According to RFC-6426, IP-Less MPLS-TP ping and MPLS-TP traceroute with the ACH header, if a node receives an MPLS-TP ping or traceroute request packet over ACH, without IP or UDP headers, the node drops the echo request packet and does not send a response when:

- the reply mode is 4
- the node does not have a return MPLS LSP path to the echo request source.

If a node receives an MPLS echo request with a reply mode other than 4 (i.e., reply via application-level control channel), the node responds to using that reply mode. If the node does not support the reply mode requested, or is unable to reply using the requested reply mode in any specific instance, the node drops the echo request packet and does not send a response.

For more information about ping and traceroute, see Implementing MPLS OAM chapter in the Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide. For more information about ping and traceroute commands, see MPLS OAM Commands chapter in the Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference.

Configure the Ping and Traceroute: Example

This section contains examples of the ping and traceroute commands, based on this topology.

This example shows multiple destinations set on the assigned LSP path.

```
RP/0/RSP0/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
```

Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide, Release 4.3.x
This example shows an extract of the ping command.

```bash
# 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, '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 = 154/232/302 ms
```

This example shows an extract of the ping command with the jitter option.

```bash
RP/0/RSP0/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.

```bash
RP/0/RSP0/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, 
'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 = 105/178/237 ms

RP/0/RSP0/CP/0# 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: 0x0/0xffffffff
  Metric Type: TE (default)
  Fast Reroute: Not Enabled, Protection Desired: None
  Record Route: Not Enabled

Destination summary: (3 up, 0 down, 0 disabled) Affinity: 0x0/0xffffffff
  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 (CTU)
  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)
Configure the Ping and Traceroute: Example

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/RSP0/CP0: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:

Codes: '!i' - 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 192.168.170.1
! 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 ping command with the responder-id of R3.

RP/0/RSP0/CP0: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: '!i' - 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 192.168.170.1

Success rate is 100 percent (1 received reply/1 expected reply),
round-trip min/avg/max = 179/179/179 ms
This example shows an extract of the traceroute command with the ttl option.

RP/0/RSP0/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


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: Bud]
[L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]

! 2 192.168.140.2 213 ms [Estimated Role: Egress]
! 2 192.168.170.1 254 ms [Estimated Role: Egress]

! 3 192.168.222.2 108 ms [Estimated Role: Bud]
[L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]

! 3 192.168.170.1 199 ms [Estimated Role: Egress]
! 3 192.168.140.2 199 ms [Estimated Role: Egress]

! 4 192.168.170.1 198 ms [Estimated Role: Egress]
! 4 192.168.222.2 206 ms [Estimated Role: Bud]
[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 responder-id option.

RP/0/RSP0/CPU0:router# traceroute mpls traffic-eng tunnel-mte 10 responder-id 13.0.0.1
Mon Apr 12 12:18:01.994 EST

Tracing MPLS MTE Label Switched Path on tunnel-mte10, timeout is 2.2 seconds


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]

d 2 192.168.222.2 118 ms [Estimated Role: Branch]
[L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]

! 2 192.168.170.1 244 ms [Estimated Role: Egress]

d 3 192.168.222.2 141 ms [Estimated Role: Branch]
[L] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]

! 3 192.168.170.1 204 ms [Estimated Role: Egress]

d 4 192.168.222.2 110 ms [Estimated Role: Branch]
This example shows an extract of the traceroute command with the jitter option.

```
RP/0/RSP0/CPU0:router# traceroute mpls traffic-eng tunnel-mte 10 responder-id 13.0.0.1 ttl 4 jitter 500
Mon Apr 12 12:19:00.292 EST
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, 'U' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'R' - 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]
[d] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]

! 2 192.168.170.1 428 ms [Estimated Role: Egress]
[d] DDMAP 0: 192.168.140.2 192.168.140.2 MRU 1500 [Labels: 16001 Exp: 0]

! 3 192.168.170.1 327 ms [Estimated Role: Egress]
[d] 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]
Implementing MPLS Transport Profile

This module describes how to implement MPLS transport profile (MPLS-TP) on the router. MPLS-TP supported by IETF enables the migration of transport networks to a packet-based network that efficiently scale to support packet services in a simple and cost-effective way. MPLS-TP combines the necessary existing capabilities of MPLS with additional minimal mechanisms in order that it can be used in a transport role. MPLS transport profile enables you to create tunnels that provide the transport network service layer over which IP and MPLS traffic traverse.

Feature History for Implementing MPLS Transport Profile

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 4.2.0</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Restrictions for MPLS-TP, page 287
- Information About Implementing MPLS Transport Profile, page 288
- How to Implement MPLS Transport Profile, page 293

Restrictions for MPLS-TP

- Penultimate hop popping is not supported. Only ultimate hop popping is supported, because label mappings are configured at the MPLS-TP endpoints.
- MPLS-TP links must be configured with IP addresses.
- IPv6 addressing is not supported.

L2VPN Restrictions

- Pseudowire ID Forward Equivalence Class (FEC) (type 128) is supported, but generalized ID FEC (type 129) is not supported.
Information About Implementing MPLS Transport Profile

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

MPLS Transport Profile

MPLS Transport Profile (TP) enables you to create tunnels that provide the transport network service layer over which IP and MPLS traffic traverse. MPLS-TP tunnels enable a transition from Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) time-division multiplexing (TDM) technologies to packet switching, to support services with high bandwidth utilization and low cost. Transport networks are connection oriented, statically provisioned, and have long-lived connections. Transport networks usually avoid control protocols that change identifiers like labels. MPLS-TP tunnels provide this functionality through statically provisioned bidirectional label switched paths (LSPs). This figure shows the MPLS-TP tunnel:

![Figure 21: MPLS Transport Profile Tunnel](image)

MPLS-TP combines the necessary existing capabilities of MPLS with additional minimal mechanisms in order that it can be used in a transport role. You can set up MPLS-TP through a CLI or a network management system.

MPLS-TP tunnels have these characteristics:

- An MPLS-TP tunnel can be associated with working LSP, protect LSP, or both LSP
- Statically provisioned bidirectional MPLS-TP label switched paths (LSPs)
- Symmetric or asymmetric bandwidth reservation
- 1:1 path protection with revertive mode for MPLS-TP LSP with revertive mode for MPLS-TP LSP
• Use of Generic Alert Label (GAL) and Generic Associated Channel Header (G-ACH) to transport control packets; for example, BFD packets and pseudowire OAM packets
• BFD is used as a continuity check (CC) mechanism over MPLS-TP LSP
• Remote Defect Indication (RDI) based on BFD
• Fault OAM functions

These services are supported over MPLS-TP tunnels:
• Dynamic spoke pseudowire (for H-VPLS) over static MPLS-TP tunnels.
• Static spoke pseudowire (for H-VPLS) over static MPLS-TP tunnels.
• MS-PW services where static and dynamic pseudowire segments can be concatenated.
• MPLS ping and traceroute over MPLS TP LSP and PW.
• Static routes over MPLS-TP tunnels.
• Pseudowire redundancy for static pseudowire.
• VPWS using static or dynamic pseudowire pinned down to MPLS-TP tunnels.
• VPLS and H-VPLS using static or dynamic pseudowire pinned down to MPLS-TP tunnels.

Bidirectional LSPs

MPLS transport profile (MPLS-TP) LSPs are bidirectional and congruent where LSPs traverse the same path in both directions. An MPLS-TP tunnel can be associated with either working MPLS-TP LSP, protect MPLS-TP LSP, or both. The working LSP is the primary LSP backed up by the protect LSP. When a working LSP goes down, protect LSP is automatically activated. In order for an MPLS-TP tunnel to be operationally up, it must be configured with at least one LSP.

MPLS-TP Path Protection

Path protection provides an end-to-end failure recovery mechanism (that is, full path protection) for MPLS-TP tunnels. MPLS-TP LSPs support 1:1 path protection. You can configure the working and protect LSPs as part of configuring the MPLS-TP tunnel. The working LSP is the primary LSP used to route traffic, while the protect LSP is a backup for a working LSP. If the working LSP fails, traffic is switched to the protect LSP until the working LSP is restored, at which time traffic forwarding reverts back to the working LSP (revertive mode).

Fault OAM Support

The fault OAM protocols and messages support the provisioning and maintenance of MPLS-TP tunnels and bidirectional LSPs:
• **Generic Associated Channel**

Generic Associated Channel (G-ACh) is the control channel mechanism associated with MPLS LSPs in addition to MPLS pseudowire. The G-ACh Label (GAL) (Label 13) is a generic alert label to identify the presence of the G-ACh in the label packet. It is taken from the reserved MPLS label space.

G-ACh or GAL is used to support in-band OAMs of MPLS-TP LSPs and pseudowires. The OAM messages are used for fault management, connection verification, continuity check and other functions.

These messages are forwarded along the specified MPLS LSP:

- **OAM Fault Management**: Alarm Indication Signal (AIS), Link Down Indication (LDI), and Lock Report (LKR) messages (GAL with fault-OAM channel)
- **OAM Connection Verification**: Ping and traceroute messages (GAL with IP channel)
- **BFD messages** (GAL with BFD channel)

These messages are forwarded along the specified pseudowire:

- Static pseudowire OAM messages (static pseudowire status)
- Pseudowire ping and traceroute messages

• **Fault Management: Alarm Indication Signal (AIS), Link Down Indication (LDI), and Lock Report (LKR) messages**

LDI messages are generated at midpoint nodes when a failure is detected. The midpoint sends the LDI message to the endpoint that is reachable with the existing failure. The midpoint node also sends LKR messages to the reachable endpoint, when an interface is administratively down.

AIS messages are not generated by Cisco platforms, but are processed if received. By default, the reception of LDI and LKR on the active LSP at an endpoint will cause a path protection switchover, while AIS will not.

• **Fault Management: Emulated Protection Switching for LSP Lockout**

You can implement a form of *Emulated Protection Switching* in support of LSP Lockout using customized fault messages. When a Cisco Lockout message is sent, it does not cause the LSP to be administratively down. The Cisco Lockout message causes a path protection switchover and prevents data traffic from using the LSP. The LSP's data path remains up so that BFD and other OAM messages can continue to traverse it. Maintenance of the LSP can take place such as reconfiguring or replacing a midpoint LSR. BFD state over LSP must be *up* and MPLS ping and traceroute can be used to verify the LSP connectivity, before the LSP is put back into service by removing the lockout. You cannot lockout working and protect LSPs simultaneously.

• **LSP ping and traceroute**

For MPLS-TP connectivity verification, you can use `ping mpls traffic-eng tunnel-tp` and `traceroute mpls traffic-eng tunnel-tp` commands. You can specify that the echo requests be sent along the working LSP or the protect LSP. You can also specify that the echo request be sent on a locked out MPLS-TP tunnel LSP (either working or protect) if the working or protect LSP is explicitly specified.
Continuity Check through BFD

BFD session is automatically created on MPLS-TP LSPs with default parameters. You can override the default BFD parameters either through global commands or per-tunnel commands. Furthermore, you can optionally specify different BFD parameters for standby LSPs. For example, when an LSP is in standby, BFD hello messages can be sent at smaller frequency to reduce line-card CPU usage. However, when a standby LSP becomes active (for example, due to protection switching), nominal BFD parameters are used for that LSPs (for example, to run BFD hello messages at higher frequency). For more information about BFD, see the Configuring Bidirectional Forwarding Detection on the Cisco ASR 9000 Series Router in the Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide.

MPLS-TP Links and Physical Interfaces

MPLS-TP link IDs may be assigned to physical interfaces only. Bundled interfaces and virtual interfaces are not supported for MPLS-TP link IDs.

The MPLS-TP link is used to create a level of indirection between the MPLS-TP tunnel and midpoint LSP configuration and the physical interface. The MPLS-TP link-id command is used to associate an MPLS-TP link ID with a physical interface and next-hop node address.

Multiple tunnels and LSPs may then refer to the MPLS-TP link to indicate they are traversing that interface. You can move the MPLS-TP link from one interface to another without reconfiguring all the MPLS-TP tunnels and LSPs that refer to the link.

Link IDs must be unique on the router or node. For more information, see the Configuring MPLS-TP Links and Physical Interfaces section.

Tunnel LSPs

Tunnel LSPs, whether endpoint or midpoint, use the same identifying information. However, it is entered differently.

- A midpoint consists of a forward LSP and a reverse LSP. A MPLS-TP LSP mid point is identified by its name, and forward LSP, reverse LSP, or both are configured under a submode.

- At the midpoint, determining which end is source and which is destination is arbitrary. That is, if you are configuring a tunnel between your router and a coworker's router, then your router is the source. However, your coworker considers his or her router to be the source. At the midpoint, either router could be considered the source. At the midpoint, the forward direction is from source to destination, and the reverse direction is from destination to source. For more information, see the Configuring MPLS-TP LSPs at Midpoints section.

- At the midpoint, the LSP number does not assume default values, and hence must be explicitly configured.

- At the endpoint, the local information (source) either comes from the global node ID and global ID, or from locally configured information using the source command after you enter the interface tunnel-tp number command, where number is the local or source tunnel-number.

- At the endpoint, the remote information (destination) is configured using the destination command after you enter the interface tunnel-tp number command. The destination command includes the destination
node ID, optionally the global ID, and optionally the destination tunnel number. If you do not specify the destination tunnel number, the source tunnel number is used.

**MPLS-TP IP-less support**

Generally, MPLS-TP functionality can be deployed with or without an IP address. However, the main motivation for the IP-less model is this: an LSR can be inserted into an MPLS-TP network without changing the configurations on adjacent LSRs. In the past Cisco IOS-XR MPLS-TP release, if an interface does not have a valid IP address, BFD packets cannot be transmitted over that link, and hence MPLS-TP LSP cannot be brought up on that link. In this release, the IP-less TP link operates only in a **point-to-point** mode.

This feature, therefore, makes the need for an IP address on a TP link optional. You may deploy LSRs running Cisco IOS-XR in MPLS-TP networks with or without an IP address. With such extra flexibility, LSRs running Cisco IOS-XR can be easily deployed not only with LSRs running IOS, but with LSRs from other vendors too.

**MPLS-TP LSP Wrapping**

In the MPLS-TP LSP Wrapping protection scheme, a protected MPLS-TP tunnel is associated with a working LSP and protect LSP. This helps to prevent traffic loss as soon as a mid-point LSR detects a failure at physical layer rather than waiting for BFD to time-out. Also, a delay in activating protection switch due to mid-point failure does not further increase the traffic loss.

MPLS-TP LSP wrapping has to enabled only on the MID node. MPLS-TP LSP wrapping helps in detecting mid-link failure scenarios; other failures and failures on end node is detected by BFD timeout and TP-OAM message.

As shown in the figure below, when an LSR (e.g., Router B) detects a failure, it forwards the incoming traffic over an impacted LSP onto the reverse LSP, if it exists. The traffic re-directed into the reverse LSP is loopback.
traffic. Looping back traffic is carried out by the forwarding engine without control plane's involvement. The label stack of a loopback packet will be modified such that the source of the traffic identifies the packet.

**Figure 22: MPLS-TP LSP Wrapping Mechanism**

When the forwarding engine at an end-point recognizes a packet from loopback traffic, it forwards the packet on protect LSP. As BFD packets over impacted LSPs are also looped-back, the end-point will drop such BFD packets so that BFD sessions over the impacted LSPs is timed-out and protection switching is activated. Optionally, when an end-point receives the first looped-back packet, it activates protection switching.

A working LSP remains wrapped until protection switching is activated. Once activated, protect LSP will carry traffic as usual. When failure is removed and BFD session comes back up resulting in activation of working LSP.

**How to Implement MPLS Transport Profile**

MPLS Transport Profile (MPLS-TP) supported by IETF enables the migration of transport networks to a packet-based network that efficiently scale to support packet services in a simple and cost effective way. These procedures are used to implement MPLS-TP:
Configuring the Node ID and Global ID

Perform this task to configure node ID and global ID on the router.

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. tp
4. node-id node-id
5. global-id num
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>Enters MPLS TE configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mpls traffic-eng</td>
<td>Enters MPLS transport profile (TP) configuration mode. You can configure MPLS TP specific parameters for the router from this mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>tp</td>
<td>Specifies the default MPLS TP node ID, which is used as the default source node ID for all MPLS TP tunnels configured on the router.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# mpls tp</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>node-id node-id</td>
<td>The node ID is a 32-bit number represented in IPv4 address format, and can be optionally assigned to each node.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-tp)# node-id 10.0.0.1</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>global-id num</td>
<td>Specifies the default global ID used for all endpoints and midpoints. This command makes the node ID globally unique in a multi-provider tunnel. Otherwise, the node ID is only locally meaningful.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-tp)# global-id 10</td>
<td>The global ID is a 32-bit number, and can be assigned to each node.</td>
</tr>
<tr>
<td>Step 6</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Pseudowire OAM Attributes

Perform this task to configure pseudowire OAM attributes.

**SUMMARY STEPS**

1. configure
2. l2vpn
3. pw-oam refresh transmit *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>
<tr>
<td>Step 2 l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# l2vpn</td>
</tr>
<tr>
<td>Step 3 pw-oam refresh transmit <em>value</em></td>
<td>Specifies the OAM timeout refresh intervals.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn)# pw-oam refresh transmit 20</td>
</tr>
<tr>
<td>Step 4 commit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring the Pseudowire Class

When you create the pseudowire class, you specify the parameters of the pseudowire, such as the use of the control word and preferred path.

**SUMMARY STEPS**

1. configure
2. l2vpn
3. pw-class *name*
4. encapsulation mpls
5. preferred-path interface tunnel-tp *tunnel-number*
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>
<tr>
<td>Step 2</td>
<td>l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>pw-class <em>name</em></td>
<td>Creates a pseudowire OAM class named foo and enters pseudowire OAM class configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn)# pw-class foo</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>encapsulation mpls</td>
<td>Sets pseudowire encapsulation to MPLS.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>preferred-path interface tunnel-tp <em>tunnel-number</em></td>
<td>Specifies TP tunnel interface 10 for the preferred-path.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn-pwc-mpls)# preferred-path interface tunnel-tp 10</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>commit</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring the Pseudowire

Perform this task to configure the pseudowire.

### SUMMARY STEPS

1. configure
2. interface *type interface-path-id*
3. pseudowire-class *class-name*
4. encapsulation mpls
5. preferred-path interface tunnel-tp *tunnel-number*
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><code>interface type interface-path-id</code></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td><code>RP/0/RSP0/CPU0:router(config)# interface tunnel-tp 20</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>pseudowire-class class-name</code></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td><code>RP/0/RSP0/CPU0:router(config-if)# pseudowire-class foo</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>encapsulation mpls</code></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td><code>RP/0/RSP0/CPU0:router# encapsulation mpls</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>preferred-path interface tunnel-tp tunnel-number</code></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td><code>RP/0/RSP0/CPU0:router# preferred-path interface tunnel-tp 10</code></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>When a PW class with tunnel-tp interface as a preferred path is defined, this specified class can be associated with any PW.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>commit</code></td>
</tr>
</tbody>
</table>

### Configuring the MPLS TP Tunnel

On the endpoint routers, create an MPLS TP tunnel and configure its parameters.
SUMMARY STEPS

1. configure
2. interface tunnel-tp number
3. description tunnel-desc
4. bandwidth num
5. source source node-ID
6. destination destination node-ID [global-id destination global ID] tunnel-id destination tunnel ID
7. working-lsp
8. in-label num
9. out-label mpls label out-link link ID
10. lsp-number value
11. exit
12. protect-lsp
13. in-label num
14. out-label mpls label out-link link ID
15. lsp-number value
16. exit
17. 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-tp number</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# interface tunnel-tp 10</td>
<td>Enters tunnel tp interface configuration mode. The range is from 0 to 65535.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>description tunnel-desc</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-if)# description head-end tunnel</td>
<td>Specifies a tunnel tp description.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>bandwidth num</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-if)# tp bandwidth 1000</td>
<td>Specifies the tunnel bandwidth in kbps. The range is from 0 to 4294967295.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong> &lt;br&gt;source source node-ID</td>
<td>Specifies the source node of the tunnel.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# source 10.0.0.1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> &lt;br&gt;destination destination node-ID [global-id destination global ID] tunnel-id destination tunnel ID</td>
<td>Specifies the destination node of the tunnel.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# destination 10.0.0.1 global-id 10 tunnel-id 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> &lt;br&gt;working-lsp</td>
<td>Specifies a working LSP, also known as the primary LSP. This LSP is used to route traffic.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# working-lsp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> &lt;br&gt;in-label num</td>
<td>Specifies the in-label.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if-work)# in-label 111</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> &lt;br&gt;out-label mpls label out-link link ID</td>
<td>Specifies the out-label.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if-work)# out-label 111 out-link 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> &lt;br&gt;lsp-number value</td>
<td>Specifies the LSP ID of the working LSP.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if-work)# lsp-number 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> &lt;br&gt;exit</td>
<td>Exits from working LSP interface configuration mode.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if-work)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> &lt;br&gt;protect-lsp</td>
<td>Specifies a backup for a working LSP. If the working LSP fails, traffic is switched to the protect LSP until the working LSP is restored, at which time traffic forwarding reverts back to the working LSP.</td>
</tr>
<tr>
<td>Example: &lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# protect-lsp</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>in-label <em>num</em></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if-protect)# in-label 113</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>out-label <em>mpls label out-link link ID</em></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if-protect)# out-label 112 out-link 2</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>lsp-number <em>value</em></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if-protect)# lsp-number 10</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if-protect)# exit</td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

### Configuring MPLS-TP LSPs at Midpoint

Perform this task to configure the MPLS-TP LSPs at the midpoint router.

**Note**
When configuring the LSPs at the midpoint routers, make sure that the configuration does not reflect traffic back to the originating node.

### SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. tp mid *name*
4. tunnel-name *name*
5. lsp-number *value*
6. source *node-ID* tunnel-id *number*
7. destination *node-ID* tunnel-id *number*
8. 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>Enters MPLS TE configuration mode.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>tp mid name</td>
</tr>
<tr>
<td>Example:</td>
<td>Specifies the MPLS-TP tunnel mid-point identifier.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# tp mid foo</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>tunnel-name name</td>
</tr>
<tr>
<td>Example:</td>
<td>Specifies the name of the tunnel whose mid point is being configured.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-tp-mid)# tunnel-name midtunnel</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>lsp-number value</td>
</tr>
<tr>
<td>Example:</td>
<td>Specifies the LSP ID.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-tp-mid)# lsp-number 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>source node-ID tunnel-id number</td>
</tr>
<tr>
<td>Example:</td>
<td>Specifies the source node ID and tunnel ID.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-tp-mid-fwd)# source 10.0.0.1 tunnel-id 12</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>destination node-ID tunnel-id number</td>
</tr>
<tr>
<td>Example:</td>
<td>Specifies the destination node ID and tunnel ID.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-tp-mid-rev)# source 10.0.0.2 tunnel-id 12</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

### Configuring MPLS-TP Links and Physical Interfaces

MPLS-TP link IDs may be assigned to physical interfaces only.
SUMMARY STEPS

1. configure
2. mpls traffic-eng
3. interface type interface-path-id
4. link-id value next-hop address
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 traffic-eng</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# mpls traffic-eng</td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# interface POS 0/6/0/0</td>
</tr>
<tr>
<td>Step 4</td>
<td>link-id value next-hop address</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te-if)# link-id 22 next-hop 10.1.1.2</td>
</tr>
<tr>
<td>Step 5</td>
<td>commit</td>
</tr>
</tbody>
</table>

Configuring MPLS-TP LSP Wrapping

Perform this task to configure the MPLS-TP LSP wrapping.

---

Note: Bundled interfaces and virtual interfaces are not supported for MPLS-TP link IDs.
When configuring the LSPs at the midpoint routers, make sure that the configuration does not reflect traffic back to the originating node.

**SUMMARY STEPS**

1. configure
2. mpls traffic-eng
3. tp mid name
4. tunnel-name name
5. fast-protect
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>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# mpls traffic-eng</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>tp mid name</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# tp mid midpt1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>tunnel-name name</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# tunnel-name midtunnel</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>fast-protect</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-mpls-te)# fast-protect</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>commit</td>
</tr>
</tbody>
</table>

---

*Note*