

MPLS Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 26.1.x, 26.2.x, 26.3.x

First Published: 2026-02-28

Last Modified: 2026-09-11

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

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.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

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: https://www.cisco.com/c/en/us/about/legal/trademarks.html. 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. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

Preface xiii

Changes to this Document xiii

Communications, Services, and Additional Information xiii

CHAPTER 1

New and Changed MPLS Features 1

New and Changed MPLS Feature Information 1

CHAPTER 2

YANG Data Models for MPLS Features 3

Using YANG Data Models 3

CHAPTER 3

Implementing MPLS Label Distribution Protocol 5

Prerequisites for Implementing MPLS Label Distribution Protocol 6

Restrictions and Recommendations 7

Information About Implementing Cisco MPLS LDP 8

IP LDP Fast Reroute Loop Free Alternate 8

IS-IS 10

Label Acceptance Control (Inbound Filtering) 10

Label Advertisement Control (Outbound Filtering) 10

Conditional label advertisement in label-switched path networks 10

Limitations 12

Configuration 12

Label Switched Paths 13

LDP Control Plane 13

LDP Control Plane: Bindings Advertisement 13

Control Plane Failure 13

Default Transport Address 14

Label Distribution Protocol Discovery Parameters 15
Downstream on Demand 15
Explicit-null and implicit-null labels 16
Label distribution protocol interior gateway protocol synchronization 18
LDP Forwarding 20
Setting up Label Distribution Protocol Graceful Restart 21
Phases in Graceful Restart 21
Recovery with Graceful-Restart 22
LDP Nonstop Routing 23
Local Label Allocation Control 23
Redistributing MPLS LDP Routes into BGP 24
Session protection 25
LDP over RSVP LSR support 27
Increase in RSVP-TE Tunnel Scale for LDP over TE 29
How to Implement MPLS LDP 30
Implementing MPLS Label Distribution Protocol 30
MPLS Restrictions 30
Enabling MLDP 30
Enabling MLDP Make-Before-Break 32
Enabling MLDP MoFRR 33
Enabling MLDP Recursive FEC 35
Enabling MLDP Static Multipoint to Multipoint LSP 36
Enabling MLDP Static Point to Multipoint LSP 38
Exchanging Label Bindings 39
Configuring Label Advertisement Control (Outbound Filtering) 40
Setting Up Implicit-Null-Override Label 41
Setting Up LDP Forwarding 42
Setting Up LDP Neighbors 44
Setting Up LDP NSF Using Graceful Restart 46
Configuring Label Acceptance Control (Inbound Filtering) 49
Configuring LDP IGP Synchronization: ISIS 50
Configure Label Distribution Protocol Targeted Neighbor 52
Configuration Example 52
Running Configuration 52

Configuring Global Transport Address 53
Configuring IPv4 as Transport Preference 54
Configuring LDP Discovery for Active Targeted Hellos 55
Configuring LDP Discovery for Passive Targeted Hellos 58
Configuring LDP Discovery Over a Link 60
Configuring Downstream on Demand 62
Configuring LDP Link: Example 63
Configuring Label Distribution Protocol Nonstop Routing 63
Configure Session Protection 64
Configuring Local Label Allocation Control 64
Configuring Transport Preference Maximum Wait Time 65
Configuring MPLS Label Security 66
Disabling Implicit IPv4 66
Disabling LDP Auto-Configuration 68
Disabling LDP IGP Synchronization: OSPF 69
Disabling MLDP 70
Auto-enable LDP on all TE tunnel interfaces 71
Limitations 73
Configure Auto-enable LDP on all TE Tunnel Interfaces 73
Enabling LDP Auto-Configuration for a Specified OSPF Instance 74
Enabling LDP Auto-Configuration in an Area for a Specified OSPF Instance 76
Implicit IPv4 Disable 77
Running Configuration 77
Verify IP LDP Fast Reroute Loop Free Alternate: Example 78
Configuration Examples for Implementing MPLS LDP 78
Configuring LDP Discovery for Targeted Hellos: Example 78
Configure IP LDP Fast Reroute Loop Free Alternate: Examples 79
Configuring Local Label Allocation Control: Example 80
Configuring LDP with Graceful Restart: Example 81
Configuring LDP Forwarding: Example 81
Configuring LDP Nonstop Forwarding with Graceful Restart: Example 81
Configuring Label Acceptance (Inbound Filtering): Example 81
Configuring LDP Discovery: Example 82
Configuring LDP Auto-Configuration: Example 82

Configuring LDP Neighbors: Example 82
Configuring LDP IGP Synchronization—ISIS: Example 82
Configuring LDP IGP Synchronization—OSPF: Example 83
Label Distribution Protocol Interior Gateway Protocol Auto-configuration 83
Controlling State Advertisements in an mLDP-Only Setup 83
Use Cases For Controlling State Advertisements 86
mLDP-Based MVPN 86
Disable Prefix-LSPs On An L2VPN/PW tLDP Session 88
ECMP and Bundle Hashing with Entropy Label 90
Load Balancing based on the Position of Entropy Label 92
Additional References 94

CHAPTER 4 **Implementing MPLS OAM** 97
IP-Less MPLS-TP Ping and MPLS-TP Traceroute 97
MPLS LSP Ping 97
MPLS LSP Traceroute 99
MPLS OAM using nil FEC 102
Self-ping probe in MPLS-TE 105
Self-ping probe for current LSP 106
Self-ping probe for reoptimized LSP 107
Restriction for self-ping probe support 107
How self-ping probe works 107
Configure the self-ping probe 108
Self-ping probe status 109

CHAPTER 5 **MPLS Static Labeling** 113
Forwarding Labeled Packets 115
Define Label Range and Enable MPLS Encapsulation 116
Identify and Clear Label Discrepancy 117
Configuring Backup within a Forwarding Set 118
Configuring Static LSP Next Hop Resolve 121
Configuring Static LSP Next Hop Resolve with Recursive Prefix 122
MPLS Static Labeling 123
MPLS Static Forwarding Over A BVI 125

MPLS Over GRE Tunnels 129

CHAPTER 6 **Implementing MPLS Traffic Engineering** 135

Prerequisites for Implementing Cisco MPLS Traffic Engineering 137

Overview of MPLS-TE Features 137

How MPLS-TE Works 139

Soft-preemption 140

Soft-preemption over FRR Backup Tunnels 141

SRLG Limitations 141

RSVP-TE Dark Bandwidth Accounting 142

Point-to-Multipoint Traffic-Engineering 143

Point-to-Multipoint Traffic-Engineering Overview 143

Point-to-Multipoint RSVP-TE 145

Point-to-Multipoint Label Switch Path 145

Path Option for Point-to-Multipoint RSVP-TE 145

Point-to-Multipoint Implicit Null 147

Configuring MPLS-TE 147

Building MPLS-TE Topology 147

Teardown and reestablishment of RSVP-TE tunnels 148

Configure Tear down and Reestablishment of RSVP-TE Tunnels 151

Configuring Automatic Bandwidth 152

Configuring Automatic Capacity With Load-Interval Configuration 153

Configuring Auto-Bandwidth Bundle TE++ 156

Restrictions and Guidelines 158

Configure Auto-Bandwidth Bundle TE++ 159

Configuring Auto-Tunnel Backup 161

Removing an AutoTunnel Backup 162

Configuring Auto-Tunnel Mesh 162

Enable LDP over TE Automatically on Cloned Tunnels 163

Attribute Set for Named Tunnels 166

Configure fast reroute 170

Configuring Flexible Name-Based Tunnel Constraints 171

Configuring Forwarding Path 171

Configure an IETF DS-TE tunnel using MAM 171

Configure an IETF DS-TE tunnel using RDM 173
Configuring an MPLS Traffic Engineering Interarea Tunneling 175
MPLS-TE path protection 176
Prerequisites 177
Restrictions 178
Configure MPLS-TE path protection 178
Configuring Next Hop Backup Tunnel 181
Configuring Point-to-Multipoint TE Tunnels 181
Configuring Point-to-Multipoint TE Auto-Tunnels 181
Enabling Soft-Preemption 181
Configuring Pre-Standard DS-TE 182
Configuring SRLG Node Protection 183
SRLG Limitations 183
Creating an MPLS-TE Tunnel 184
Traffic steering using autoroute announce 186
Configure autoroute announce 188
Configure IPv6 routing over IPv4 MPLS-TE tunnels 189
Configure IPv6 routing over IPv6 MPLS-TE tunnels 191
Configure IPv6-only autoroute announce for MPLS-TE tunnels 192
Configure autoroute announce to prevent IP traffic steering over MPLS tunnel 193
Prevention of IP traffic steering to destinations with segment routing prefix SID 193
Enable per-application accounting for interfaces 194
Configure dark bandwidth accounting 195
Configure Autoroute Tunnel as Designated Path 201
Restrictions for Configure Autoroute Tunnel as Designated Path 203
Configure Autoroute Tunnel as Designated Path 203
Configure MPLS over UDP 204
Restriction for Configuring MPLS over UDP 207
Configure MPLS over UDP 207
Improved scale for MPLS over UDP tunnels 212
Limitations and restrictions for improved scale for MPLS over UDP tunnels 213
Configure improved scale for MPLS over UDP tunnels 213
Common Label for IPv4 and IPv6 Address Families 215
Guidelines and Limitations for Using Common Label for IPv4 and IPv6 Address Families 216

Configure a Common Label for IPv4 and IPv6 Address Families 217
Policy-based tunnel selection 218
Guidelines and Limitations 222
Configure Policy-Based Tunnel Selection 223
Configure IPv6 routing over IPv4 MPLS-TE tunnels 230
Configure IPv6 routing over IPv6 MPLS-TE tunnels 231
Event history for MPLS-TE tunnels 233
Benefits of tunnel event history 235
Restrictions for tunnel event history 235
Configure tunnel event history and event count 235
Dynamic bandwidth update for MPLS-TE LSPs 239
Restrictions for existing MPLS-TE LSP bandwidth update 241
Regular expressions in MPLS-TE tunnel names 241
Limitations of using regex in MPLS-TE tunnels 242
Usage guidelines of regex in MPLS-TE tunnel names 243
Regular expressions in MPLS-TE tunnel name filtering commands 244
MPLS-TE Features - Details 245
Configuring Performance Measurement 249
Additional References 250

CHAPTER 7 **RSVP-TE Transport Solution with Flex LSP** 253
Signaling Methods and Object Association for Flex LSPs 254
Associated Bidirectional Co-routed LSPs 254
Restrictions for Flex LSP 255
Key Features supported in Flex LSP 255
Flex LSP Scale Details 257
How to Configure Co-routed Flex LSPs 257
Configuring Co-routed Flex LSPs 257
Verifying the Co-routed Flex LSP Configuration 259
RSVP-TE Transport Solution with Flex LSP Headend 263
Flex LSP headend scale details 267
Limitations and restrictions for flex LSP headend 267
Configure the co-routed flex LSPs headend 268
RSVP-TE transport solution using PCE 274

Limitations and restrictions for RSVP-TE transport solution using PCE 278
How RSVP-TE transport solution with PCE works 278
Configure the RSVP-TE transport solution with flex LSP headend and PCE 281

CHAPTER 8 **Implementing RSVP for MPLS-TE** 289
Usage Guidelines and Limitations 290
Setting up MPLS LSP Using RSVP 291
Overview of RSVP for MPLS-TE Features 291
Bandwidth Reservation Percentage 292
Caveats for Out-of-Sequence 292
Keychain Configuration For RSVP Authentication 292
Configuring RSVP for MPLS-TE 292
Configuring Traffic Engineering Tunnel Bandwidth 293
Confirming DiffServ-TE Bandwidth 293
Global, Interface, and Neighbor Authentication Modes 293
Configure RSVP message authentication globally 294
Configuring RSVP Authentication for an Interface 296
Configuring RSVP Authentication on a Neighbor 297
RSVP Authentication by Using All the Modes: Example 297
MPLS-TE PCE authentication 298
Configure MPLS-TE PCE authentication 299
Configuring Graceful Restart 300
Change the Restart-Time: Example 301
Configure refresh reduction 301
Change the Hello Interval: Example 306
Disable Refresh Reduction: Example 306
RSVP Prefix Filtering 306
Configuring ACL Based Prefix Filtering 306
Configuring RSVP Packet Dropping 307
Enabling RSVP Traps 307
Eliminating Security Associations for RSVP Authentication 308
Extended display and filtering options for interface monitoring 308
RSVP for MPLS-TE Features - Details 311
Additional References 314

CHAPTER 9 **MPLS MTU** 317

Overview of MPLS MTU 318

Restrictions for MPLS MTU 320

Configuration Example for MPLS MTU 320

CHAPTER 10 **Implementing GMPLS UNI** 323

Implementing GMPLS UNI 323

Configure GMPLS UNI 331

Verify GMPLS UNI Configurations 334

Additional References 343



Preface

This guide describes the Cisco IOS XR configurations.

The preface contains the following sections:

- Changes to this Document, on page xiii
- Communications, Services, and Additional Information, on page xiii

Changes to this Document

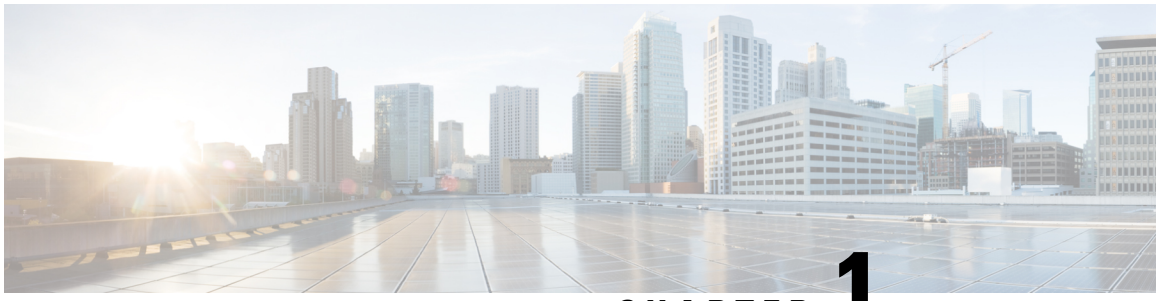
Date	Change Summary
September 2026	Republished for Release 26.3.1
June 2026	Republished for Release 26.2.1
February 2026	Initial release of this document

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

New and Changed MPLS Features

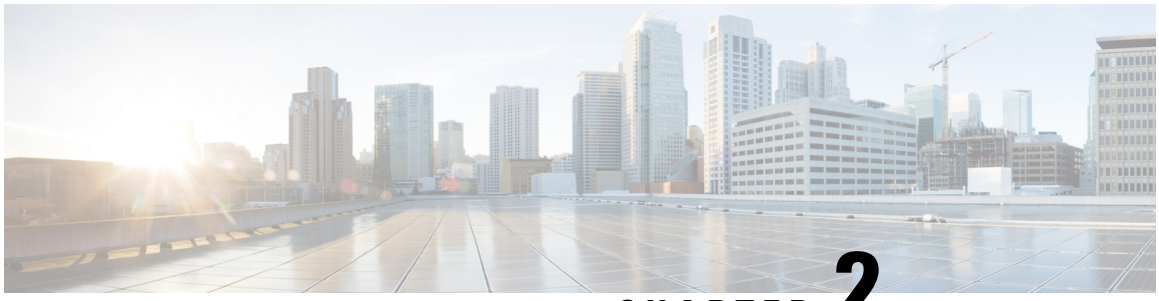
This table summarizes the new and changed feature information for the *MPLS Configuration Guide for Cisco 8000 Series Routers*, and tells you where they are documented.

- New and Changed MPLS Feature Information, on page 1

New and Changed MPLS Feature Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Regular expressions in MPLS-TE tunnel names	This feature was introduced	Release 26.2.1	Regular expressions in MPLS-TE tunnel names
Extended display and filtering options for interface monitoring	This feature was introduced	Release 26.1.1	Configuring RSVP for MPLS-TE



CHAPTER 2

YANG Data Models for MPLS Features

This chapter provides information about the YANG data models for MPLS features.

• Using YANG Data Models, on page 3

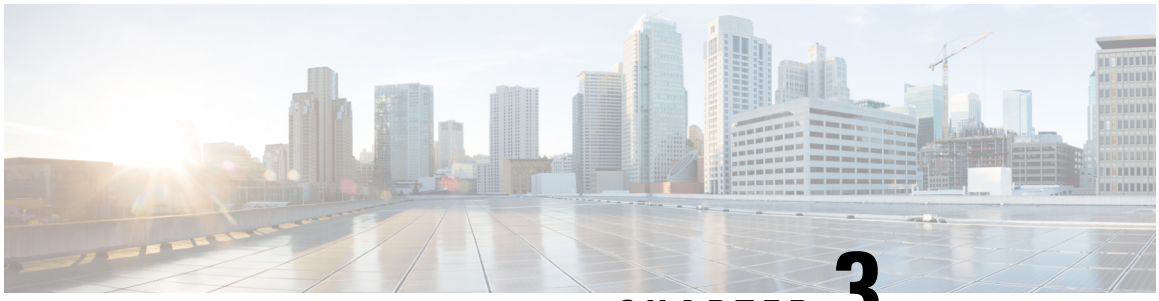
Using YANG Data Models

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the Github repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPaths. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository.

You can also view the data model definitions using the YANG Data Models Navigator tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.



CHAPTER 3

Implementing MPLS Label Distribution Protocol

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Implementing MPLS label distribution protocol	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Implementing MPLS label distribution protocol	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Implementing MPLS label distribution protocol	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Implementing MPLS label distribution protocol	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

In IP forwarding, when a packet arrives at a router the router looks at the destination address in the IP header, performs a route lookup, and then forwards the packet to the next hop. MPLS is a forwarding mechanism in which packets are forwarded based on labels. Label Distribution Protocols assign, distribute, and install the labels in an MPLS environment. It is the set of procedures and messages by which Label Switched Routers (LSRs) establish LSPs through a network by mapping network-layer routing information directly to data-link layer switched paths. These LSPs may have an endpoint at a directly attached neighbor (comparable to IP hop-by-hop forwarding), or may have an endpoint at a network egress node, enabling switching via all intermediary nodes.

LSPs 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 enables LSRs to discover their potential peer routers and to establish LDP sessions with those peers to exchange label binding information. Once label bindings are learned, the LDP is ready to setup the MPLS forwarding plane.

For MPLS LDP, Graceful Out of Resource (OOR) handling is supported from Release 7.3.2 onwards.

- Prerequisites for Implementing MPLS Label Distribution Protocol, on page 6
- Restrictions and Recommendations, on page 7
- Information About Implementing Cisco MPLS LDP, on page 8
- How to Implement MPLS LDP, on page 30
- Configuration Examples for Implementing MPLS LDP, on page 78
- Controlling State Advertisements in an mLDP-Only Setup, on page 83
- Use Cases For Controlling State Advertisements, on page 86
- mLDP-Based MVPN, on page 86
- Disable Prefix-LSPs On An L2VPN/PW tLDP Session, on page 88
- ECMP and Bundle Hashing with Entropy Label, on page 90
- Load Balancing based on the Position of Entropy Label, on page 92
- Additional References, on page 94

Prerequisites for Implementing MPLS Label Distribution Protocol

The following are the prerequisites 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.

Note This point is not appplicable for a Cisco NCS 540 Series Router.

- 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.

Restrictions and Recommendations

The following restrictions and recommendations apply to the MPLS LDP CSC feature:

- Only IPv4 address family is supported for a default or a non-default VRF.
- No T-LDP support in a VRF context.
- An address family under VRF and VRF interface must be configured for non-default VRFs.
- Following scenarios are not supported :
  - Different VRFs between a given PE-CE device pair (VRFs configured on different links and interfaces)
  - LDP/BGP CSC co-existence on a given VRF between a given PE-CE device pair:
    - Single link
    - Parallel links: LDP CSC on one link and BGP CSC on the other
- LDP router-id must be configured per-VRF. If not configured for non-default VRF, LDP computes router-id from available loopback interfaces under the VRF.
- It is recommended to configure a routable discovery transport address under a VRF IPv4 address-family submode for deterministic transport endpoint and connection.
- When LDP CSC is configured and in use:
  - BGP label allocation policy for VRF prefixes must be per-prefix
  - Selective VRF Download (SVD) feature must be disabled
- From Release 24.4.1 onwards, to improve user readability of MPLS route statistics summary when there is a large number of MPLS routes coming from Service Layer API (SL-API) clients, the **show service-layer mpls label**[*label-id* | **client** *client-id*] command now shows a shortened output per path under MPLS. Prior to this release, this command provided detailed output per path under MPLS.

  Example:

  Sample Output - Old Behavior

```
RP/0/RP0/CPU0:ios#show service-layer mpls label
MPLS Label :46002,, tag: 0, distance: 2
    path: 1, 1.1.1.1, via HundredGigE 0/0/0/0,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
        load metric: 0,
        remote address:
        remote labels: 60001,
        interface name: HundredGigE 0/0/0/0,

    path: 2, 2.1.1.1, via HundredGigE 0/0/0/1,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
```

```
        load metric: 0,
        remote address:
        remote labels: 60011,
        interface name: HundredGigE 0/0/0/1,

    path: 3, ……
MPLS Label :46003,, tag: 0, distance: 2
    path: 1, 1.1.1.1, via HundredGigE 0/0/0/0,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
        load metric: 0,
        remote address:
        remote labels: 60001,
        interface name: HundredGigE 0/0/0/0,

    path: 2, 2.1.1.1, via HundredGigE 0/0/0/1,
      ref count: 10, protected bitmap: 0x0, path id: 0, flags: 0x0
        load metric: 0,
        remote address:
        remote labels: 60011,
        interface name: HundredGigE 0/0/0/1,

    path: 3, ……
MPLS Label :46004,, ……
……
```

Sample Output - New Behavior

```
MPLS Label :46002, tag: 0, distance: 2
    path: 1, 1.1.1.1 via HundredGigE 0/0/0/0
    path: 2, 2.1.1.1 via HundredGigE 0/0/0/1
    path: 3, 3.1.1.1 via HundredGigE 0/0/0/2
MPLS Label :46003, tag: 0, distance: 2
    path: 1, 1.1.1.1 via HundredGigE 0/0/0/0
    path: 2, 2.1.1.1 via HundredGigE 0/0/0/1
    path: 3, 3.1.1.1 via HundredGigE 0/0/0/2
.
.
.
```

Information About Implementing Cisco MPLS LDP

To implement MPLS LDP, you should understand these concepts:

IP LDP Fast Reroute Loop Free Alternate

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

The goal of LFA FRR is to reduce failure reaction time to 50 milliseconds by using a pre-computed alternate next hop, in the event that the currently selected primary next hop fails, so that the alternate can be rapidly used when the failure is detected.

This feature targets to address the fast convergence ability by detecting, computing, updating or enabling prefix independent pre-computed alternate loop-free paths at the time of failure.

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

Prerequisites

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

The Label 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.
- In a multi-topology scenerio, 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 *Routing Configuration Guide for Cisco 8000 Series Routers*.

IS-IS

Intermediate System-to-Intermediate System (IS-IS) is an Interior Gateway Protocol (IGP) that advertises link-state information throughout the network to create a picture of the network topology. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to IPv4.

Previously, IS-IS supported registration of only LDP IPv4 sync status change. This has now been enhanced to support registration of notifications of LDP IPv6 sync status change. IS-IS determines the link-metrics to be advertised based on the LDP-IGP sync status on the IPv4 and IPv6 address families.

IS-IS supports non-stop forwarding (NSF) by preserving the LDPv6-IGP sync status across high availability (HA) events of IS-IS process restarts and failover.

IS-IS also supports LDPv6-IGP sync for LFA-FRR by checking the sync status of the backup interface (if it is configured with LDP IPv6 sync).

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.

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*.

Conditional label advertisement in label-switched path networks

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
Conditional label advertisement in label-switched path networks	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.

Conditional label advertisement in label-switched path networks	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Conditional label advertisement in label-switched path networks	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Conditional label advertisement in label-switched path networks	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Conditional label advertisement in label-switched path networks	Release 24.2.11	You can now enhance your network's stability and performance with the streamlined label management. This can be achieved by configuring LDP to advertise labels to peers only when at least one labeled path is available for a prefix. Previously, LDP would advertise local labels to peers even if all next-hop paths for a specific Forwarding Equivalence Class (FEC) had no labels. This release has the following changes: **CLI:** • Introduced a new keyword **unlabelled-all** in **show mpls ldp forwarding** command. • **conditional minimum-one-labelled-nexthop**

LDP Behavior in Label Advertisement

In an LDP-enabled network, when LDP isn't active on certain links, LDP assigns an unlabeled route to tunnels or interfaces which are inoperative. However, LDP continues to advertise labels to its peers even when all possible paths for a specific prefix are unlabeled. This behavior becomes problematic when the forwarding

state of a Forwarding Equivalence Class (FEC) is unstable, causing frequent changes in the network's state (known as flapping). This flapping results in a high volume of label updates and withdrawals, creating unnecessary signaling traffic, known as churn, in the network.

When LDP assigns an unlabeled route to a prefix, it can disrupt Equal-Cost Multi-Path (ECMP) routing, leading to failed paths. Despite of these issues, LDP doesn't automatically retract the assigned label even if the traffic is being forwarded without labels.

With this feature enabled, LDP advertises labels to its peers if there's at least one label path is available for a prefix. If LDP detects that all outgoing paths for a prefix are unlabeled, it withdraws the previously advertised label.

Default Behavior of LSP Networks in Label Advertisement

This feature is disabled by default.

Limitations

This feature doesn't impact BGP route types as LDP allocates and installs no forwarding for the following scenarios:

- Interactions between autonomous systems in BGP, which involves different policies for label distribution and route propagation.
- Carrier Supporting Carrier (CSC) where one carrier provides MPLS services to another carrier.
- Seamless-MPLS to simplify network operations and support end-to-end services.
- Interworking of LDP-based MPLS systems with the BGP-SR (Segment Routing with BGP) systems.

Configuration

Configuration Example

```
Router# configure
Router(config)# mpls ldp
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)# label local advertise conditional minimum-one-labelled-nexthop
Router(config-ldp-af)# exit
```

Verification

To view prefixes with all paths being unlabelled, use **show mpls ldp forwarding next-hop unlabelled-all** command.

```
Router# show mpls ldp forwarding next-hop unlabelled-all

Prefix          Label-In  Label(s)-Out  Outgoing-Interface Next Hop     Flags G S R E
--------------- ------- -------------- ------------ ---------------------------------
14.14.14.14/32  24006     Unlabelled     Gi0/2/0/2         13.13.13.2
15.15.15.15/32  24007     Unlabelled     Gi0/2/0/2         13.13.13.2
16.16.16.16/32  24008     Unlabelled     Gi0/2/0/0         10.10.10.2
                                         Unlabelled        Gi0/2/0/2
```

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.

LDP Control Plane: Bindings Advertisement

LDP base specification allows exchange of IPv4/IPv6 bindings (address/label) on an established session. When both IPv4 and IPv6 address families are enabled under LDP, LDP distributes address/label bindings for both address families to its established peer according to local policies. Following are a few significant points pertaining to bindings support for IPv6:

- LDP allocates/advertises local label bindings for link-local IPv6 address prefixes. If received, such FEC bindings are ignored.
- LDP sends only the Prefix FEC of the single address family type in a FEC TLV and not include both. If such a FEC binding is received, the entire message is ignored.
- LDP sends only the addresses belonging to same address family in a single address list TLV (in address or address withdraw message).

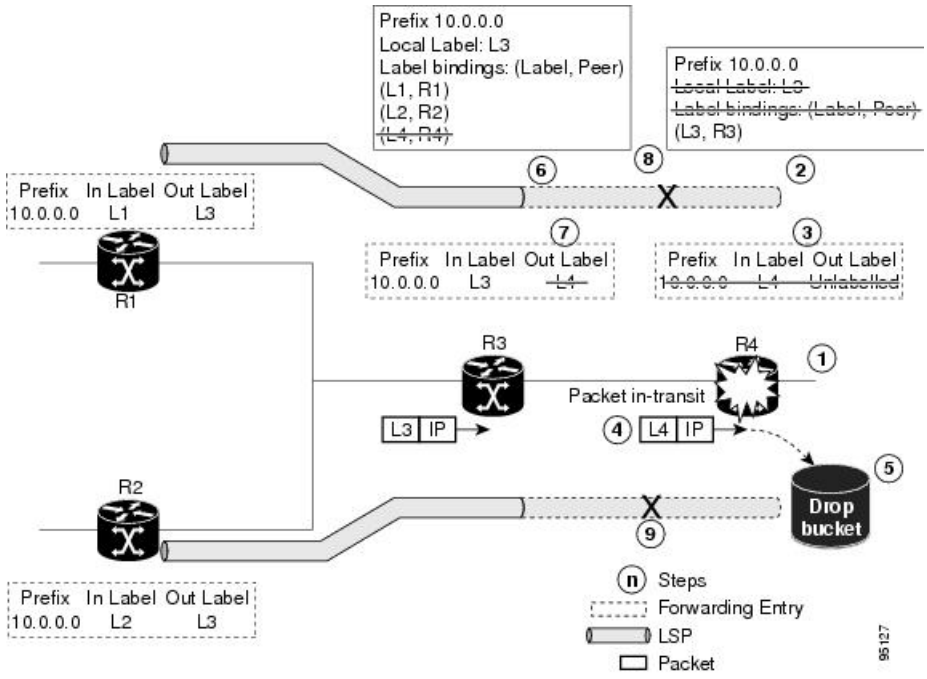
If an address family is not enabled on receiving LSR, LDP discards any bindings received from peer(s) for the address family. This means that when address family is enabled, LDP needs to reset existing sessions with the peers in order to re-learn the discarded bindings. The implementation is optimized to reset only those sessions which were previously known to be dual-stack and had sent bindings for both address families.

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.

Figure 1: Control Plane Failure

This figure illustrates a control plane failure and shows the process and results of a control plane failure leading to loss of connectivity.



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.

Default Transport Address

LDP computes default local transport address for IPv6 from its IPv6 interface or address database by picking the lowest operational loopback interface with global unicast IPv6 address. This means that any change in this loopback state or address, flaps or changes the default transport address for IPv6 and may cause session flaps using such an address as transport endpoint. For example, if a session is currently active on Loopback2 as during it's inception it was the lowest loopback with an IPv6 address, and a lower loopback, Loopback0, is configured with an IPv6 address, the session does not flap. However, if it does flap, the next time the session is attempted, Loopback0 is used.

The session flaps when configuring discovery transport address explicitly.

Use the `discovery transport-address` command under the LDP address family submode to specify the global transport address for IPv4 or IPv6.

It is recommended to configure global transport-address for IPv6 address family to avoid a potentially unstable default transport address.

Label Distribution Protocol Discovery Parameters

Discovery parameter specifies the time periods between transmitted and not received hello messages.

Configuration Example

A discovery parameter specifies time of the discovered neighbor (15 seconds) which is kept without receipt of any subsequent hello messages. After the specified time period, there is an interval of 5 seconds between the transmission of consecutive hello messages.

Configuration of Label Distribution Protocol Discovery Parameters

```
Router(config)#mpls ldp
Router(config-ldp)#router-id 192.168.70.1
Router(config-ldp)#discovery hello holdtime 15
Router(config-ldp)#discovery targeted-hello holdtime 5
Router(config-ldp)#commit
```

Verification

Displays all the current MPLS LDP parameters.

```
RP/0/RP0/CPU0:router# show mpls ldp parameters
LDP Parameters:
Role: Active
Protocol Version: 1
Router ID: 192.168.70.1


Discovery:
Link Hellos:     Holdtime:15 sec, Interval:5 sec
Targeted Hellos: Holdtime:5 sec, Interval:10 sec
Quick-start: Enabled (by default)
Transport address:       IPv4: 192.168.70.1
```

Downstream on Demand

The 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.

Explicit-null and implicit-null labels

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Explicit-null and implicit-null labels	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Explicit-null and implicit-null labels	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Explicit-null and implicit-null labels	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
Explicit-null and implicit-null labels	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The MPLS Label Distribution Protocol (LDP) is integrated to streamline label distribution and management in MPLS networks. This enhancement optimizes label switching by establishing LDP sessions between routers to exchange label mapping information efficiently. LDP supports the creation of LSPs (Label Switched Paths), ensuring effective and scalable data forwarding across the network. By facilitating dynamic label assignment, it reduces manual configuration efforts and enhances network flexibility, contributing to more reliable and efficient traffic management. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

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.

Label distribution protocol interior gateway protocol synchronization

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Label distribution protocol interior gateway protocol synchronization	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Label distribution protocol interior gateway protocol synchronization	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Label distribution protocol interior gateway protocol synchronization	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Label distribution protocol interior gateway protocol synchronization	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC:K100])(select variants only*). The MPLS Label Distribution Protocol (LDP) enables the establishment of label-switched paths (LSPs) by mapping network routes to labels, facilitating efficient packet forwarding. This feature streamlines the process of routing by dynamically distributing labels to network devices, enhancing scalability and reducing configuration complexity. LDP is integral for optimizing network performance in MPLS environments, ensuring reliable and efficient data transmission. *Previously this feature was supported on Q200 and Q100. It is now extended to Cisco 8712-MOD-M routers.
Label distribution protocol interior gateway protocol synchronization	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 8212-48FH-M • 8711-32FH-M

Lack of synchronization between LDP and Interior Gateway Protocol (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 coordinates 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, 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 check-point recovered LDP graceful restart session is used and treated as converged and is given an opportunity to connect and resynchronize.

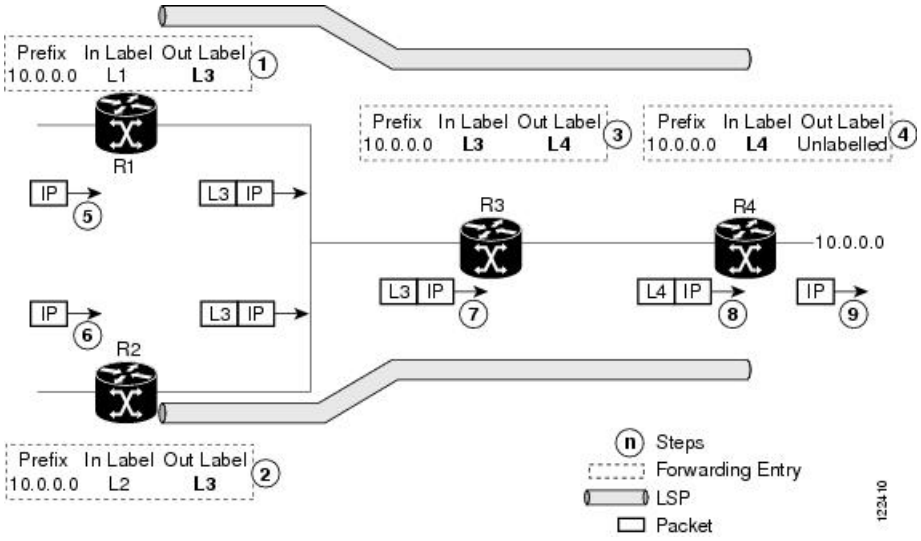
Under certain circumstances, it might be required to delay declaration of re-synchronization to a configurable interval. LDP provides a configuration option to delay declaring synchronization up for up to 60 seconds. LDP communicates this information to IGP upon linkup or session down events.

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.

Figure 2: Forwarding Setup

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



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.

Note For local labels, only up to 12000 rewrites are supported. If the rewrites exceed this limit, MPLS LSD or MPLS LDP or both the processes may crash.

Setting up Label Distribution Protocol Graceful Restart

Configuration Example

This example shows how to configure LDP graceful restart. In this example, the amount of time that a neighboring router maintains the forwarding state about the gracefully restarting router is specified as 180 seconds. Also, the amount of time the LDP neighbor should wait for a reconnection from the gracefully restarting router in the event of a LDP session failure is specified as 169 seconds.

```
Router(config)#mpls ldp
Router(config-ldp)#interface TenGigE 0/0/0/5
Router(config-ldp-if)#exit
Router(config-ldp)#graceful-restart
Router(config-ldp)#graceful-restart forwarding-state-holdtime 180
Router(config-ldp)#graceful-restart reconnect-timeout 169
Router(config-ldp)#commit
```

Verification

```
RP/0/RP0/CPU0:router#show mpls ldp graceful-restart
Forwarding State Hold timer : Not Running
GR Neighbors : 1

Neighbor ID      Up  Connect Count  Liveness Timer      Recovery Timer
---------------  --  -------------  ------------------  ------------------
8.8.8.8          Y        1               -                   -
```

```
RP/0/RP0/CPU0:router#show mpls ldp parameters
Graceful Restart:Enabled
Reconnect Timeout:169 sec, Forwarding State Holdtime:180 sec
NSR: Disabled, Not Sync-ed
```

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.

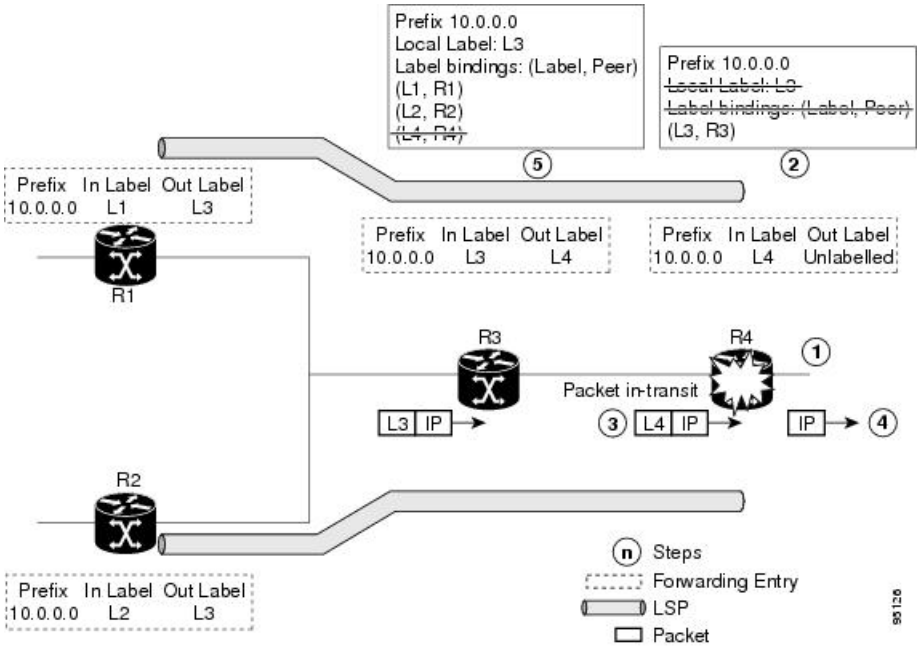
Control state recovery

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.

Recovery with Graceful-Restart

Figure 3: Recovering with Graceful Restart

This figure illustrates the process of failure recovery using 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.

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 *IP Addresses and Services Configuration Guide for Cisco 8000 Series Routers*.

Local Label Allocation Control

Label Distribution Protocol allocates local labels for all prefixes that are not Border Gateway Protocol (BGP) prefixes[1]. 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

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

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.

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

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**redistribute bgp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# redistribute bgp advertise-to acl_1`	Allows the redistribution of BGP routes into an MPLS LDP processes. **Note** Autonomous system numbers (ASNs) are gl (ASs) and enable ASs to exchange exterior is allocated to each AS for use in BGP routi BGP.
Step 4	**end** or **commit**	• When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 5	**show run mpls ldp** **Example:** `RP/0/RP0/CPU0:router# show run mpls ldp`	Displays information about the redistributed route information.

Session protection

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Session protection	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Session protection	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*)(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D

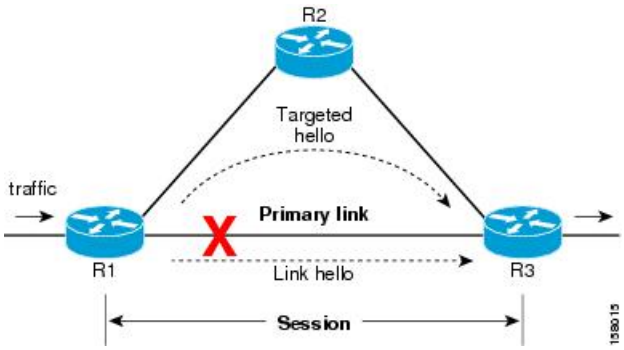
Session protection	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Session protection	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) Improve network robustness with MPLS LDP session protection, which safeguards Label Distribution Protocol sessions from interruptions. This feature maintains session continuity during network instability, reducing the risk of label distribution loss. It ensures stable MPLS operations by quickly recovering LDP sessions after disruptions, thereby enhancing overall network reliability and performance. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

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.

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

Figure 4: Session protection



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

LDP over RSVP LSR support

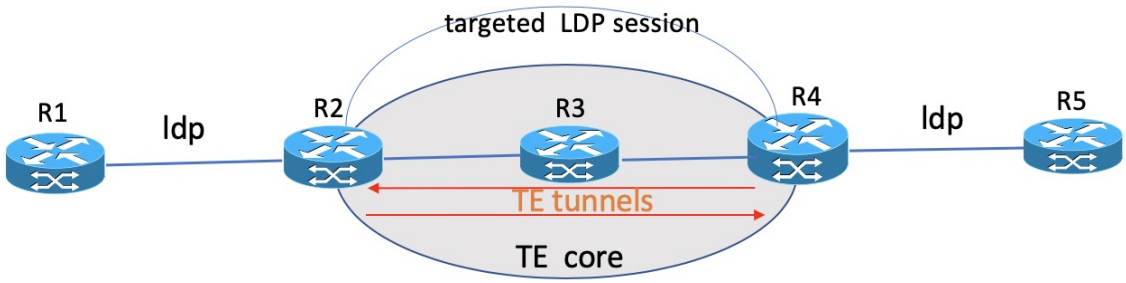
Table 7: Feature History Table

Feature Name	Release Information	Feature Description
LDP over RSVP LSR support	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
LDP over RSVP LSR support	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
LDP over RSVP LSR support	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
LDP over RSVP LSR support	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
LDP over RSVP LSR support	Release 7.3.1	With this feature, users can transport LDP traffic over an RSVP-TE network automatically, through a targeted LDP session. The automatic configuration for LDP over RSVP-TE supports 1000 TE tunnels.

Consider this topology of an RSVP-TE network spanning R2 to R4. LDP traffic is transported from R1 to R5. A targeted LDP session is established between R2 and R4 so that LDP traffic is transported over the TE tunnel network.

Figure 5: LDP Over RSVP



No additional configuration is required to enable the LDP over RSVP-TE function. Up to 1000 tunnels are supported by default. The **autoroute announce** command is enabled on the edge routers of the RSVP-TE network.

If you need more than 1000 TE tunnels, enable the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command on the edge routers R2 and R4. However, when you enable this command, the LDP over TE feature gets disabled.

The following configuration disables the LDP over TE function, and allows you run more than 1000 TE tunnels.

```
Router# configure terminal
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
```

```
Router(config)# commit
Router# reload
```

Increase in RSVP-TE Tunnel Scale for LDP over TE

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Increase in RSVP-TE Tunnel Scale for LDP over TE	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Increase in RSVP-TE Tunnel Scale for LDP over TE	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M
Increase in RSVP-TE Tunnel Scale for LDP over TE	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Increase in RSVP-TE Tunnel Scale for LDP over TE	Release 7.5.3	We allow you to use a maximum of 4000 RSVP-TE tunnels with LDP over TE using the **hw-module profile cef te-tunnel highscale-ldp-over-te-no-sr-over-srte** command. Earlier, only 1000 RSVP-TE tunnels were supported. If you further need to increase the number of RSVP-TE tunnels up to 8000, enable the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command on the edge routers. However, the LDP over TE is disabled when you enable this command.

You can scale up to 4000 RSVP-TE tunnels only on routers with line cards based on Q200 Silicon. In this scenario, LDP over TE feature is enabled which means you have the ability to steer LDP prefixes or labeled SR prefixes over a higher number of RSVP-TE tunnels.

To enable up to a maximum of 4000 RSVP-TE tunnels with LDP over TE feature, use the **hw-module profile cef te-tunnel highscale-ldp-over-te-no-sr-over-srte** command on the edge routers.

```
Router# configure
Router(config)# hw-module profile cef te-tunnel highscale-ldp-over-te-no-sr-over-srte
Router(config)# commit
Router# reload
```

In scenarios where you require additional tunnels without requiring to steer LDP prefixes or labeled SR prefixes over RSVP-TE tunnels, use the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command on the edge routers to configure up to a maximum of 8000 TE tunnels.

```
Router# configure terminal
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
Router(config)# commit
Router# reload
```

Note Reload the line card for changes to take effect.

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 :

Implementing MPLS Label Distribution Protocol

MPLS (Multi Protocol Label Switching) is a forwarding mechanism based on label switching. In an MPLS network, data packets are assigned labels and packet-forwarding decisions are taken based on the contents of the label. To switch labeled packets across the MPLS network, predetermined paths are established for various source-destination pairs. These predetermined paths are known as Label Switched Paths (LSPs). To establish LSPs, MPLS signaling protocols are used. Label Distribution Protocol (LDP) is an MPLS signaling protocol used for establishing LSPs. This module provides information about how to configure MPLS LDP.

MPLS Restrictions

Labeled packets are not supported on Linux interfaces.

Enabling MLDP

Perform this task to enable Multicast Label Distribution Protocol (MLDP) in MPLS LDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**

3. **mldp**
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `Router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.

Step 3	**mldp**
	Example:
	`Router(config-ldp)# mldp`
	`Router(config-ldp-mldp)#`
		Enables MLDP.

Step 4	**end** or **commit**
	Example:
	`Router(config-ldp-mldp)# end`
	or
	`Router(config-ldp-mldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Make-Before-Break

Perform this task to enable the make-before-break (MBB) feature in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **make-before-break** [**delay** *seconds*]
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**make-before-break** [**delay** *seconds*] **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# make-before-break delay 10`	Enables the make-before-break feature. (Optional) Configures the MBB forwarding delay in seconds. Range is 0 to 600.

	Command or Action	Purpose

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP MoFRR

Perform this task to enable multicast only fast reroute (MoFRR) support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **mofrr**
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

	Command or Action	Purpose
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**mofrr** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# mofrr`	Enables MoFRR support.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Recursive FEC

Perform this task to enable recursive forwarding equivalence class (FEC) support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **recursive-fec**
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**recursive-fec** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# recursive-fec`	Enables recursive FEC support.

	Command or Action	Purpose

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Static Multipoint to Multipoint LSP

Perform this task to enable static multipoint to multipoint (MP2MP) LSP support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **static mp2mp** *ip-address*
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

	Command or Action	Purpose
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**static mp2mp** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# static mp2mp 10.10.10.10 1`	Enables static MP2MP LSP support and specifies MP2MP LSP root IP address followed by the number of LSPs in the range 1 to 1000.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling MLDP Static Point to Multipoint LSP

Perform this task to enable static point to multipoint (P2MP) LSP support in MPLS MLDP.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **mldp**
4. **address-family ipv4**
5. **static p2mp** *ip-address*
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**mldp** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# mldp`	Enables MLDP.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4`	Enables MLDP for IPv4 address family.
Step 5	**static p2mp** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-ldp-mldp-af)# static p2mp 10.0.0.1 1`	Enables static P2MP LSP support and specifies P2MP LSP root IP address followed by the number of LSPs in the range 1 to 1000.

	Command or Action	Purpose

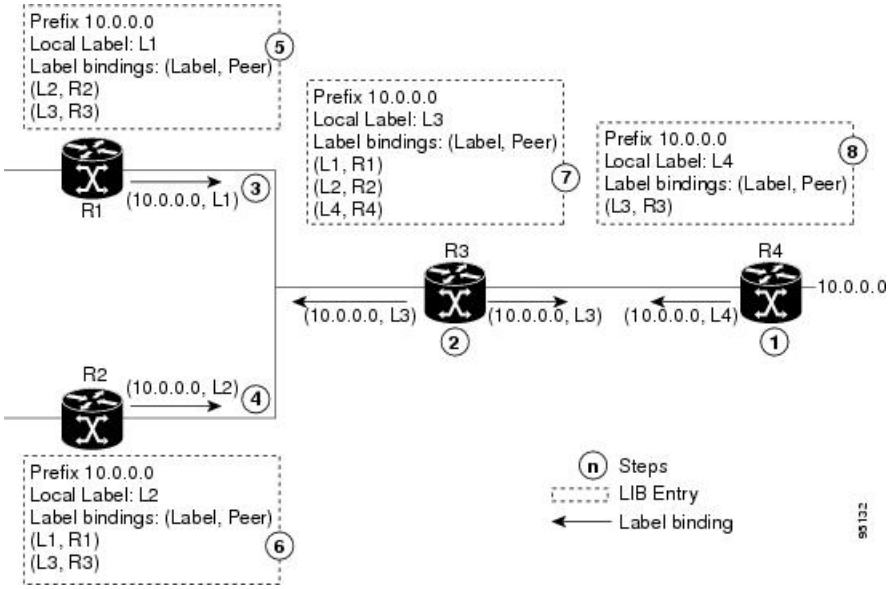
Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-mldp-af)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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.

Figure 6: Setting Up Label Switched Paths

This figure illustrates the process of label binding exchange for setting up LSPs.



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.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:**	Enters global configuration mode.

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

Step 3	**label advertise** { **disable** | **for** *prefix-acl* [**to** *peer-acl*] | **interface** *type interface-path-id* }
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# label advertise interface HundredGigE 0/0/0/2`
	`RP/0/RP0/CPU0:router(config-ldp)# for pfx_acl1 to peer_acl1`
		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.
		for ***prefix-acl***
		to ***peer-acl***
		Specifies neighbors to advertise and receive label advertisements.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**label** **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# label`	Configures the allocation, advertisement ,and acceptance of labels.
Step 4	**implicit-null-override for** *access-list* **Example:** `RP/0/RP0/CPU0:router(config-ldp-af-lbl)# implicit-null-override for 70`	Configures implicit-null local label for non-egress prefixes. **Note** This feature works with any prefix including
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Setting Up LDP Forwarding

Perform this task to set up LDP forwarding.

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

Before you begin

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

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **explicit-null**
4. Use the **commit** or **end** command.
5. (Optional) **show mpls ldp forwarding**
6. (Optional) **show mpls forwarding**
7. (Optional) **ping** *ip-address*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**explicit-null** **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# explicit-null`	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).
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

	Command or Action	Purpose
Step 5	(Optional) **show mpls ldp forwarding** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp forwarding`	Displays the MPLS LDP view of installed forwarding states (rewrites). **Note** For local labels, only up to 12000 rewrites a MPLS LDP or both the processes may crash
Step 6	(Optional) **show mpls forwarding** **Example:** `RP/0/RP0/CPU0:router# show mpls forwarding`	Displays a global view of all MPLS installed forwarding states (rewrites) by various applications (LDP, TE, and static).
Step 7	(Optional) **ping** *ip-address* **Example:** `RP/0/RP0/CPU0:router# ping 192.168.2.55`	Checks for connectivity to a particular IP address (going through MPLS LSP as shown in the **show mpls forwarding** command).

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. Use the **commit** or **end** command.
10. (Optional) **show mpls ldp neighbor**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`	Enters interface configuration mode for the LDP protocol.
Step 4	**discovery transport-address** [*ip-address* | **interface**] **Example:** or `RP/0/RP0/CPU0:router(config-ldp-if-af)# discovery transport-address interface`	Provides an alternative transport address for a TCP connection. • Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID. • Transport address configuration is applied for a given LDP-enabled interface. • If the interface version of the command is used, the configured IP address of the interface is passed to its neighbors as the transport address.
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.
Step 6	**holdtime** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# holdtime 30`	Changes the time for which an LDP session is maintained in the absence of LDP messages from the peer. • Outgoing keepalive interval is adjusted accordingly (to make three keepalives in a given holdtime) with a change in session holdtime value. • Session holdtime is also exchanged when the session is established. • In this example holdtime is set to 30 seconds, which causes the peer session to timeout in 30 seconds, as

	Command or Action	Purpose
		well as transmitting outgoing keepalive messages toward the peer every 10 seconds.
Step 7	**neighbor** *ip-address* **password** [*encryption*] *password* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# neighbor 192.168.2.44 password secretpasswd`	Configures password authentication (using the TCP MD5 option) for a given neighbor.
Step 8	**backoff** *initial maximum* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# backoff 10 20`	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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 10	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp neighbor`	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.

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. Use the **commit** or **end** command.
9. (Optional) **show mpls ldp parameters**
10. (Optional) **show mpls ldp neighbor**
11. (Optional) **show mpls ldp graceful-restart**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

Step 3	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`
	`RP/0/RP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol.

Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# exit`	Exits the current configuration mode.
Step 5	**graceful-restart** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# graceful-restart`	Enables the LDP graceful restart feature.

	Command or Action	Purpose
Step 6	**graceful-restart forwarding-state-holdtime** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# graceful-restart forwarding-state-holdtime 180`	Specifies the length of time that forwarding can keep LDP-installed forwarding states and rewrites, and specifies wh en the LDP control plane restarts. • 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. • Recovery time sent after restart is computed as the current remaining value of the forwarding state hold timer.
Step 7	**graceful-restart reconnect-timeout** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# graceful-restart reconnect-timeout 169`	Specifies the length of time a neighbor waits before restarting the node to reconnect before declaring an earlier graceful restart session as down. This command is used to start a timer on the peer (upon a neighbor restart). This timer is referred to as *Neighbor Liveness* timer.
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 9	(Optional) **show mpls ldp parameters** **Example:** `RP/0/RP0/CPU0:router # show mpls ldp parameters`	Displays all the current MPLS LDP parameters.
Step 10	(Optional) **show mpls ldp neighbor** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp neighbor`	Displays the status of the LDP session with its neighbors. This command can be run with various filters as well as with the brief option.
Step 11	(Optional) **show mpls ldp graceful-restart** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp graceful-restart`	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.

Configuring Label Acceptance Control (Inbound Filtering)

Perform this task to configure LDP inbound label filtering.

Note 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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.

Step 3	**label accept for** *prefix-acl* **from** *ip-address*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# label accept for pfx_acl_1 from 192.168.1.1`
	`RP/0/RP0/CPU0:router(config-ldp)# label accept for pfx_acl_2 from 192.168.2.2`
		Configures inbound label acceptance for prefixes specified by prefix-acl from neighbor (as specified by its IP address).

Step 4	[**vrf** *vrf-name*] **address-family** { **ipv4**} **Example:** `RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4`	(Optional) Specifies a non-default VRF. Enables the LDP IPv4 or IPv6 address family.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# address-family ipv6`	
Step 5	**label remote accept from** *ldp-id* **for** *prefix-acl* **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# label remote accept from 192.168.1.1:0 for pfx_acl_1`	Configures inbound label acceptance control for prefixes specified by prefix-acl from neighbor (as specified by its LDP ID).
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**router isis** *instance-id*
	Example:
	`RP/0/RP0/CPU0:router(config)# router isis 100`
	`RP/0/RP0/CPU0:router(config-isis)#`
		Enables the Intermediate System-to-Intermediate System (IS-IS) routing protocol and defines an IS-IS instance.

Step 3	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-isis)# interface HundredGigE 0/0/0/2`
	`RP/0/RP0/CPU0:router(config-isis-if)#`
		Configures the IS-IS protocol on an interface and enters ISIS interface configuration mode.

Step 4	**address-family** {**ipv4** } **unicast**
	Example:
	`RP/0/RP0/CPU0:router(config-isis-if)# address-family ipv4 unicast`
	`RP/0/RP0/CPU0:router(config-isis-if-af)#`
		Enters address family configuration mode for configuring IS-IS routing for a standard IP version 4 (IPv4) address prefix.

Step 5	**mpls ldp sync** **Example:** `RP/0/RP0/CPU0:router(config-isis-if-af)# mpls ldp sync`	Enables LDP IGP synchronization.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configure Label Distribution Protocol Targeted Neighbor

LDP session between LSRs that are not directly connected is known as targeted LDP session. For LDP neighbors which are not directly connected, you must manually configure the LDP neighborship on both the routers.

Configuration Example

This example shows how to configure LDP for non-directly connected routers.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# mpls ldp
RP/0/RSP0/CPU0:router(config-ldp)# router-id 192.0.2.1
RP/0/RSP0/CPU0:router(config-ldp)# neighbor 198.51.100.1:0 password encrypted 13061E010803
RP/0/RSP0/CPU0:router(config-ldp)# address-family ipv4
RP/0/RSP0/CPU0:router(config-ldp-af)# discovery targeted-hello accept
RP/0/RSP0/CPU0:router(config-ldp-af)# neighbor 198.51.100.1 targeted
RP/0/RSP0/CPU0:router(config-ldp-af)# commit
```

Running Configuration

This section shows the LDP targeted neighbor running configuration.

```
mpls ldp
router-id 192.0.2.1
neighbor 198.51.100.1:0 password encrypted 13061E010803
address-family ipv4
  discovery targeted-hello accept
  neighbor 198.51.100.1 targeted
!
```

Verification

Verify LDP targeted neighbor configuration.

```
RP/0/RSP0/CPU0:router#show mpls ldp discovery
Wed Nov 28 04:30:31.862 UTC

Local LDP Identifier: 192.0.2.1:0
Discovery Sources:
  Targeted Hellos:  <<< targeted hellos based session
    192.0.2.1 -> 198.51.100.1(active/passive), xmit/recv   <<< both transmit and receive
of targeted hellos between the neighbors
      LDP Id: 198.51.100.1:0
          Hold time: 90 sec (local:90 sec, peer:90 sec)
          Established: Nov 28 04:19:55.340 (00:10:36 ago)

RP/0/RSP0/CPU0:router#show mpls ldp neigbhor
Wed Nov 28 04:30:38.272 UTC

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:0:13183 - 192.0.2.1:646; MD5 on
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 20/20; Downstream-Unsolicited
  Up time: 00:10:30
  LDP Discovery Sources:
    IPv4: (1)
```

```
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active/passive) <<< targeted LDP based
session
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (4)
     198.51.100.1        10.0.0.1       172.16.0.1       192.168.0.1
    IPv6: (0)
```

Configuring Global Transport Address

Perform this task to configure global transport address for the IPv4 address family.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **address-family ipv4**
4. **discovery transport-address** *ip-address*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4`	Enables LDP IPv4 address family.
Step 4	**discovery transport-address** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-ldp-af)# discovery transport-address 192.168.1.42`	Provides an alternative transport address for a TCP connection. • Default transport address advertised by an LSR (for TCP connections) to its peer is the router ID.
Step 5	**end** or **commit**	• When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	Example:
	`RP/0/RP0/CPU0:router (config-ldp-af)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp-af)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv4 as Transport Preference

Perform this task to configure IPv4 as the preferred transport (overriding the default setting of IPv6 as preferred transport) to establish connection for a set of dual-stack peers.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **neighbor dual-stack transport-connection prefer ipv4 for-peers** *peer lsr-id*
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:**	Enters MPLS LDP configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# mpls ldp`	
Step 3	**neighbor dual-stack transport-connection prefer ipv4 for-peers** *peer lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# neighbor dual-stack transport-connection prefer ipv4 for-peers 5.5.5.5`	Configures IPv4 as the preferred transport connection for the specified peer.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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. Use the **commit** or **end** command.
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

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. In Cisco IOS XR software, the router ID is specified as an interface name or IP address or LSR ID. By default, LDP uses the global router ID (configured by global router ID process).
Step 4	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`	Enters interface configuration mode for the LDP protocol.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session.

	Command or Action	Purpose
		end —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery`	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	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

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. Use the **commit** or **end** command.
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

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. • In Cisco IOS XR software, the router ID is specified as an interface IP address or LSR ID. By default, LDP

	Command or Action	Purpose
		uses the global router ID (configured by global router ID process).
Step 4	**discovery targeted-hello accept** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# discovery targeted-hello accept`	Directs the system to accept targeted hello messages from any source and activates passive mode on the LSR for targeted hello acceptance. • This command is executed on the receiver node (with respect to a given MPLS TE tunnel). • You can control the targeted-hello acceptance using the **discovery targeted-hello accept** command.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery`	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	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.

	Command or Action	Purpose
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

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. Use the **commit** or **end** command.
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

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	[**vrf** *vrf-name*] **router-id** *ip-address lsr-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# router-id 192.168.70.1`	(Optional) Specifies a non-default VRF. Specifies the router ID of the local node. • 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).

Step 4	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ldp)# interface tunnel-te 12001`
	`RP/0/RP0/CPU0:router(config-ldp-if)#`
		Enters interface configuration mode for the LDP protocol. Interface type must be Tunnel-TE.

Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) **show mpls ldp discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery`	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.

	Command or Action	Purpose
Step 7	(Optional) **show mpls ldp vrf** *vrf-name* **discovery** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf red discovery`	Displays the status of the LDP discovery process for the specified VRF.
Step 8	(Optional) **show mpls ldp vrf all discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs.
Step 9	(Optional) **show mpls ldp vrf all discovery brief** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all discovery brief`	Displays the brief status of the LDP discovery process for all VRFs.
Step 10	(Optional) **show mpls ldp vrf all ipv4 discovery summary** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp vrf all ipv4 discovery summary`	Displays the summarized status of the LDP discovery process for all VRFs for the IPv4 address family.
Step 11	(Optional) **show mpls ldp discovery summary all** **Example:** `RP/0/RP0/CPU0:router# show mpls ldp discovery summary all`	Displays the aggregate summary across all the LDP discovery processes.

Configuring Downstream on Demand

By default, LDP uses downstream unsolicited mode in which label advertisements for all routes are received from all LDP peers. The 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.

In downstream on demand configuration, an ACL is used to specify the set of peers for downstream on demand mode. For down stream on demand to be enabled, it needs 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.

Configuration Example

This example shows how to configure LDP Downstream on Demand.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# session downstream-on-demand with ACL1
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Configuring LDP Link: Example

The example shows how to configure LDP link parameters.

```
mpls ldp
 interface HundredGigE 0/0/0/2
 !
!

show mpls ldp discovery
```

Configuring Label Distribution Protocol Nonstop Routing

LDP nonstop routing (NSR) functionality makes failures, such as Route Processor (RP) or Distributed Route Processor (DRP) fail over, 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
- 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. L2VPN configuration is not supported on NSR. Process failures of active LDP results in session loss and, as a result, NSR cannot be provided unless RP switchover is configured as a recovery action.

Configuration Example

This example shows how to configure LDP Non-Stop Routing.

```
RP/0/RP0/CPU0:Router(config)# mpls ldp
RP/0/RP0/CPU0:Router(config-ldp)# nsr
RP/0/RP0/CPU0:Router(config-ldp)# commit
```

Verification

```
RP/0/RP0/CPU0:Router# show mpls ldp nsr summary
Mon Dec  7 04:02:16.259 UTC
Sessions:
Total: 1, NSR-eligible: 1, Sync-ed: 0
(1 Ready)
```

Configure Session Protection

Configuration Example

As per the configuration, LDP session protection for peers specified by peer-acl is configured to maximum duration of 60 seconds.

```
Router(config)#mpls ldp
Router(config-ldp)#session protection for peer_acl_1 duration 60
Router(config-ldp)#commit
```

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters the MPLS LDP configuration mode.
Step 3	**label allocate for** *prefix-acl* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# label allocate for pfx_acl_1`	Configures label allocation control for prefixes as specified by prefix-acl.

	Command or Action	Purpose
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring Transport Preference Maximum Wait Time

Perform this task to configure the maximum time (in seconds) the preferred address family connection must wait to establish transport connection before resorting to non-preferred address family.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **neighbor dual-stack transport-connection max-wait** *seconds*
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**neighbor dual-stack transport-connection max-wait** *seconds* **Example:**	Configures the maximum wait time.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ldp)# neighbor dual-stack transport-connection max-wait 5`	

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring MPLS Label Security

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

Configuration Example

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# mpls label-security rpf
Router(config-if)# commit
```

Verification

Use the **show mpls forwarding label-security interface** command to view MPLS label security configuration on an interface.

Disabling Implicit IPv4

Perform this task to disable the implicitly enabled IPv4 address family for default VRF.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**

3. **default-vrf implicit-ipv4 disable**
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.
Step 3	**default-vrf implicit-ipv4 disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp)# default-vrf implicit-ipv4 disable`	Disables the implicitly enabled IPv4 address family for default VRF.

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router (config-ldp)# end`
	or
	`RP/0/RP0/CPU0:router (config-ldp)# commit`
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

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

Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`	Enters interface configuration mode and configures an interface.
Step 4	**igp auto-config disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# igp auto-config disable`	Disables auto-configuration on the specified interface.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		• **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Disabling LDP IGP Synchronization: OSPF

Perform this task to disable LDP IGP Synchronization under OSPF.

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 109`	Identifies the OSPF routing process and enters OSPF configuration mode.
Step 3	Use one of the following commands: • **area** *area-id* **mpls ldp sync disable** • **area** *area-id* **interface** *name* **mpls ldp sync disable** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 1 mpls ldp sync disable`	Disables LDP IGP synchronization on an interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ospf)# area 1 interface HundredGigE 0/0/0/2 mpls ldp sync disable`	
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Disabling MLDP

Perform this task to disable MLDP on Label Distribution Protocol (LDP) enabled interfaces.

SUMMARY STEPS

1. **configure**
2. **mpls ldp**
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** }
5. **igp mldp disable**
6. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**mpls ldp** **Example:** `RP/0/RP0/CPU0:router(config)# mpls ldp`	Enters MPLS LDP configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ldp)# interface HundredGigE 0/0/0/2`	Enters interface configuration mode for the LDP protocol.
Step 4	**address-family** {**ipv4** } **Example:** `RP/0/RP0/CPU0:router(config-ldp-if)# address-family ipv4` or	Enables the LDP IPv4 address family.
Step 5	**igp mldp disable** **Example:** `RP/0/RP0/CPU0:router(config-ldp-if-af)# igp mldp disable`	Disables MLDP.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Auto-enable LDP on all TE tunnel interfaces

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
Auto-enable LDP on all TE tunnel interfaces	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.

Feature Name	Release Information	Feature Description
Auto-enable LDP on all TE tunnel interfaces	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Auto-enable LDP on all TE tunnel interfaces	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Auto-enable LDP on all TE tunnel interfaces	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Auto-enable LDP on all TE tunnel interfaces	Release 7.11.1	You can now automatically enable all the configured LDP tunnels over IPv4 traffic-engineering (TE) with upto 2000 unique destinations to achieve time efficiency and consistency of LDP tunnel configurations in routers within an MPLS network. Earlier, you could enable the LDP tunnels over TE manually. This feature introduces the following changes: **CLI**: mpls ldp address-family ipv4 traffic-eng tunnels command. **YANG Data Model**: Cisco-IOS-XR-mpls-ldp-cfg (see GitHub, YANG Data Models Navigator)

Label Distribution Protocol over Traffic Engineering (LDPoTE) enables the establishment of label-switched paths (LSPs) in an MPLS network with specific traffic engineering requirements, such as link bandwidth, quality of service (QoS), and path optimization. LDPoTE enhances the scalability and efficiency of MPLS networks. It provides the ability to control traffic paths, allocate network resources, and optimize network performance based on traffic engineering policies.

From Release 7.11.1, you can automatically enable (auto-enable) all the configured tunnels over IPv4 TE at once upto 2000 unique destination routers using the **mpls ldp address-family ipv4 traffic-eng tunnels** command.

You can auto-enable:

- Named tunnels—Tunnels that are identified by specific identifiers or names. For example: ldp_tunnel
- Numbered tunnels—Tunnels that are identified by numerical values. For example: tunnel0
- Using a regular expression—Tunnels that are matched with the regular expression. For example: .*-1.

Limitations

Auto-enable LDP on all TE tunnel interfaces doesn't support:

- IPv6 LDP tunnel configuration
- Automatically establishing the LDP tunnels in a mesh network
- Auto-configuring of cloned tunnels

Configure Auto-enable LDP on all TE Tunnel Interfaces

Auto-enable LDP on all TE tunnel interfaces can be done for:

- All the named and numbered tunnels.
- All the tunnels which are matched with the regular expression.

Configuration Example for Auto-enabling LDP on all Named and Numbered TE Tunnel Interfaces

```
Router# configure
Router(config)# mpls ldp address-family ipv4 traffic-eng tunnels all
Router(config)# commit
Router(config)# end
```

Running Configuration

```
mpls ldp
 address-family ipv4
  traffic-eng
   tunnels all
  !
 !
```

Verification

Use the **show mpls ldp interface** command to verify the auto-enabled tunnels.

```
Router# show mpls ldp interface
Interface interface tunnel-te 1 (0xf0)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All
Interface interface tunnel-te 2 (0x110)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All
Interface interface tunnel-te 3 (0x210)
```

```
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All
```

Configuration Example for Auto-enabling LDP on TE Tunnel Interfaces With Regular Expression

```
Router# configure
Router(config)# mpls ldp address-family ipv4 traffic-eng tunnels regular-expression .*-1
Router(config)# end
```

Running Configuration

```
mpls ldp
 address-family ipv4
  traffic-eng
   tunnels regular-expression .*-1
  !
 !
```

Verification

```
Router# show mpls ldp interface
Interface cisco-1 (0x110)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All with regex '.*-1'
Interface tunnel-te1-1 (0x210)
    VRF:  'Default' (0x6000000)
    Enabled Via config: TE Auto-config All with regex '.*-1'
```

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# router ospf 190`
	`RP/0/RP0/CPU0:router(config-ospf)#`
		Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

Step 3	**mpls ldp auto-config** **Example:** `RP/0/RP0/CPU0:router(config-ospf)# mpls ldp auto-config`	Enables LDP auto-configuration.
Step 4	**area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 8`	Configures an OSPF area and identifier. ***area-id*** Either a decimal value or an IP address.
Step 5	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface HundredGigE 0/0/0/2`	Enables LDP auto-configuration on the specified interface. **Note** LDP configurable limit for maximum num
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Related Topics

Disabling LDP Auto-Configuration, on page 68

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**router ospf** *process-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# router ospf 100`
	`RP/0/RP0/CPU0:router(config-ospf)#`
		Enters a uniquely identifiable OSPF routing process. The process name is any alphanumeric string no longer than 40 characters without spaces.

Step 3	**area** *area-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ospf)# area 8`
	`RP/0/RP0/CPU0:router(config-ospf-ar)#`
		Configures an OSPF area and identifier.
		area-id
		Either a decimal value or an IP address.

Step 4	**mpls ldp auto-config** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# mpls ldp auto-config`	Enables LDP auto-configuration.

	Command or Action	Purpose

Step 5	**interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-ospf-ar)# interface HundredGigE 0/0/0/2`
	`RP/0/RP0/CPU0:router(config-ospf-ar-if)`
		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.

Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Related Topics

Disabling LDP Auto-Configuration, on page 68

Implicit IPv4 Disable

The LDP configuration model was changed with the introduction of explicit address family enabling under LDP (VRF) global and LDP (VRF) interfaces. However, in order to support backward compatibility, the old configuration model was still supported for default VRF. There was, however, no option to disable the implicitly enabled IPv4 address family under default VRF's global or interface level.

A new configuration **mpls ldp default-vrf implicit-ipv4 disable** is now available to the user to disable the implicitly enabled IPv4 address family for the default VRF. The new configuration provides a step towards migration to new configuration model for the default VRF that mandates enabling address family explicitly. This means that if the new option is configured, the user has to explicitly enable IPv4 address family for default VRF global and interface levels. It is recommended to migrate to this explicitly enabled IPv4 configuration model.

For detailed configuration steps, see Disabling Implicit IPv4, on page 66

Running Configuration

This section shows the LDP targeted neighbor running configuration.

```
mpls ldp
router-id 192.0.2.1
neighbor 198.51.100.1:0 password encrypted 13061E010803
address-family ipv4
  discovery targeted-hello accept
  neighbor 198.51.100.1 targeted
!
```

Verification

Verify LDP targeted neighbor configuration.

```
RP/0/RSP0/CPU0:router#show mpls ldp discovery
Wed Nov 28 04:30:31.862 UTC

Local LDP Identifier: 192.0.2.1:0
Discovery Sources:
  Targeted Hellos:  <<< targeted hellos based session
    192.0.2.1 -> 198.51.100.1(active/passive), xmit/recv   <<< both transmit and receive
of targeted hellos between the neighbors
      LDP Id: 198.51.100.1:0
          Hold time: 90 sec (local:90 sec, peer:90 sec)
          Established: Nov 28 04:19:55.340 (00:10:36 ago)

RP/0/RSP0/CPU0:router#show mpls ldp neigbhor
Wed Nov 28 04:30:38.272 UTC

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:0:13183 - 192.0.2.1:646; MD5 on
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 20/20; Downstream-Unsolicited
  Up time: 00:10:30
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active/passive) <<< targeted LDP based
session
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (4)
     198.51.100.1        10.0.0.1       172.16.0.1       192.168.0.1
    IPv6: (0)
```

Verify IP LDP Fast Reroute Loop Free Alternate: Example

Configuration Examples for Implementing MPLS LDP

These configuration examples are provided to implement LDP:

Configuring LDP Discovery for Targeted Hellos: Example

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

Active (tunnel head)

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

Passive (tunnel tail)

```
mpls ldp
   router-id 192.168.70.2
```

```
   discovery targeted-hello accept
  !
```

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 HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path HundredGigE 0/0/0/0 will exclude the interface
   # HundredGigE 0/0/0/3 in LFA backup path computation.
   fast-reroute per-prefix exclude interface HundredGigE 0/0/0/3
!
 interface HundredGigE 0/0/0/1
  point-to-point
  address-family ipv4 unicast
!
 interface HundredGigE 0/0/0/2
  point-to-point
  address-family ipv4 unicast
!
 interface HundredGigE 0/0/0/3
  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 HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
   # primary path HundredGigE 0/0/0/0 will exclude the interface
   # HundredGigE 0/0/0/3 in LFA backup path computation. TE tunnel 1001
   # is using the link HundredGigE 0/0/0/3.
   fast-reroute per-prefix exclude interface HundredGigE 0/0/0/3
   fast-reroute per-prefix lfa-candidate interface tunnel-te1001
!
 interface HundredGigE 0/0/0/3
  point-to-point
  address-family ipv4 unicast
!
```

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

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

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

Sample configuration:

router isis TEST
 net 49.0001.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  fast-reroute per-prefix tiebreaker downstream index 60
  fast-reroute per-prefix tiebreaker lc-disjoint index 10
  fast-reroute per-prefix tiebreaker lowest-backup-metric index 40
  fast-reroute per-prefix tiebreaker node-protecting index 30
  fast-reroute per-prefix tiebreaker primary-path index 20
  fast-reroute per-prefix tiebreaker secondary-path index 50
!
interface HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface HundredGigE 0/0/0/0
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
!
 interface HundredGigE 0/0/0/1.1
  point-to-point
  address-family ipv4 unicast
!
 interface HundredGigE 0/0/0/1.2
  point-to-point
  address-family ipv4 unicast
```

Configuring Local Label Allocation Control: Example

The example shows how to configure local label allocation control.

```
  mpls ldp
   label
  allocate for pfx_acl_1
   !
  !
```

Configuring LDP with Graceful Restart: Example

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

interface.

```
mpls ldp
 graceful-restart
 interface HundredGigE 0/0/0/2
!
```

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
```

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
 graceful-restart forwarding state-holdtime 180
 graceful-restart reconnect-timeout 15
 interface HundredGigE 0/0/0/2
!

show mpls ldp graceful-restart
show mpls ldp neighbor gr
show mpls ldp forwarding
show mpls forwarding
```

Configuring Label Acceptance (Inbound Filtering): Example

The example shows how to configure inbound label filtering.

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

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

Configuring LDP Discovery: Example

The example shows how to configure LDP discovery parameters.

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

show mpls ldp parameters
show mpls ldp discovery
```

Configuring LDP 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 HundredGigE 0/0/0/2
```

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 HundredGigE 0/0/0/2
```

Configuring LDP Neighbors: Example

The example shows how to disable label advertisement.

```
mpls ldp
    router-id 192.168.70.1
    neighbor 10.0.0.1 password encrypted 110A1016141E
    neighbor 172.16.0.1 implicit-withdraw
!
```

Configuring LDP IGP Synchronization—ISIS: Example

The example shows how to configure LDP IGP synchronization.

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

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
!
```

Label Distribution Protocol Interior Gateway Protocol Auto-configuration

Interior Gateway Protocol (IGP) auto-configuration allows you to 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.

Controlling State Advertisements in an mLDP-Only Setup

Table 10: Feature History Table

Feature Name	Release Information	Description
Controlling State Advertisements in an mLDP-Only Setup	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D

Feature Name	Release Information	Description
Controlling State Advertisements in an mLDP-Only Setup	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M
Controlling State Advertisements in an mLDP-Only Setup	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Controlling State Advertisements in an mLDP-Only Setup	Release 7.5.2	In conformance with RFC 7473, you can control state advertisements of non-negotiated Label Distribution Protocol (LDP) applications in a Multipoint LDP (mLDP)-only environment. In such an environment, participating routers don't need to exchange any unicast binding information. As a result, the flow of LDP state information in an mLDP-only setup is faster. Also, when routers come up after a network event, the network convergence time is shorter.

This function explains controlling of state advertisements of non-negotiated Label Distribution Protocol (LDP) applications. This implementation is in conformance with RFC 7473 (Controlling State Advertisements of Non-negotiated LDP Applications).

The main purpose of documenting this function is to use it in a Multipoint LDP (mLDP)-only environment, wherein participating routers don’t need to exchange any unicast binding information.

Non-Negotiated LDP Applications

The LDP capabilities framework enables LDP applications' capabilities exchange and negotiation, thereby enabling LSRs to send necessary LDP state. However, for the applications that existed prior to the definition of the framework (called *non-negotiated* LDP applications), there is no capability negotiation done. When an LDP session comes up, an LDP speaker may unnecessarily advertise its local state (without waiting for any capabilities exchange and negotiation). In other words, even when the peer session is established for Multipoint LDP (mLDP), the LSR advertises the state for these early LDP applications.

One example is *IPv4/IPv6 Prefix LSPs Setup* (used to set up Label Switched Paths [LSPs] for IP prefixes). Another example is *L2VPN P2P FEC 128 and FEC 129 PWs Signaling* (an LDP application that signals point-to-point [P2P] Pseudowires [PWs] for Layer 2 Virtual Private Networks [L2VPNs]).

In an mLDP-only setup, you can disable these non-negotiated LDP applications and avoid unnecessary LDP state advertisement. An LDP speaker that only runs mLDP announces to its peer(s) its disinterest (or non-support) in non-negotiated LDP applications. That is, it announces to its peers its disinterest to set up IP Prefix LSPs or to signal L2VPN P2P PW, at the time of session establishment.

Upon receipt of such a capability, the receiving LDP speaker, if supporting the capability, disables the advertisement of the state related to the application towards the sender of the capability. This new capability can also be sent later in a Capability message, either to disable a previously enabled application's state advertisement, or to enable a previously disabled application's state advertisement.

As a result, the flow of LDP state information in an mLDP-only setup is faster. When routers come up after a network event, the network convergence time is fast too.

IP Address Bindings In An mLDP Setup

An LSR typically uses peer IP address(es) to map an IP routing next hop to an LDP peer in order to implement its control plane procedures. mLDP uses a peer's IP address(es) to determine its upstream LSR to reach the root node, and to select the forwarding interface towards its downstream LSR. Hence, in an mLDP-only network, while it is desirable to disable advertisement of label bindings for IP (unicast) prefixes, disabling advertisement of IP address bindings will break mLDP functionality.

Uninteresting State - For the *Prefix-LSP* LDP application, *uninteresting* state refers to any state related to IP Prefix FEC, such as FEC label bindings and LDP Status. IP address bindings are not considered as an *uninteresting* state.

For the P2P-PW application LDP application, *uninteresting* state refers to any state related to P2P PW FEC 128 or FEC 129, such as FEC label bindings, MAC address withdrawal, and LDP PW status.

Control State Advertisement

To control advertisement of *uninteresting* state of non-negotiated LDP applications, the capability parameter TLV *State Advertisement Control Capability* is used. This TLV is only present in the Initialization and Capability messages, and the TLV can hold one or more State Advertisement Control (SAC) Elements.

As an example, consider two LSRs, S (LDP speaker) and P (LDP peer), that support all non-negotiated applications. S is participating (or set to participate) in an mLDP-only setup. Pointers for this scenario:

- By default, the LSRs will advertise state for all LDP applications to their peers, as soon as an LDP session is established.
- The **capabilities sac mldp-only** function is enabled on S.
- P receives an update from S via a Capability message that specifies to disable all four non-negotiated applications states.

- P's outbound policy towards S blocks and disables state for the unneeded applications.
- S only receives mLDP advertisements from specific mLDP-participating peers.

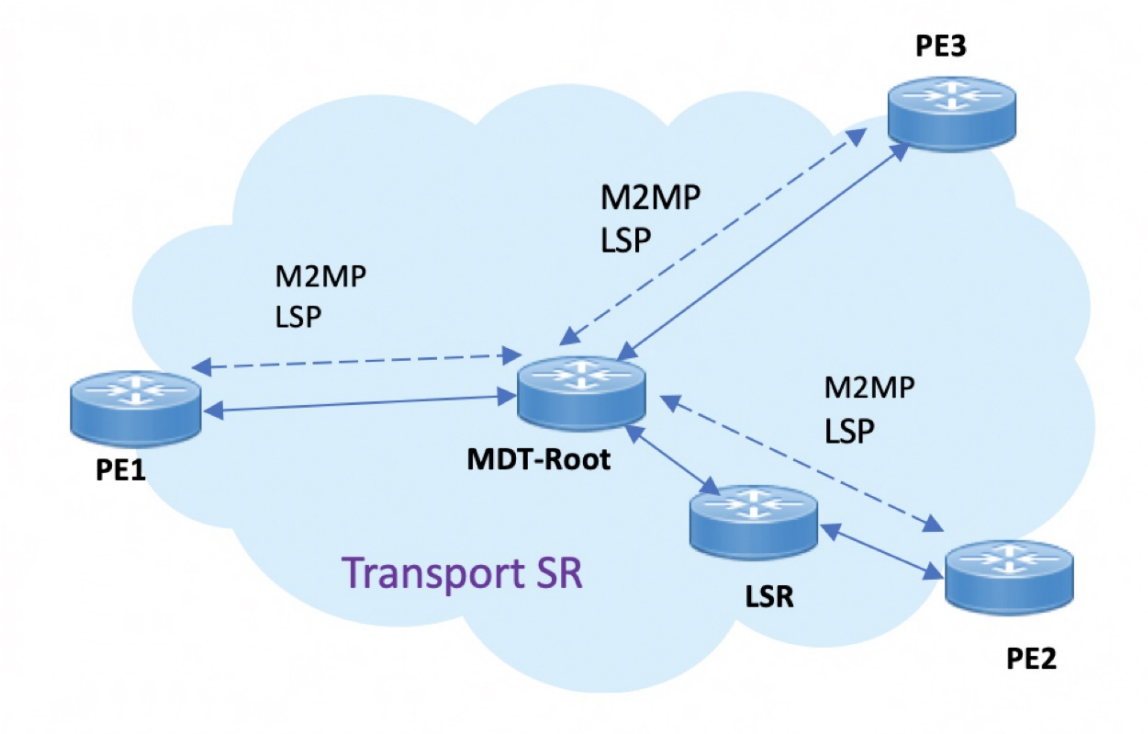
Use Cases For Controlling State Advertisements

Two use cases are explained, **mLDP-Based MVPN** and **Disable Prefix-LSPs On An L2VPN/PW tLDP Session**.

mLDP-Based MVPN

A sample topology and relevant configurations are noted below.

Figure 7: mLDP-Based MVPN Over Segment Routing



- The topology represents an MVPN profile 1 where an mLDP-based MVPN service is deployed over a Segment Routing core setup
- mLDP is required to signal MP2MP LSPs, whereas SR handles the transport.
- SAC capabilities are used to signal *mLDP-only* capability, which blocks unrequired unicast IPv4, IPv6, FEC128, and FEC129 related label binding advertisements.
- The **mldp-only** option is enabled on PE routers and P routers to remove unwanted advertisements.

Configuration

PE1 Configuration

Configure mLDP SAC capability on PE1.

```
PE1(config)# mpls ldp
PE1(config-ldp)# capabilities sac mldp-only
PE1(config-ldp)# commit
```

PE2 Configuration

Configure mLDP SAC capability on PE2.

```
PE2(config)# mpls ldp
PE2(config-ldp)# capabilities sac mldp-only
PE2(config-ldp)# commit
```

Verification

LDP peers (PE1 and PE2) are configured with **mldp-only** option, disabling all other SAC capabilities.

```
PE1# show running-config mpls ldp

mpls ldp
   capabilities sac mldp-only
   mldp
        address-family ipv4
        !

PE2# show running-config mpls ldp

mpls ldp
   capabilities sac mldp-only
   mldp
        address-family ipv4
        !
```

On PE1, verify PE2's SAC capabilities:

```
PE1# show mpls ldp neighbor 209.165.201.20 capabilities detail

Peer LDP Identifier: 209.165.201.20:0
  Capabilities:
    Sent:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}{IPv6-disable}{FEC128-disable}{FEC129-disable} ](length 4)
    Received:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}{IPv6-disable}{FEC128-disable}{FEC129-disable} ](length 4)
```

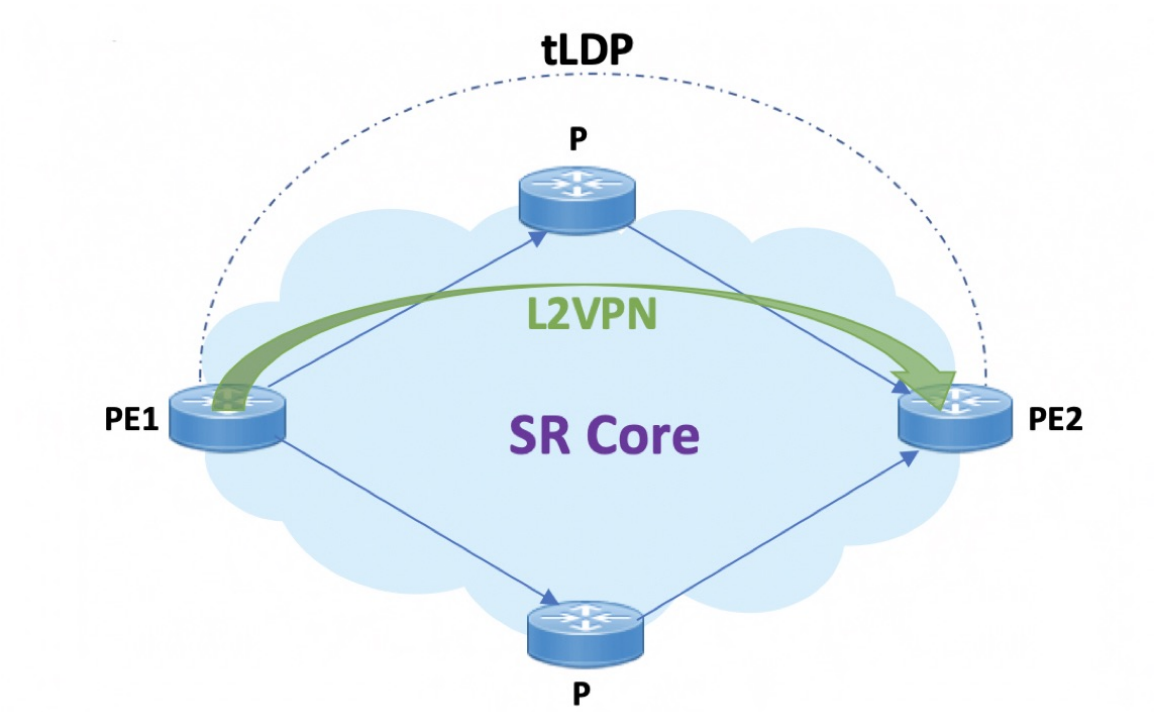
Capabilities Sent shows that **mldp-only** option disables all other advertisements.

Capabilities Received shows that **mldp-only** is enabled on peer PE2 too.

Disable Prefix-LSPs On An L2VPN/PW tLDP Session

A sample topology and relevant configurations are noted below.

Figure 8: L2VPN Xconnect Service Over Segment Routing



- The topology represents an L2VPN Xconnect service over a Segment Routing core setup.
- By default, Xconnect uses tLDP to signal service labels to remote PEs.
- By default, tLDP not only signals the service label, but also known (IPv4 and IPv6) label bindings to the tLDP peer, which is not required.
- The LDP SAC capabilities is an optional configuration enabled under LDP, and users can block IPv4 and IPv6 label bindings by applying configurations on PE1 and PE2.

Configuration

PE1 Configuration

Disable IPv4 prefix LSP binding advertisements on PE1:

```
PE1(config)# mpls ldp capabilities sac ipv4-disable
PE1(config)# commit
```

Disable IPv6-prefix LSP binding advertisements on PE1:

```
PE1(config)# mpls ldp capabilities sac ipv4-disable ipv6-disable
PE1(config)# commit
```



Note Whenever you disable a non-negotiated LDP application state on a router, you must include previously disabled non-negotiated LDP applications too, in the same command line. If not, the latest configuration overwrites the existing ones. You can see that ipv4-disable is added again, though it was already disabled.

PE2 Configuration

Enable SAC capability awareness on PE2, and make PE2 stop sending IPv4 prefix LSP binding advertisements to PE1:

```
PE2(config)#mpls ldp capabilities sac
PE2(config)#commit
```

Verification

On PE1, verify PE2's SAC capabilities:

```
PE1# show mpls ldp neighbor 198.51.100.1 detail

Peer LDP Identifier: 198.51.100.1:0
  TCP connection: 198.51.100.1:29132 - 192.0.2.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 14/14; Downstream-Unsolicited
  Up time: 00:03:30
  LDP Discovery Sources:
    IPv4: (1)
      Targeted Hello (192.0.2.1 -> 198.51.100.1, active)
    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (3)
      203.0.113.1     209.165.201.1    10.0.0.1   198.51.100.1
      172.16.0.1
    IPv6: (0)
  Peer holdtime: 180 sec; KA interval: 60 sec; Peer state: Estab
  NSR: Disabled
  Clients: AToM
  Capabilities:
    Sent:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
        [ {IPv4-disable}  ] (length 1)
    Received:
      0x508  (MP: Point-to-Multipoint (P2MP))
      0x509  (MP: Multipoint-to-Multipoint (MP2MP))
      0x50b  (Typed Wildcard FEC)
      0x50d  (State Advertisement Control)
```

Capabilities Sent SAC capability **ipv4-disable** is sent, and local IPv4 label bindings are not generated.

Capabilities Received The peer (PE2) understands SAC capability and won't send its local IPv4 label bindings to local PE.

On PE1, verify SAC capabilities:

```
PE1# show mpls ldp capabilities detail

  Type     Description                                Owner
  ------   ----------------------------------------   ------------
```

```
0x50b    Typed Wildcard FEC                          LDP
   Capability data: None

0x3eff   Cisco IOS-XR                                LDP
   Capability data:
     Length: 12
     Desc  : [ host=PE1; platform=ASR9000; release=07.01.01 ]

0x508    MP: Point-to-Multipoint (P2MP)              mLDP
   Capability data: None

0x509    MP: Multipoint-to-Multipoint (MP2MP)        mLDP
   Capability data: None

0x50d    State Advertisement Control                 LDP
   Capability data:
     Length: 1
     Desc  : [ {IPv4-disable}  ]

0x703    P2MP PW                                     L2VPN-AToM
   Capability data: None
```

On PE1, verify that local and remote FEC bindings are removed.

```
PE1# show mpls ldp neighbor 198.51.100.1
Wed March 3 13:42:13.359 EDTs
```

ECMP and Bundle Hashing with Entropy Label

Entropy label (EL) improves load balancing across a network. Load balancing helps in planning the capacity of a network by distributing traffic across multiple paths that are based on hashing functions.

Note The routers do not support imposition or disposition of EL.

Traffic load balancing over Equal Cost Multipath (ECMP) or Link Aggregation Groups (LAGs) is based on a hashing function. To arrive at the hash calculations, the node that performs the load balancing must read the header fields in the incoming packets. Currently, Label Switching Routers (LSRs) at each transit point must do a Deep Packet Inspection (DPI) along the path of a given Label Switched Path (LSP). This includes extracting the appropriate keys for load balancing. If the LSR is unable to infer the protocol, it uses the topmost MPLS labels in the label stack as keys to balance load. This may result in an unbalanced distribution of traffic.

Entropy labels enhance load balancing by eliminating the need for DPI at the transit LSRs. The transit router recognizes the incoming MPLS packets with an entropy label and performs the load balancing and forwards the MPLS packet on a selected path.

The ingress LSR of an LSP computes the hash that is based on appropriate fields from a given packet and places the result in a label that is called an entropy label as part of the MPLS label stack. Using the entropy label in the hash keys reduces the need of a DPI in the LSR. The transit LSR can use the entire label stack of the MPLS packet to perform load balancing, as the entropy label introduces the right level of order into the label stack.

For more information on EL, see *RFC 6790*.

MPLS Hashing

The hashing uses the label stack and the payload when the label stack contains EL. However, load balancing functionality considers EL like any other label.

Starting from the Cisco IOS-XR Release 7.3.1, hashing is performed as described in the following table:

Table 11: MPLS Hashing

MPLS Payload	Fields Considered for Hashing	Description
IPv4	Router ID + Label stack + Src IP + Dst IP + L4 Protocol + Src Port + Dst Port	If IPv4 is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • Source and destination ports are used if the protocol is UDP or TCP. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.
IPv6	Router ID + Label stack + Src IP + Dst IP + Flow-label + L4 Protocol + Src Port + Dst Port	If IPv6 is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • Source and destination ports are used if the protocol is UDP or TCP. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.
GTP-u (IPv4 or IPv6)	Router ID + Label stack + GTP TEID	GPRS tuneling protocol. GTP-u uses UDP destination port 2152. The same fields are used for both IPv4 and IPv6. GTP TEID is a 32-bit tunnel end-point identifier.

MPLS Payload	Fields Considered for Hashing	Description
Ethernet	Router ID + Label stack + Dest MAC + Src MAC + Ether-type + VLAN	If Ethernet is the MPLS payload, hashing uses: • Up to 14 labels in the label stack. • For untagged Ethernet, destination MAC address, source MAC address, and Ethernet type are used. • For tagged Ethernet, destination MAC address, source MAC address, Ethernet type, and first VLAN tags are used. • When L2VPN is configured with the Control word, destination MAC address and source MAC address are used. The cyclic redundancy check (CRC) hash value(s) for the same packet varies based on whether the ingress network processing unit (NPU) is Q100 or Q200.

Load Balancing based on the Position of Entropy Label

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
Load Balancing based on the Position of Entropy Label	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Load Balancing based on the Position of Entropy Label	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Load Balancing based on the Position of Entropy Label	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Load Balancing based on the Position of Entropy Label	Release 7.5.2	The router achieves multi-pathing or traffic load balancing based on the Entropy Label (EL) and Entropy Label Indicator (ELI) position. The load balancing is based on the router ID and label stack if the ELI is in any of the three bottom labels. If not, the load balancing is based on all labels in the label stack plus the MPLS payload.

MPLS label stack contains Entropy Label (EL) and Entropy Label Indicator (ELI). The label immediately preceding an EL in the MPLS label stack is an ELI. The ELI uses a reserved label value of 7.

Starting from the Cisco IOS-XR Release 7.5.2, the load balancing is based on the placement of the EL and ELI. If the ELI and EL are placed at the bottom three label entry positions, load balancing uses all labels, else uses the MPLS payload along with other labels.

For GTP payload, the presence of EL does not change the hashing mechanism and it remains the same as described in the *MPLS Hashing* table.

The following table shows how load balancing is performed based on the position of ELI.

Position of ELI (Label Value 7)	Description
Bottom three labels	If there is ELI in any of the three bottom label positions, the load balacing is based on the router ID and label stack. The load balacing uses up to 14 labels in the label stack.
Anywhere else in the stack	The load balacing is performed based on the entropy label as described in the *MPLS Hashing* table. The load balacing uses all labels in the label stack and MPLS payload.

Additional References

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

Related Documents

Related Topic	Document Title
LDP Commands	*MPLS Label Distribution Protocol Commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

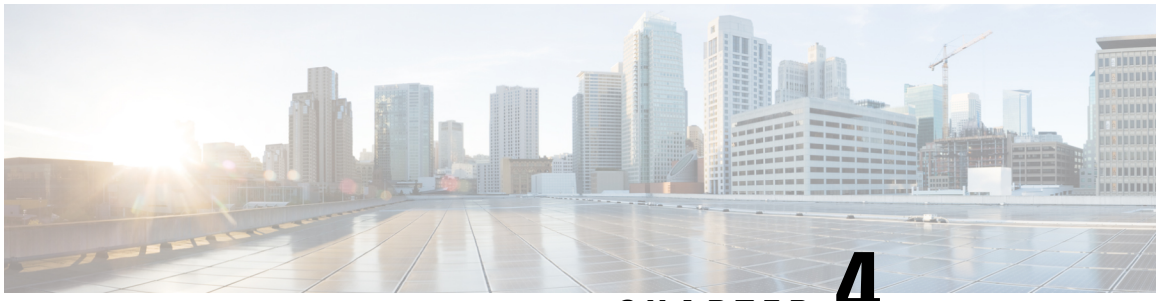
MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

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

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 4

Implementing MPLS OAM

- IP-Less MPLS-TP Ping and MPLS-TP Traceroute , on page 97
- MPLS LSP Ping, on page 97
- MPLS LSP Traceroute, on page 99
- MPLS OAM using nil FEC, on page 102
- Self-ping probe in MPLS-TE, on page 105

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

In Label Switched Path (LSP) ping or traceroute with IP encapsulation over ACH, IP encapsulated ping or traceroute packets are sent over the MPLS LSP using the control channel (ACH). The application-level control channel in this case is the reverse path of the LSP using ACH. The on-demand ping or traceroute echo response message is sent on the reverse path of the LSP. The response uses ACH and is IP encapsulated. The destination address in the IP header is set to that of the sender of the echo request message, and the source address in the IP header is set to a valid address of the replying node.

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

MPLS LSP Ping

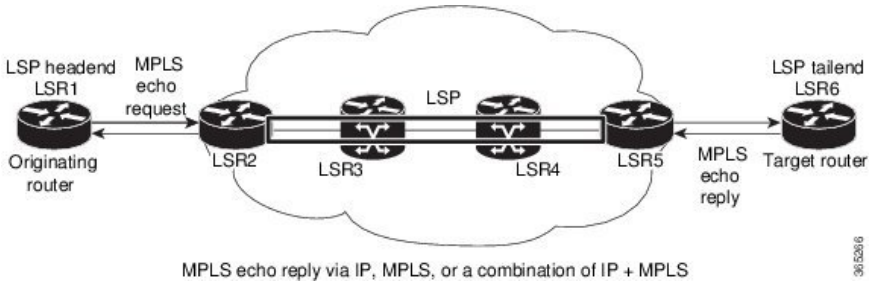
The MPLS LSP Ping feature is used to check the connectivity between Ingress LSR and egress LSRs along an LSP. MPLS LSP ping uses MPLS echo request and reply messages, similar to Internet Control Message Protocol (ICMP) echo request and reply messages, to validate an LSP. While ICMP echo request and reply messages validate IP networks, MPLS echo and reply messages validate MPLS networks. 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 and it 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.

The following figure shows MPLS LSP ping echo request and echo reply paths.

Figure 9: MPLS LSP Ping Echo Request and Reply Paths



By default, the **ping mpls ipv4** command tries to determine the Forwarding Equivalence Class (FEC) being used automatically. However, this is only applicable at head-end and works only if the FEC at the destination is same as the source. If the source and destination FEC types are not the same, the **ping mpls ipv4** command may fail to identify the targeted FEC type. You can overcome this limitation by specifying the FEC type in MPLS LSP ping using the **fec-type** command option. If the user is not sure about the FEC type at the transit or the destination, or it may change through network, use of the **generic** FEC type command option is recommended. Generic FEC is not coupled to a particular control plane and allows path verification when the advertising protocol is unknown, or may change during the path of the echo request. If you are aware of the destination FEC type, specify the target FEC as BGP or LDP.

Configuration Examples

This example shows how to use MPLS LSP ping to test the connectivity of an IPv4 LDP LSP. The destination is specified as a Label Distribution Protocol (LDP) IPv4 address.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 verbose

Sun Nov 15 11:27:43.070 UTC

Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 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

Type escape sequence to abort.

!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4 ms
```

In this example, the destination is specified as a Label Distribution Protocol (LDP) IPv4 prefix and Forwarding Equivalence Class (FEC) type is specified as generic.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 fec-type generic

Wed Nov 25 03:36:33.143 UTC
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 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

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

In this example, the destination is specified as a Label Distribution Protocol (LDP) IPv4 prefix and the FEC type is specified as BGP.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 fec-type bgp

Wed Nov 25 03:38:33.143 UTC
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 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

Type escape sequence to abort.

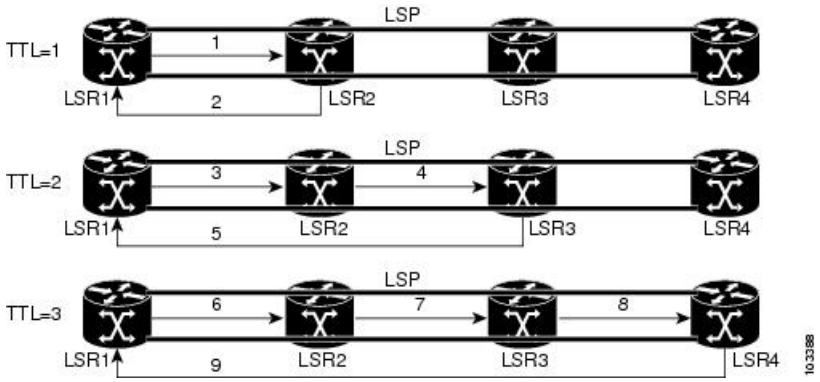
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

MPLS LSP Traceroute

The MPLS LSP Traceroute feature is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP Traceroute feature relies on the expiration of the Time to Live (TTL) value of the packet that carries the echo request. When the MPLS echo request message hits a transit node, it checks the TTL value and if it is expired, the packet is passed to the control plane, else the message is forwarded. If the echo message is passed to the control plane, a reply message is generated based on the contents of the request message.

The following figure shows an MPLS LSP traceroute example with an LSP from LSR1 to LSR4.

Figure 10: MPLS LSP Traceroute



By default, the **traceroute mpls ipv4** command tries to determine the Forwarding Equivalence Class (FEC) being used automatically. However, this is only applicable at head-end and works only if the FEC at the destination is same as the source. If the source and destination FEC types are not the same, the **traceroute mpls ipv4** command may fail to identify the targeted FEC type. You can overcome this limitation by specifying the FEC type in MPLS LSP traceroute using the **fec-type** command option. If the user is not sure about the FEC type at the transit or the destination, or it may change through network, use of the **generic** FEC type command option is recommended. Generic FEC is not coupled to a particular control plane and allows path verification when the advertising protocol is unknown, or may change during the path of the echo request. If you are aware of the destination FEC type, specify the target FEC as BGP or LDP.

Configuration Examples

This example shows how to use the **traceroute** command to trace to a destination.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 destination 127.0.0.3 127.0.0.6 2
Sat Jan 27 03:50:23.746 UTC

Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

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

Type escape sequence to abort.

Destination address 127.0.0.3
  0 10.2.1.2 MRU 1500 [Labels: 24000 Exp: 0]
L 1 10.2.1.1 MRU 1500 [Labels: implicit-null Exp: 0] 8 ms
! 2 10.1.0.2 3 ms

Destination address 127.0.0.5
  0 10.2.1.2 MRU 1500 [Labels: 24000 Exp: 0]
L 1 10.2.1.1 MRU 1500 [Labels: implicit-null Exp: 0] 5 ms
! 2 10.1.0.2 2 ms
```

This example shows how to use the **traceroute** command and how to specify the maximum number of hops for the traceroute to traverse by specifying the **ttl** value.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 ttl 1
Sun Nov 15 12:20:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

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

Type escape sequence to abort.

  0 10.1.0.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.1.0.2 3 ms
```

This example shows how to use the **traceroute** command to trace to a destination and FEC type is specified as generic.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 fec-type generic
Sun Nov 15 12:25:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

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
Type escape sequence to abort.
0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms
```

This example shows how to use the **traceroute** command to trace to a destination and FEC type is specified as BGP.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 fec-type bgp
Sun Nov 15 12:25:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

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
Type escape sequence to abort.
0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms
```

MPLS OAM using nil FEC

Table 13: Feature History Table

Feature Name	Release Information	Feature Description
MPLS OAM using nil FEC	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
MPLS OAM using nil FEC	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is now supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
MPLS OAM using nil FEC	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
MPLS OAM using nil FEC	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The Nil-FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute. MPLS ping and traceroute requires at least one forwarding equivalence class (FEC) in the target FEC stack. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

The Nil-FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute. MPLS ping and traceroute requires at least one forwarding equivalence class (FEC) in the target FEC stack. In Nil-FEC ping and traceroute operations, an explicit FEC is not associated with the label. Nil-FEC LSP ping and traceroute support MPLS static LSPs and also act as an additional diagnostic tool for all other LSP types. Nil-FEC LSP ping and traceroute allow network operators to provide the ability to freely test any label stack by allowing them to specify the following:

- label stack
- outgoing interface
- nexthop address

The following table shows the syntax for the ping and traceroute commands.

Table 14: LSP Ping and Traceroute Nil FEC Commands

Command Syntax
ping mpls nil-fec labels {*label*[**,***label*]} [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]]
traceroute mpls nil-fec labels {*label*[**,***label*]} [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]]

Examples: LSP Ping Nil FEC and LSP Traceroute Nil FEC

The examples in this section use the following topology:

```
Node loopback IP address: 172.18.1.3   172.18.1.4   172.18.1.5   172.18.1.7
Node label:               16003        16004        16005        16007
Nodes:                    Arizona ---- Utah ------- Wyoming ---- Texas

Interface:            GigabitEthernet0/2/0/1   GigabitEthernet0/2/0/1
Interface IP address:         10.1.1.3              10.1.1.4
```

```
RP/0/RP0/CPU0:router-arizona# show mpls forwarding

Tue May  2 13:44:31.999 EDT
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16004  Pop         No ID              Gi0/2/0/1    10.1.1.4        1392
       Pop         No ID              Gi0/2/0/2    10.1.2.2        0
16005  16005       No ID              Gi0/2/0/0    10.1.1.4        0
       16005       No ID              Gi0/2/0/1    10.1.2.2        0
16007  16007       No ID              Gi0/2/0/0    10.1.1.4        4752
       16007       No ID              Gi0/2/0/1    10.1.2.2        0
```

This example shows how to use Nil-FEC LSP ping to test a label stack.

```
RP/0/RP0/CPU0:router-arizona# ping mpls nil-fec labels 16005,16007 output interface
GigabitEthernet 0/2/0/1 nexthop 10.1.1.4 repeat 1
Sending 1, 72-byte MPLS Echos with Nil FEC labels 16005,16007,
     timeout is 2 seconds, send interval is 0 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 label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 1/1/1 ms
 Total Time Elapsed 0 ms
```

This example shows how to use Nil-FEC LSP traceroute for a label stack.

```
RP/0/RP0/CPU0:router-arizona# traceroute mpls nil-fec labels 16005,16007 output interface
GigabitEthernet 0/2/0/1 nexthop 10.1.1.4
Tracing MPLS Label Switched Path with Nil FEC labels 16005,16007, timeout is 2 seconds

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 label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'd' - see DDMAP for return code,
```

```
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 10.1.1.3 MRU 1500 [Labels: 16005/16007/explicit-null Exp: 0/0/0]
L 1 10.1.1.4 MRU 1500 [Labels: implicit-null/16007/explicit-null Exp: 0/0/0] 1 ms
L 2 10.1.1.5 MRU 1500 [Labels: implicit-null/explicit-null Exp: 0/0] 1 ms
! 3 10.1.1.7 1 ms
```

Self-ping probe in MPLS-TE

A self-ping probe is a process that

- sends a test packet, usually UDP, from the ingress Label Edge Router (LER) across a newly established Label Switched Path (LSP)
- receives the returned packet from the egress LER, and
- verifies that the LSP is forwarding data correctly.

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
Self-ping probe for reoptimized LSP	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*)(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Self-ping probe for current LSP	Release 25.3.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC:Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) The self-ping probe functionality is extended to support the current LSP that allows the network to promptly confirm the readiness of existing LSP to handle traffic. This immediate verification helps prevent traffic drops by ensuring that the data path is fully operational before forwarding actual user traffic. The self-ping option is configured under a named tunnel and the operational status of the tunnel is determined by probe packet or during session timeout.

Self-ping probe for reoptimized LSP	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M
Self-ping probe for reoptimized LSP	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Self-ping probe for reoptimized LSP	Release 7.5.3	In a reoptimized LSP, the reopt-install timer helps to prevent traffic issue. This process is made possible by enabling the label edge router (LER) to send self-ping probes over the reoptimized LSP to the ingress LER. As soon the probe reaches the LER, there's confirmation that the RSVP programming is complete along the path. Post this confirmation, the LER switches traffic to the reoptimized LSP with no drop in traffic. This feature introduces the **self-ping** keyword in the **named-tunnels tunnel-te** command.

Supported scenarios in self-ping probe

- Self-ping probe for current LSP—refers to the active or presently established Label Switched Path being used for forwarding traffic between a specific pair of ingress and egress routers in an MPLS network.
- Self-ping probe for reoptimized LSP—refers to a Label Switched Path that has been recalculated and reestablished to improve its performance, efficiency, or resource utilization.

Self-ping probe for current LSP

When a tunnel lacks an active path—such as when it is newly created or has just recovered from a down state—a new LSP is established. In the traditional process, the tunnel is considered operational and ready for use as soon as RSVP signaling is completed. However, there can be delays in programming the forwarding state at some nodes along the path, which means that if traffic is sent immediately, it may be lost if those nodes are not yet fully ready.

To address this, self-ping introduces an extra verification step to LSP setup.The ingress LER sends a test probe to the egress LER. When the probe is successfully returned to the ingress LER, the tunnel is ready for carrying traffic. This ensures that the tunnel is ready for data forwarding and prevents any traffic loss.

Self-ping probe for reoptimized LSP

During the reoptimization of an LSP, the LER must wait until the reoptimized path is fully established before switching traffic onto the new LSP.

If the LER does not wait long enough and RSVP programming for MPLS forwarding is not complete at a specific hop along the path, traffic may be dropped at that hop.

The LER operates at the edge of an MPLS network and serves as the entry and exit point for the network.

Reopt-install timer function

The edge router is equipped with an internal reopt-install timer that provides sufficient time for the reoptimized path to be established before directing traffic onto it. If the LER waits too long before switching, traffic continues to use the old path, potentially causing congestion.

To avoid potential traffic loss, operators often use a cautious approach by waiting longer before transitioning traffic to the reoptimized LSP. The self-ping can bypass long waiting time and switch traffic to reoptimitzed LSP as soon as the self-ping probe proves that the LSP is operational.

Restriction for self-ping probe support

- Self-ping is supported only with named tunnels.
- Self -ping is not applied for LSP which has **1-hop long** path even it is configured for the tunnel.

How self-ping probe works

Summary

The self-ping probe mechanism dynamically and reliably detects when a reoptimized LSP is ready for traffic switching. The key components involved in the process are:

- Ingress Label Edge Router (LER): Sends self-ping probes over the current or reoptimized LSP.
- Egress Label Edge Router (LER): Receives probes and sends responses.
- Self-ping probe: A probe sent once per second to verify LSP readiness.

The process involves these stages:

1. The ingress LER sends a self-ping probe to the egress LER over the current or reoptimized LSP at a fixed rate of one probe per second. This frequency is consistent regardless of the LSP type and is not configurable.
2. Upon receiving the probe response, the ingress LER declares the LSP as Up.
3. The ingress LER immediately switches traffic to the new, reoptimized path.
4. After a successful switchover, no further probes are sent for that LSP.

5. If the self-ping mechanism sends the specified number of probes without receiving a response, additional handling occurs.

Result:

This process ensures timely and reliable detection of reoptimized LSP readiness, enabling immediate traffic switchover to the optimal path and minimizing downtime.

Configure the self-ping probe

The purpose of this task is to enable self-ping probe in current and reoptimized LSPs.

Procedure

Step 1 Configure an MPLS-TE tunnel and enable the self-ping feature to verify the tunnel's path before traffic is sent over it.

Example:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels tunnel-te ABC
Router(config-te-tun-name)# self-ping
```

Step 2 *Optional:* Set the maximum number of self-ping probes.

Example:

```
Router(config-te-tun-name)# max-count 10
Router(config-te-tun-name)# commit
```

The ingress LER sends up to 10 probes to check the LSP before making a decision.

If not set manually, the default number of probes sent is 60.

Step 3 Run the **show mpls traffic-eng tunnels name** *tunnel-name* detail to verify the readiness of the MPLS-TE tunnel.

Example:

```
Router# show mpls traffic-eng tunnels name ABC detail

Name: ABC  Destination: 192.168.0.4  Ifhandle:0x2d0
  Tunnel-ID: 32783
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Thu Apr  7 14:54:30 2022 (00:00:01 ago)
  Config Parameters:
    ...
Load-interval: 300 seconds
    Auto-bw: disabled
    Auto-Capacity: Disabled:
    Self-ping: Enabled
      Maximum-probes: 10
      Probes-period: 1 second(s)
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
```

```
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  Self-ping:
    Status: Not executed
  SNMP Index: 0
  Binding SID: 0
  History:
    Tunnel has been up for: 00:00:00 (since Thu Jan 01 01:00:00 BST 1970)
    Current LSP:
      Uptime: 00:00:00 (since Thu Jan 01 01:00:00 BST 1970)
  Current LSP Info:
    Instance: 2, Signaling Area: OSPF 100 area 0
    Uptime: 00:00:00 (since Thu Apr 07 14:54:31 BST 2022)
    Outgoing Interface: GigabitEthernet0/2/0/0, Outgoing Label: 24000
    Next Hop: 10.10.10.2      Neighbor Next Hop: 0.0.0.0
    Router-IDs: local      192.168.0.1
                downstream 192.168.0.2
...
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Step 4 Run the **show mpls traffic-eng self-ping statistics** command to provide an overview of the activity, success, and failure rates of MPLS-TE tunnel self-ping operations.

Example:

```
Router# show mpls traffic-eng self-ping statistics
Wed Jun 15 14:24:29.316 BST
Self-Ping Statistics:
  Collected since: Tue Jun 14 09:35:52 2022 (1d04h ago)
  Operations:
    Started 2
    Running 0
    Successful 1
    Timed-out 1
    Terminated 0
  Probes sent 11
  Probes failed 0
  Received responses 1 (Average response time 00:00:00)
  Mismatched responses 0
```

You can clear the statistics using **clear mpls traffic-eng self-ping statistics** command. If there are any active sessions running, this field is not reset to zero, and the started count is also set to this value.

Self-ping probe status

During the self-ping procedure, the tunnel goes through various stages and the LER shows the status when you run the **show mpls traffic-eng tunnels detail** command.

This table describes the status and outcomes of self-ping probes in the MPLS RSVP-TE environment.

Table 16: Self-ping probe status

Self-ping status	Description	Sample ouput
Not executed	Although self-ping is configured on the ingress LER, the request to setup new LSP is not triggered.	`Self-ping:` `  Status: Not executed`

Self-ping status	Description	Sample ouput

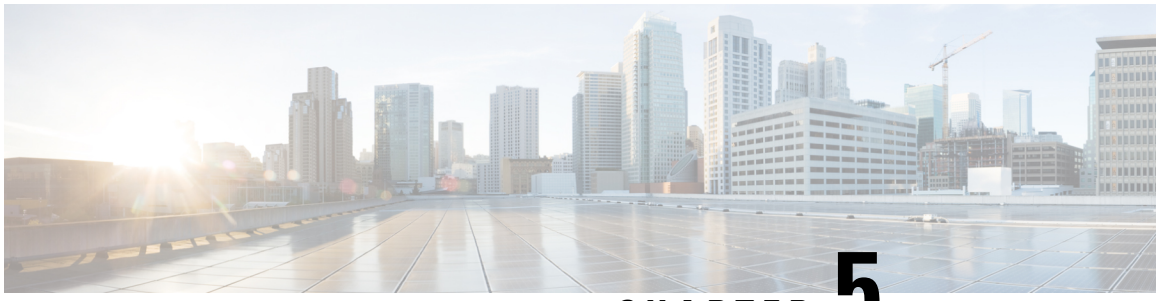
Underway	LSP verification is ongoing and the self-ping is actively sending probes and waiting to receive the response.	`Self-ping:`
		`  Status: Underway`
		`  LSP-ID: 6`
		`  Started: 00:00:00 (since Thu Apr 07 15:03:33 BST 2022)`

Succeeded	The ingress LER receives the response from the egress LER and and LSP is ready to carry the traffic.	`Self-ping:`
		`  Status: Succeeded (in 0 seconds)`
		`  LSP-ID: 4`
		`  Probes sent: 1`
		`  Started: 00:00:00 (since Thu Apr 07 15:02:23 BST 2022)`
		`  Stopped: 00:00:00 (since Thu Apr 07 15:02:23 BST 2022)`
		`  Response Received: 00:00:00 (since Thu Apr 07 15:02:23`
		`  BST 2022)`

Timed-out	After reaching the maximum number of probes with no response, the ingress LER falls back to the reopt-install timer like behavior and LSP is considered active, and traffic is switched to the reoptimized LSP after the reopt-install timer expires.	`Self-ping:`
		`  Status: Timed-out (in 9 seconds)`
		`  LSP-ID: 6`
		`  Probes sent: 10`
		`  Started: 00:00:43 (since Thu Apr 07 15:03:33 BST 2022)`
		`  Stopped: 00:00:34 (since Thu Apr 07 15:03:42 BST 2022)`

Self-ping status	Description	Sample ouput

Terminated	The LER terminates self-ping execution when:
	• LSP under self-ping is obsolete. For example:
	  • The LSP under setup has failed.
	  • LSP path becomes invalid.
	  • New LSP is obsolete.
	  • Tunnel is being shut down.
	  • Self-ping configuration is removed or whole tunnel is removed.
	• The self-ping process is managed by the self-ping watchdog timer. If self-ping does not return a success or timeout result, the watchdog timer will forcefully stop and clean up the self-ping session, and the tunnel will proceed as if a self-ping timeout had occurred.
		`Self-ping:`
		`  Status: Terminated (in 20 seconds)`
		`  LSP-ID: 8`
		`  Started: 00:00:21 (since Thu Apr 07 15:05:44 BST 2022)`
		`  Stopped: 00:00:01 (since Thu Apr 07 15:06:04 BST 2022)`



CHAPTER 5

MPLS Static Labeling

Table 17: Feature History Table

Feature Name	Release Information	Feature Description
MPLS static labeling	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
MPLS static labeling	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
MPLS static labeling	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

MPLS static labeling	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) MPLS Static Labeling allows for manual assignment of labels to specific routes, providing fine-grained control over label distribution and traffic engineering. This feature is ideal for network scenarios where predictable and consistent routing paths are required. By configuring static labels, network operators can optimize resource allocation and ensure precise traffic management without relying on dynamic label distribution protocols. *Previously this feature was supported on Q200 and Q100. • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

The MPLS static feature enables you to statically assign local labels to an IPv4/Ipv6 prefix. Also, Label Switched Paths (LSPs) can be provisioned for these static labels by specifying the next-hop information that is required to forward the packets containing static label.

If there is any discrepancy between labels assigned statically and dynamically, the router issues a warning message in the console log. By means of this warning message, the discrepancy can be identified and resolved.

The advantages of static labels over dynamic labels are:

- Improves security because the risk of receiving unwanted labels from peers (running a compromised MPLS dynamic labeling protocol) is reduced.
- Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

Restrictions

- The router does not prevent label discrepancy at the time of configuring static labels. Any generated discrepancy needs to be subsequently cleared.
- Equal-cost multi-path routing (ECMP) is not supported.
- Interfaces must be explicitly configured to handle traffic with static MPLS labels.

- The MPLS per-VRF labels cannot be shared between MPLS static and other applications.

- Forwarding Labeled Packets, on page 115
- Define Label Range and Enable MPLS Encapsulation, on page 116
- Identify and Clear Label Discrepancy, on page 117
- Configuring Backup within a Forwarding Set, on page 118
- Configuring Static LSP Next Hop Resolve, on page 121
- Configuring Static LSP Next Hop Resolve with Recursive Prefix, on page 122
- MPLS Static Labeling , on page 123
- MPLS Static Forwarding Over A BVI , on page 125
- MPLS Over GRE Tunnels, on page 129

Forwarding Labeled Packets

This section describes how labeled packets are forwarded in MPLS networks, how forwarding labeled packets are different from forwarding IP packets, how labeled packets are load-balanced, and what a LSR does with a packet with an unknown label.

Top Label Value

When a labeled packet is received, the label value at the top of the stack is looked up. The LSR sees the 20-bit field in the top label, which carries the actual value of the label. As a result of a successful lookup, the LSR learns:

- the next hop to which the packet is to be forwarded.
- what label operation to be performed before forwarding - swap, push, or pop.

The processing is always based on the top label, without regard to the possibility that in the past some other number of another label may have been "above it", or at present that some other number of another label may be below it. An unlabeled packet can be thought of as a packet whose label stack is empty (that is, a packet whose label stack has depth zero).

IP Lookup Versus Label Lookup

When a router receives an IP packet, an IP lookup is done. This means that the packet is looked up in the Cisco Express Forwarding (CEF) table. When a router receives a labeled packet, the label forwarding information base (LFIB) of the router is looked up. The router knows by looking at the protocol field in the Layer 2 header what type of packet it receives: a labeled packet or an IP packet.

Load Balancing Labeled Packets

If multiple equal-cost paths exist for an IPv4 prefix, Cisco IOS XR Software can load-balance labeled packets. When labeled packets are load-balanced, they can have the same or different outgoing labels. The outgoing labels are the same if the two links are between a pair of routers and both links belong to the platform label space. If multiple next-hop LSRs exist, the outgoing label for each path is usually different, because the next-hop LSRs assign labels independently.

Unknown Label

In regular operations, an LSR should receive only a labeled packet with a label at the top of the stack that is known to the LSR, because the LSR would have previously advertised that label. However, it is possible, in some cases, when something goes amiss in the MPLS network, the LSR starts receiving labeled packets with a top label that the LSR does not find in its LFIB. In such cases, the LSR drops the packet.

Define Label Range and Enable MPLS Encapsulation

By default, MPLS encapsulation is disabled on all interfaces. MPLS encapsulation has to be explicitly enabled on all ingress and egress MPLS interfaces through which the static MPLS labeled traffic travels.

Also, the dynamic label range needs to be defined. Any label that falls outside this dynamic range is available for manually allocating as static labels. The router does not verify statically-configured labels against the specified label range. Therefore, to prevent label discrepancy, ensure that you do not configure static MPLS labels that fall within the dynamic label range.



Note For Cisco IOS XR software release 7.5.2 onwards, MPLS static supports 200G Ethernet.

Configuration Example

You have to accomplish the following to complete the MPLS static labeling configuration. Values are provided as an example.

1. Define a dynamic label range, which in this task is set between 17000 and 18000.
2. Enable MPLS encapsulation on the required interface.
3. Setup a static MPLS LSP for a specific ingress label 24035.
4. Specify the forwarding information so that for packets that are received with the label, 24035, the MPLS protocol swaps labels and applies the label, 24036. After applying the new label, it forwards the packets to the next hop, 10.2.2.2, through the specified interface.

```
Router(config)#mpls label range table 0 17000 18000
Router(config)#commit
Router(config)#mpls static
Router(config-mpls-static)#interface HundredGigE 0/0/0/25
Router(config-mpls-static)#address-family ipv4 unicast
Router(config-mpls-static-af)#local-label 24035 allocate
Router(config-mpls-static-af-lbl)#forward
Router(config-mpls-static-af-lbl-fwd)#path 1 nexthop HundredGigE 0/0/0/27 10.2.2.2 out-label
 24036
Router(config-mpls-static-af-lbl-fwd)# commit
```

Verification

Verify the interfaces on which MPLS is enabled

```
Router# show  mpls interfaces

Interface                  LDP      Tunnel   Static   Enabled
-------------------------- -------- -------- -------- --------
HundredGigE 0/0/0/25             No       No       Yes    Yes
```

Verify that the status is "Created" for the specified label value.

```
Router#show mpls static local-label all

Label   VRF             Type         Prefix           RW Configured   Status
```

```
------- --------------- ------------ ---------------- --------------- --------
24035   default         X-Connect    NA               Yes             Created
```

Check the dynamic range and ensure that the specified local-label value is outside this range.

```
Router#show mpls label range

Range for dynamic labels: Min/Max: 17000/18000
```

Verify that the MPLS static configuration has taken effect, and the label forwarding is taking place.

```
Router#show mpls lsd forwarding

In_Label, (ID), Path_Info: <Type>
24035, (Static), 1 Paths
   1/1: IPv4, 'default':4U, BE1.2, nh=10.20.3.1, lbl=35001, flags=0x0, ext_flags=0x0
```

Associated Commands

- mpls static
- mpls label range
- show mpls interfaces

Identify and Clear Label Discrepancy

During configuring or de-configuring static labels or a label range, a label discrepancy can get generated when:

- A static label is configured for an IP prefix that already has a binding with a dynamic label.
- A static label is configured for an IP prefix, when the same label value is dynamically allocated to another IP prefix.

Verification

Identify label discrepancy by using these show commands.

```
Router#show mpls static local-label discrepancy
Tue Apr 22 18:36:31.614 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect     NA              Yes             Discrepancy


Router#show mpls static local-label all
Tue Apr 22 18:36:31.614 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect    N/A              Yes             Discrepancy
24035   default         X-Connect    N/A              Yes             Created


Router#show log

Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level warnings, 199 messages logged
    Monitor logging: level debugging, 0 messages logged
```

```
    Trap logging: level informational, 0 messages logged
    Buffer logging: level debugging, 2 messages logged

Log Buffer (307200 bytes):

RP/0/RSP0/CPU0:Apr 24 14:18:53.743 : mpls_static[1043]:
%ROUTING-MPLS_STATIC-7-ERR_STATIC_LABEL_DISCREPANCY :
The system detected 1 label discrepancies (static label could not be allocated due to
conflict with other applications).
Please use 'clear mpls static local-label discrepancy' to fix this issue.
RP/0/RSP0/CPU0:Apr 24 14:18:53.937 : config[65762]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'cisco'.
Use 'show configuration commit changes 1000000020' to view the changes.
```

Rectification

Label discrepancy is cleared by allocating a new label to those IP prefixes that are allocated dynamic label. The static label configuration takes precedence while clearing discrepancy. Traffic can be affected while clearing discrepancy.

```
Router# clear mpls static local-label discrepancy all
```

Verify that the discrepancy is cleared.

```
Router# show mpls static local-label all
Wed Nov 25 21:45:50.368 UTC
Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
24000   default         X-Connect    N/A              Yes             Created
24035   default         X-Connect    N/A              Yes             Created
```

Associated Commands

- show mpls static local-label discrepancy
- clear mpls static local-label discrepancy all

Configuring Backup within a Forwarding Set

Various types of FRR backups can be configured between links with in a forwarding path set. You can configure the following types of FRR backups:

- Pure FRR Backup
- Reciprocal FRR backup
- One-way FRR backup

In pure FRR backup, there will be separate primary paths and backup paths. In reciprocal FRR backup, each path can act as both primary and backup. In one-way FRR backup, some paths act as both primary and backup while other paths may be just primary paths or backup paths.

Configuration Example: Pure FRR Backup

This example shows how to configure pure FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
 out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
 out-label 25001 backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
10.5.0.1 out-label pop backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
10.6.0.2 out-label pop backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for pure FRR backup. Here P1-F and P2-F are the forwarding paths and P1-B and P2-B are the backup paths.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Backup • P1-B: N/A P2-B: N/A
P1-F Down	P1-F FRR to P2-F	• P1-F: Up P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Flow • P1-B: Backup P2-B: N/A
P2-F Down	P2-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Backup
P1-B Down	P1-B FRR to P2-B	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Flow

Configuration Example: Reciprocal FRR Backup

This example shows how to configure reciprocal FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
```

```
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
 out-label 25000 primary-and-backup backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
 out-label 25001 primary-and-backup backup-id 1
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
10.5.0.1 out-label pop primary-and-backup backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
10.6.0.2 out-label pop primary-and-backup backup-id 1
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for reciprocal FRR backup.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F FRR to P1-F	• P1-F: Down P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P2-B Down	P2-B FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Down	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Down

Configuration Example: One-way FRR Backup

This example shows how to configure one-way FRR backup with in a forwarding path set.

```
RP/0/0/CPU0:Router# configure terminal
RP/0/0/CPU0:Router(config)# mpls static
RP/0/0/CPU0:Router(config-mpls-static)# lsp lsp1
RP/0/0/CPU0:Router(config-mpls-static-lsp)# in-label 25000 allocate
RP/0/0/CPU0:Router(config-mpls-static-lsp)# forward
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 1 nexthop hundredGigE 0/0/0/25 10.1.0.1
 out-label 25000 backup-id 2
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# path 2 nexthop hundredGigE 0/0/0/26 10.1.0.3
 out-label 25001 primary-and-backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-fwd)# exit
RP/0/0/CPU0:Router(config-mpls-static-lsp)# backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 1 nexthop hundredGigE 0/0/0/27
10.5.0.1 out-label pop backup-id 2
```

```
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# path 2 nexthop hundredGigE 0/0/0/28
10.6.0.2 out-label pop primary-and-backup
RP/0/0/CPU0:Router(config-mpls-static-lsp-backup)# exit
```

The following table describes the forwarding behavior for one-way FRR backup.

Action	Transient State	Interface Steady State	Forward Steady State
N/A	N/A	• P1-F: Up P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Flow • P1-B: N/A P2-B: N/A
P2-F Down	P2-F NO-FRR to P1-F	• P1-F: Down P2-F: Up • P1-B: Up P2-B: Up	• P1-F: Flow P2-F: Down • P1-B: Backup P2-B: N/A
P1-F Down	P1-F FRR to P1-B	• P1-F: Down P2-F: Down • P1-B: Up P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Flow P2-B: Flow
P1-B Down	P1-B FRR to P2-B	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Up	• P1-F: Down P2-F: Down • P1-B: Down P2-B: Flow

Configuring Static LSP Next Hop Resolve

You can specify the outgoing next hop instead of explicitly specifying the outgoing path while configuring static LSPs. This next hop is resolved using the routing information base (RIB) which provides a list of paths to auto-configure. While specifying the next hop for the incoming label in a static LSP, you can specify the next hop address with out the interface using the **resolve-nexthop** command.

The following restrictions apply for this feature:

- Only supports a single next hop address which may resolve to multiple paths.
- Non-default VRFs are not supported.

Configuration Example

This example shows how to configure the static LSP next hop without specifying the interface using the **resolve-nexthop** command.

```
Router# configure terminal
Router(config)# mpls static
```

```
Router(config-mpls-static)# lsp ipv6-2
Router(config-mpls-static-lsp)# in-label 25000 allocate per-prefix 2001:DB8:0:1::/64  or
24:24:1::/64
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 2001:DB8:0:2::64 out-label pop
Router(config-mpls-static-lsp-fwd)# exit
```

Configuring Static LSP Next Hop Resolve with Recursive Prefix

When a routing table entry references to another IP address and not to a directly connected exit interface, the next-hop IP address is resolved using another route with an exit interface. This is known as a recursive look up because multiple lookups are required to resolve the next-hop IP address. Static LSP next hop resolve with recursive prefix feature supports resolution of recursive routes for static LSPs. In this feature, you can specify a next hop which is not directly connected using the **resolve-nexthop** command for a static LSP.

Restrictions

The following restrictions apply for this feature:

- Only eBGP routes are supported.

Configuration Example

This example shows how to configure the static LSP next hop resolve with recursive prefix. Here 192.168.2.1 is a recursive route learnt through eBGP.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp anycast_5001
Router(config-mpls-static-lsp)# in-label 5001 allocate
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 192.168.2.1 out-label pop
Router(config-mpls-static-lsp-fwd)# exit
```

Verification

This example shows how to verify the static LSP next hop resolve with recursive prefix configuration.

```
Router# show  mpls static lsp anycast_5001 detail

LSP Name             Label   VRF             AFI  Type         Prefix           RW Configured
   Status
-------------------- ------- --------------- ---- ------------ -----------------
--------------- --------
anycast_5001          5001    default          N/A  X-Connect    N/A              Yes
      Created
  PRIMARY SET:
    [resolve-mode: nexthop 192.168.2.1]
    Path 0 : nexthop BVI1 10.1.1.3, out-label Pop, Role: primary, Path-id: 0, Status: valid

    Path 1 : nexthop BVI1 10.1.1.4, out-label Pop, Role: primary, Path-id: 0, Status: valid

    Path 2 : nexthop BVI1 10.1.1.5, out-label Pop, Role: primary, Path-id: 0, Status: valid

    Path 3 : nexthop BVI1 10.1.1.6, out-label Pop, Role: primary, Path-id: 0, Status: valid
```

MPLS Static Labeling

Table 18: Feature History Table

Feature Name	Release Information	Feature Description
MPLS static labeling	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
MPLS static labeling	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
MPLS static labeling	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

MPLS static labeling	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) MPLS Static Labeling allows for manual assignment of labels to specific routes, providing fine-grained control over label distribution and traffic engineering. This feature is ideal for network scenarios where predictable and consistent routing paths are required. By configuring static labels, network operators can optimize resource allocation and ensure precise traffic management without relying on dynamic label distribution protocols. *Previously this feature was supported on Q200 and Q100. • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

The MPLS static feature enables you to statically assign local labels to an IPv4/Ipv6 prefix. Also, Label Switched Paths (LSPs) can be provisioned for these static labels by specifying the next-hop information that is required to forward the packets containing static label.

If there is any discrepancy between labels assigned statically and dynamically, the router issues a warning message in the console log. By means of this warning message, the discrepancy can be identified and resolved.

The advantages of static labels over dynamic labels are:

- Improves security because the risk of receiving unwanted labels from peers (running a compromised MPLS dynamic labeling protocol) is reduced.
- Gives users full control over defined LSPs.
- Utilize system resources optimally because dynamic labeling is not processed.

Restrictions

- The router does not prevent label discrepancy at the time of configuring static labels. Any generated discrepancy needs to be subsequently cleared.
- Equal-cost multi-path routing (ECMP) is not supported.
- Interfaces must be explicitly configured to handle traffic with static MPLS labels.

- The MPLS per-VRF labels cannot be shared between MPLS static and other applications.

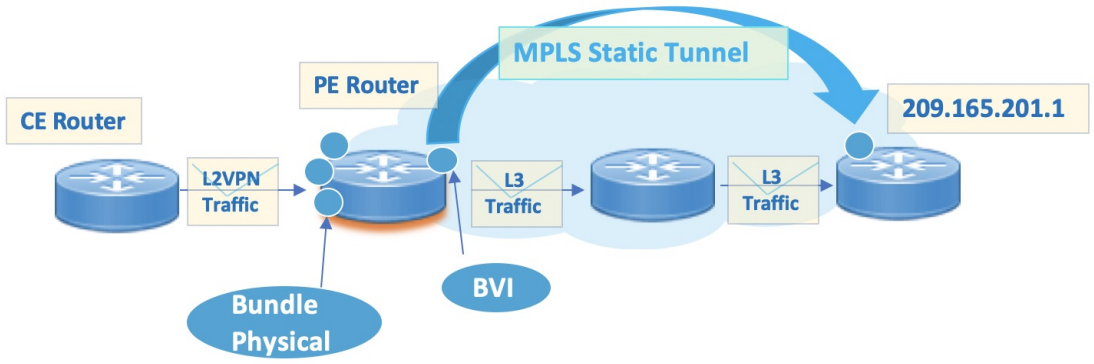
MPLS Static Forwarding Over A BVI

Table 19: Feature History Table

Feature Name	Release Information	Feature Description
MPLS Static Forwarding Over A BVI	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8712-MOD-M routers.
MPLS Static Forwarding Over A BVI	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
MPLS Static Forwarding Over A BVI	Release 7.3.1	The router can receive MPLS L2VPN traffic from an L2 bridge domain, and forward the L3 (customer) traffic over an egress BVI, using an MPLS static LSP. For the incoming L2VPN traffic, the BVI serves as an L3 gateway. Since the router can perform switching for L2 traffic and routing for incoming L3 MPLS traffic, it enhances flexibility for transporting MPLS traffic.

Consider this sample topology, connecting a CE router to a PE router.

Figure 11: MPLS Static Forwarding Over A BVI



Pointers

- L2VPN packets are attached to specific bridge domains, and correspond to a VLAN (or 802.1Q tag). In turn, the VLANs are associated with specific bundle or physical interfaces for sending traffic between the CE and PE routers. These associations have to be configured on the CE and PE routers for transporting L2VPN traffic.
- The L2VPN traffic encapsulates (IPv4 or IPv6) customer payload, and is sent from the CE router to the PE router.
- The PE router does an MPLS label lookup on the incoming MPLS traffic, and removes the VLAN (or 802.1Q) header. In general, the router can perform a label operation like swap, PHP, or pop. After removing the VLAN header, the (previously) encapsulated IP traffic is sent towards the bridge-group virtual interface (BVI).
- With BVI support for the MPLS static function, the incoming labelled traffic can be resolved using a static LSP. The BVI resolves the nexthop to an L3 interface.

BVI pointers

- A BVI next hop can be a static route, a directly connected route, or a route resolved through BGP or an IGP.
- Only an MPLS static LSP can use a BVI as a next hop.

Configuration

The configurations explain how to enable forwarding of (incoming) L2VPN traffic over a (outgoing) BVI, through an MPLS static LSP.

Interfaces Configuration

- The **l2transport** keyword indicates that the interface is an L2 interface, and the L2 traffic belongs to the VLANs specified in the dot1q tags.
- The **rewrite ingress tag pop** command form instructs the router to remove the 802.1Q (or VLAN) tag and forward the payload.
- Corresponding configurations should also be enabled on the CE router.

```
Router# configure terminal
Router(config)# interface Bundle-Ether101.101 l2transport
Router(config-if)# encapsulation dot1q 2001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# mtu 2000

Router# configure terminal
Router(config)# interface Bundle-Ether102.101 l2transport
Router(config-if)# encapsulation dot1q 3001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# mtu 2000

Router# configure terminal
Router(config)# interface HundredGigE0/11/0/25.101 l2transport
Router(config-if)# encapsulation dot1q 101
Router(config-if)# rewrite ingress tag pop 1 symmetric

Router# configure terminal
Router(config)# interface HundredGigE0/11/0/31.101 l2transport
Router(config-if)# encapsulation dot1q 1001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# exit
Router(config)# commit
```

BVI Configuration

The BVI acts as an L3 gateway for the ingress L2VPN traffic. The customer IP traffic is sent to the BVI IP address that is configured in this task.

```
Router# configure terminal
Router(config)# interface BVI101
Router(config-if)# ipv4 address 192.0.2.1 255.255.255.0
Router(config-if)# ipv6 address 2001:DB8:A:B::1/64
Router(config-if)# mac-address 0.10.5500
Router(config-if)# load-interval 30
Router(config-if)# commit
```

L2VPN Configuration

- L2VPN is configured, and a bridge domain (bd1) is associated with it.
- A bundle interface and BVI are associated with the BD. The instructions enable VLAN traffic received at the bundle interface (Bundle-Ether101.101) to be routed through the L3 gateway, BVI BVI101. VLAN-to-interface association was done in an earlier step.

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface Bundle-Ether101.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
```

Similar configurations are done for other interfaces.

```
Router(config-l2vpn-bg-bd)# interface Bundle-Ether102.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
.
Router(config-l2vpn-bg-bd)# interface HundredGigE0/11/0/25.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
.
Router(config-l2vpn-bg-bd)# interface HundredGigE0/11/0/31.101
Router(config-l2vpn-bg-bd-ac)# routed interface BVI101
Router(config-l2vpn-bg-bd-ac)# commit
```

MPLS Static Configuration

- BVI BVI101 is associated with the MPLS static LSP with a destination IP address 209.165.201.1.
- MPLS configurations (such as attaching an incoming label [1000101 and removing the MPLS label at the LSP egress device) are added.

```
Router# configure terminal
Router(config)# mpls static
Router(config-mpls-static)# lsp bvi-101
Router(config-mpls-static-lsp)# in-label 1000101 allocate
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd)# path 1 resolve-nexthop 209.165.201.1 out-label pop
Router(config-mpls-static-lsp-fwd)# commit
```

Traffic Engineering

- Segment Routing (SR)is used for Traffic Engineering (TE).
- For the specified SR policy, the end point (or destination) is 209.165.201.1, same as the MPLS LSP destination for transporting L2VPN traffic.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy sr-static-mpls-to-LER2
Router(config-sr-te-policy)# binding-sid mpls 250100
Router(config-sr-te-policy)# color 100 end-point ipv4 209.165.201.1
Router(config-sr-te-policy)# candidate-paths
Router(config-sr-te-policy-path)# preference 100
Router(config-sr-te-policy-path-pref)# explicit segment-list to-LER2
Router(config-sr-te-policy-path-pref)# commit
```

A segment list is created with MPLS labels.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# segment-list to-LER2
Router(config-sr-te-sl)# index 10 mpls label 1000101
Router(config-sr-te-sl)# index 20 mpls label 300501
Router(config-sr-te-sl)# commit
```

Verification

Use this command to view MPLS static LSP details specific to the BVI:

```
Router# show mpls static lsp bvi-101 detail

LSP Name             Label   VRF             AFI  Type         Prefix           RW Configured
   Status
-------------------- ------- --------------- ---- ------------ ---------------- ------
bvi-101              1000101 default         N/A  X-Connect    N/A              Yes
      Created
  PRIMARY SET:
    [resolve-mode: nexthop 209.165.201.1]
    Path 0 : nexthop Bundle-Ether1 198.51.100.1, out-label Pop, Role: primary, Path-id: 2,
 Status: valid
    Path 1 : nexthop Bundle-Ether2 203.0.113.100, out-label Pop, Role: primary, Path-id:
1, Status: valid
```

Use this command to view MPLS static LSP details specific to the router:

```
Router# show mpls static local-label 1000101 detail

Label   VRF             Type         Prefix           RW Configured   Status
------- --------------- ------------ ---------------- --------------- --------
```

```
1000101 default         X-Connect    N/A              Yes              Created

  PRIMARY SET:
    [resolve-mode: nexthop 209.165.201.1]
    Path 0 : nexthop Bundle-Ether1 198.51.100.1, out-label Pop, Role: primary, Path-id: 2,
 Status: valid
    Path 1 : nexthop Bundle-Ether2 203.0.113.100, out-label Pop, Role: primary, Path-id:
1, Status: valid
```

MPLS Over GRE Tunnels

Table 20: Feature History Table

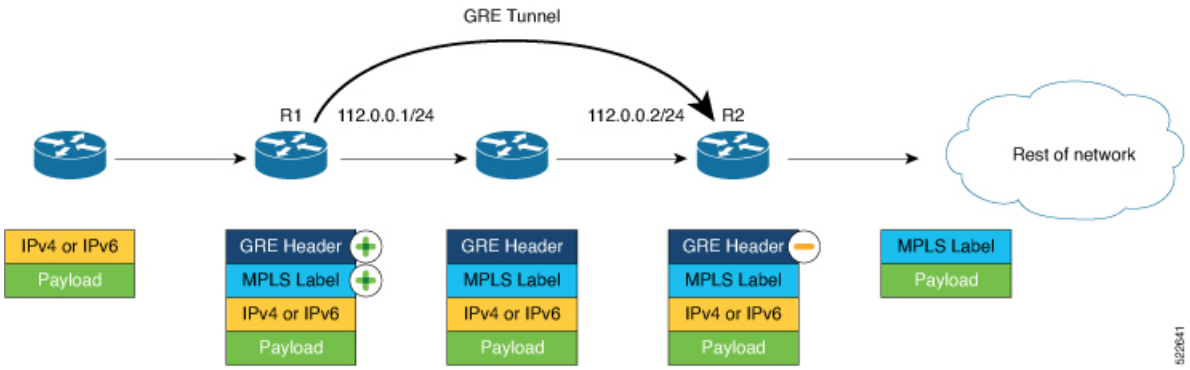
Feature Name	Release Information	Feature Description
MPLS Over GRE Tunnels	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
MPLS Over GRE Tunnels	Release 7.5.2	The MPLS over generic routing encapsulation (MPLSoGRE) provides a mechanism for tunneling MPLS packets over a non-MPLS network. This feature uses MPLS over GRE to encapsulate MPLS packets inside GRE tunnels to create a virtual point-to-point link across non-MPLS networks. With this feature, the core network can be configured with IPv4 addresses to interconnect the MPLS networks through the IP network.

MPLS over GRE supports MPLS static forwarding over a GRE tunnel at line rate, which is the normal speed at which the traffic is sent through networks. For more information on line rate, see the Cisco 8000 Series Routers Data Sheet.

You can configure a provider router to send incoming customer traffic over the GRE tunnel, addressed to a set of load-balancing servers.

The GRE tunnel is configured with encapsulation that adds an MPLS packet and a GRE header to the incoming packet at the starting point of the tunnel. When the packet reaches the endpoint of the GRE tunnel, the GRE header is removed and the payload is forwarded to the destination based on the MPLS label.

Figure 12: MPLS Over GRE Tunnel



In the image, you can see that the GRE tunnel begins at router R1. R1 uses the policy based routing (PBR) process for GRE tunnel encapsulation, adds an MPLS label to the incoming packet, and then adds a GRE header. Then it sends the traffic towards router R2.

R2 uses the PBR process for GRE tunnel decapsulation, and based on the MPLS label, it forwards the traffic towards its destination.

Configuration Example

This example shows how to enable MPLS static forwarding over GRE tunnel.

Configuration on router R1.

1. Configure a tunnel interface.
2. Configure the mode of encapsulation as GRE for the tunnel interface.
3. Configure a policy map and redirect the traffic to next hop IP.
4. Assign the policy map to a VLAN subinterface.
5. Configure MPLS out labels to be applied to the incoming packets.

Configuration on router R2.

1. Configure a tunnel interface.
2. Configure the mode of decapsulation as GRE for the tunnel interface.
3. Configure a class map with match criteria for the source and destination IP addresses.
4. Configure a policy map and specify the class name configured in the class map.
5. Configure GRE decapsulation.

GRE Tunnel Configuration on R1

The GRE tunnel starts on R1.

The GRE tunnel destination must be a valid IPv4 address.

```
Router# configure
Router(config)# interface tunnel-ip1
Router(config-if)# ipv4 address 112.0.0.1 255.255.255.0
```

```
Router(config-if)# tunnel mode gre ipv4 encap
Router(config-if)# tunnel source Loopback 0
Router(config-if)# tunnel destination 50.0.0.1
Router(config-if)# commit
```

PBR Configuration for Encapsulation on R1

GRE encapsulation must be based on a policy map. Configure a policy map and redirect the traffic to next hop IP.

```
Router(config)# policy-map type pbr PBR_ENCAP_1
Router(config-pmap)# class type traffic class-default
Router(config-pmap-c)# redirect ipv4 nexthop 111.0.0.1
Router(config-pmap-c)# end-policy-map
```

Assign the policy map to one or more VLAN subinterfaces.

```
Router(config)# interface Bundle-Ether 111.1
Router(config-if)# service-policy type pbr input PBR_ENCAP_1
Router(config-if)# ipv4 address 50.0.2.1 255.255.255.0
Router(config-if)# encapsulation dot1q 1
```

MPLS Static Configuration on R1

Configure MPLS out labels. Ensure that the out label is the same for all paths of an MPLS static label switch path(LSP). You can configure up to a maximum of 16 paths. After applying the labels, the packets are forwarded to the specified next hop.

```
Router(config)# mpls static
Router(config-mpls-static)# lsp v4-encap-payload-1
Router(config-mpls-static-lsp)# in-label 10001 allocate per-prefix 111.0.0.1/32
Router(config-mpls-static-lsp)# forward
Router(config-mpls-static-lsp-fwd))# path 1 nexthop tunnel-ip1 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 2 nexthop tunnel-ip2 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 3 nexthop tunnel-ip3 out-label 20001
Router(config-mpls-static-lsp-fwd))# path 4 nexthop tunnel-ip4 out-label 20001
Router(config-mpls-static-lsp-fwd)# commit
```

GRE Tunnel Configuration on R2

The GRE tunnel stops on R2.

```
Router # configure
Router(config)# interface tunnel-ip1
Router(config-if)# ipv4 address 112.0.0.2 255.255.255.0
Router(config-if)# tunnel mode gre ipv4 decap
Router(config-if)# tunnel source Loopback 0
Router(config-if)# tunnel destination 10.0.0.1
Router(config-if)# commit
```

PBR Configuration for GRE Tunnel Decapsulation on R2

```
Router(config)# class-map type traffic match-all test_gre1
Router(config-cmap)# match protocol gre
Router(config-cmap)# match destination-address ipv4 50.0.0.1 255.255.255.255
Router(config-cmap)# match source-address ipv4 10.0.0.1 255.255.255.255
Router(config-cmap)# end-class-map
Router(config)# policy-map type pbr P1-test
```

```
Router(config-pmap)# class type traffic test_gre1
Router(config-pmap-c)#decapsulate gre
Router(config-pmap-c)# end-policy-map
Router(config)# vrf-policy vrf default address-family ipv4 policy type pbr input P1-test
```

Running Configuration

Use the following show commands to view the configuration.

Tunnel-IP configuration on R1

```
Router# show running-config interface tunnel-ip 1

interface tunnel-ip1
    ipv4 address 112.0.0.1 255.255.255.0
    tunnel mode gre ipv4 encap
    tunnel source Loopback 0
    tunnel destination 50.0.0.1
!
```

PBR Configuration for GRE Tunnel Encapsulation on R1

```
Router# show running-config policy-map type pbr *

policy-map type pbr PBR_ENCAP_1
 class type traffic class-default
  redirect ipv4 nexthop 111.0.0.1
 !
 end-policy-map
 !
```

MPLS Static Configuration on R1

```
Router# show running-config mpls static

mpls static
  lsp v4-encap-payload-1
     in-label 10001 allocate per-prefix 111.0.0.1/32
     forward
        path 1 nexthop tunnel-ip1 out-label 20001
        path 2 nexthop tunnel-ip2 out-label 20001
        path 3 nexthop tunnel-ip3 out-label 20001
        path 4 nexthop tunnel-ip4 out-label 20001
        !
```

Tunnel-IP Configuration on R2

```
Router# show running-config int tunnel-ip 1

interface tunnel-ip1
 ipv4 address 112.0.0.2 255.255.255.0
 tunnel mode gre ipv4 decap
 tunnel source Loopback 0
 tunnel destination 10.0.0.1
!
```

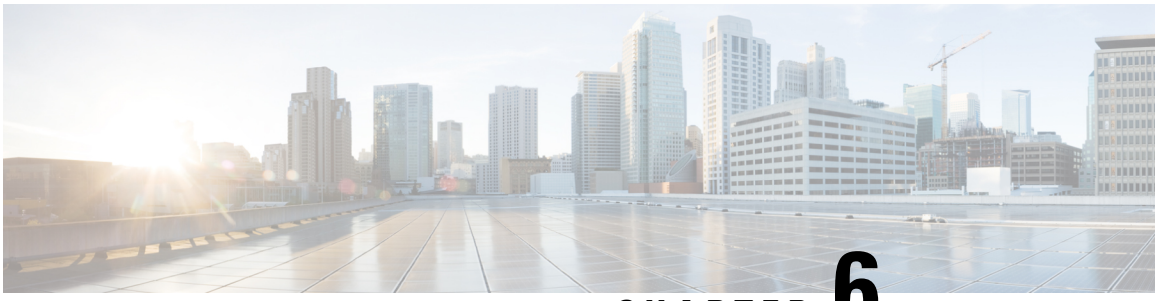
PBR Configuration for GRE Tunnel Decapsulation on R2

```
Router# show running-config class-map type traffic match-all

class-map type traffic match-all test_gre1
```

```
    match protocol gre
    match destination-address ipv4 50.0.0.1 255.255.255.255
    match source-address ipv4 10.0.0.1 255.255.255.255
end-class-map
!
policy-map type pbr P1-test
   class type traffic test_gre1
   decapsulate gre
!
   class type traffic class-default
!
end-policy-map
!

vrf-policy
vrf default address-family ipv4 policy type pbr input P1-test
!
```



CHAPTER 6

Implementing MPLS Traffic Engineering

Table 21: Feature History Table

Feature Name	Release Information	Feature Description
Implementing MPLS traffic engineering	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Implementing MPLS traffic engineering	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Implementing MPLS traffic engineering	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Implementing MPLS traffic engineering	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) You can now enhance network performance by optimizing the routing of data flows across the network. This feature allows precise control over network paths, ensuring efficient use of resources and improved traffic management. MPLS Traffic Engineering supports dynamic rerouting to maintain high availability and reliability, even during network changes or failures. It leverages label-switched paths (LSPs) for directing traffic through non-standard paths, reducing congestion, and balancing loads across the network. This capability is critical for maintaining service quality and optimizing the overall efficiency of network operations. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

Traditional IP routing emphasizes on forwarding traffic to the destination as fast as possible. As a result, the routing protocols find out the least-cost route according to its metric to each destination in the network and every router forwards the packet based on the destination IP address and packets are forwarded hop-by-hop. Thus, traditional IP routing does not consider the available bandwidth of the link. This can cause some links to be over-utilized compared to others and bandwidth is not efficiently utilized. Traffic Engineering (TE) is used when the problems result from inefficient mapping of traffic streams onto the network resources. Traffic engineering allows you to control the path that data packets follow and moves traffic flows from congested links to non-congested links that would not be possible by the automatically computed destination-based shortest path.

Multiprotocol Label Switching (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) relies on the MPLS backbone to replicate and expand upon the TE capabilities of Layer 2 ATM and Frame Relay networks.

MPLS-TE learns the topology and resources available in a network and then maps traffic flows to particular paths based on resource requirements and network resources such as bandwidth. MPLS-TE builds a unidirectional tunnel from a source to a destination in the form of a label switched path (LSP), which is then used to forward traffic. The point where the tunnel begins is called the tunnel headend or tunnel source, and the node where the tunnel ends is called the tunnel tailend or tunnel destination. A router through which the tunnel passes is called the mid-point of the tunnel.

MPLS uses extensions to a link-state based Interior Gateway Protocol (IGP), such as Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF). MPLS calculates TE tunnels at the LSP head based on required and available resources (constraint-based routing). If configured, the IGP automatically routes the traffic onto these LSPs. Typically, a packet that crosses the MPLS-TE backbone travels on a single LSP that connects the ingress point to the egress point. MPLS TE automatically establishes and maintains the LSPs across the MPLS network by using the Resource Reservation Protocol (RSVP).

Note Combination of unlabelled paths protected by labelled paths is not supported.

- Prerequisites for Implementing Cisco MPLS Traffic Engineering, on page 137
- Overview of MPLS-TE Features, on page 137
- How MPLS-TE Works, on page 139
- Soft-preemption, on page 140
- Soft-preemption over FRR Backup Tunnels, on page 141
- SRLG Limitations, on page 141
- RSVP-TE Dark Bandwidth Accounting, on page 142
- Point-to-Multipoint Traffic-Engineering, on page 143
- Configuring MPLS-TE, on page 147
- Regular expressions in MPLS-TE tunnel names, on page 241
- MPLS-TE Features - Details, on page 245
- Configuring Performance Measurement, on page 249
- Additional References, on page 250

Prerequisites for Implementing Cisco MPLS Traffic Engineering

These prerequisites are required to implement MPLS TE:

- Router that runs Cisco IOS XR software .
- Installed composite mini-image and the MPLS package, or a full composite image.
- IGP activated.

Overview of MPLS-TE Features

In MPLS traffic engineering, IGP extensions flood the TE information across the network. Once the IGP distributes the link attributes and bandwidth information, the headend router calculates the best path from head to tail for the MPLS-TE tunnel. This path can also be configured explicitly. Once the path is calculated, RSVP-TE is used to set up the TE LSP (Labeled Switch Path).

To forward the traffic, you can configure autoroute, forward adjacency, or static routing. The autoroute feature announces the routes assigned by the tailend router and its downstream routes to the routing table of the headend router and the tunnel is considered as a directly connected link to the tunnel.

If forward adjacency is enabled, MPLS-TE tunnel is advertised as a link in an IGP network with the link's cost associated with it. Routers outside of the TE domain can see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network.

MPLS-TE provides protection mechanism known as fast reroute to minimize packet loss during a failure. For fast reroute, you need to create back up tunnels. The autotunnel backup feature enables a router to dynamically build backup tunnels when they are needed instead of pre-configuring each backup tunnel and then assign the backup tunnel to the protected interfaces.

DiffServ Aware Traffic Engineering (DS-TE) enables you to configure multiple bandwidth constraints on an MPLS-enabled interface to support various classes of service (CoS). These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

The MPLS traffic engineering auto-tunnel mesh feature allows you to set up full mesh of TE tunnels automatically with a minimal set of MPLS traffic engineering configurations. The MPLS-TE auto bandwidth feature allows you to automatically adjusts bandwidth based on traffic patterns without traffic disruption.

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

For detailed information about MPLS-TE features, see the *MPLS-TE Features - Details* topic.

Note MPLS-TE Nonstop Routing (NSR) is enabled by default without any user configuration and cannot be disabled.

MPLS-TE NSR means the application is in hot-standby mode and standby MPLS-TE instance is ready to take over from the active instance quickly on RP failover.

Note that the MPLS-TE does not do routing. If there is standby card available then the MPLS-TE instance is in a hot-standby position.

The following output shows the status of MPLS-TE NSR:

```
Router#show mpls traffic-eng nsr status

TE Process Role        : V1 Active
Current Status      : Ready
    Ready since        : Tue Nov 01 10:42:34 UTC 2022 (1w3d ago)
   IDT started         : Tue Nov 01 03:28:48 UTC 2022 (1w3d ago)
    IDT ended          : Tue Nov 01 03:28:48 UTC 2022 (1w3d ago)
Previous Status        : Not ready
    Not ready reason   : Collaborator disconnected
    Not ready since    : Tue Nov 01 10:42:34 UTC 2022 (1w3d ago)
```

During any issues with the MPLS-TE, the NSR on the router gets affected which is displayed in the show redundancy output as follows:

```
Router#show mpls traffic-eng nsr status details
.
.
.

Current active rmf state: 4 (I_READY)
All standby not-ready bits clear - standby should be ready

Current active rmf state for NSR: Not ready
<jid> <node> <name> Reason for standby not NSR-ready
1082 0/RP0/CPU0 te_control TE NSR session not synchronized
Not ready set Wed Nov 19 17:28:14 2022: 5 hours, 23 minutes ago
1082 0/RP1/CPU0 te_control Standby not connected
Not ready set Wed Nov 19 17:29:11 2022: 5 hours, 22 minutes ago
```

Note If you have disabled MPLS-TE NSR and configured only MPLS-TE nonstop forwarding (NSF), performing an RP failover results in a verification error causing unprotected LSPs to flap and triggering FRR on protected LSPs. To avoid the verification error, you must block the path verification using the **mpls traffic-eng topology block-verify-after-rpfo** command.

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 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.

The following sections describe the components of MPLS-TE:

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 keep track on topology and resource information to be flooded.

Link-state IGP

Either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF) can be used as IGPs. These IGPs are used to globally flood topology and resource information from the link management module.

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.

Soft-preemption

MPLS-TE preemption helps to free up the resources of an established LSP and assign them to a new LSP. The freeing up 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 the traffic disruption over the preempted LSP.

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.

Table 22: Feature History Table

Feature Name	Release Information	Feature Description
Soft-preemption	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Soft-preemption	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is now supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Soft-preemption	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Soft-preemption	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The soft-preemption feature minimizes or eliminate traffic loss by preempting the LSPs in a graceful manner. However, the link might be over-subscribed for a period of time. *This feature, which was previously supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

Soft-preemption over FRR Backup Tunnels

The soft-preemption over FRR backup tunnels feature enables to move LSP traffic over the backup tunnels when the LSP is soft- preempted. MPLS TE tunnel soft-preemption allows removal of extra TE traffic in a graceful manner, by giving the preempted LSP a grace period to move away from the link. Though this mechanism saves the traffic of the preempted LSP from being dropped, this might cause traffic drops due to congestion as more bandwidth is reserved on the link than what is available. When the soft-preemption over FRR backup tunnel is enabled, the traffic of the preempted LSP is moved onto the FRR backup, if it is available and ready. This way, the capacity of the backup tunnel is used to remove the potential congestion that might be caused by soft-preemption.

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 signaled, they are verified dynamically in the next path verification cycle.

RSVP-TE Dark Bandwidth Accounting

This section describes the RSVP-TE Dark Bandwidth Accounting feature that allows for the co-existence of non-zero bandwidth RSVP-TE tunnels and Segment Routing (SR) in the same network domain. This feature measures dark bandwidth traffic and accounts for it in the RSVP-TE bandwidth reservations to avoid overbooking the links in the network.

Dark bandwidth is the actual utilization of the link by the subset of the traffic that is not explicitly admission controlled by RSVP-TE. Dark bandwidth is not considered during path computation and admission control for distributed RSVP-TE LSPs.

In this solution, SR is assumed to be the main source of dark bandwidth on the links in the network. In addition, SR traffic is considered to have a higher priority than any other traffic transported by RSVP-TE LSPs. Therefore, the bandwidth consumed by SR effectively reduces the link bandwidth available to RSVP-TE LSPs.

The RSVP-TE Dark Bandwidth Accounting feature consists of the following:

- The measurement of SR traffic on interfaces via new per-interface aggregate SR counters
- The calculation of dark bandwidth rate based on the measured SR traffic statistics
- The calculation of the RSVP-TE effective maximum reservable bandwidth (BMRe).

  The BMRe is used for the purpose of pre-emption as well as advertisement (flooding) via IGP. A threshold is evaluated before triggering flooding.

Computing the Dark Bandwidth and RSVP-TE Effective Maximum Reservable Bandwidth

The statistics collector process (statsD) is responsible for returning statistics counters for each feature. For each traffic engineering (TE)-enabled interface, the TE process collects new SR bandwidth rate statistics (samples) from the statsD process, within a specified sampling interval. These samples are collected over a period of time called an application interval.

After each application interval, the average value of the collected rate samples is used to compute the dark bandwidth rate and the BMRe rate.

The following example shows how the BMRe is computed (assuming a link capacity of 10Gbps and a configured BMR [BMRc] of 90%):

- Link capacity = 10Gbps
- BMRc = RSVP percentage of link capacity = 9Gbps
- Calculated dark bandwidth rate = 2Gpbs
- BMRe = 7Gbps

In this example, the bandwidth available for RSVP-TE LSP admission is 7Gbps. This value is flooded in the network if the flooding threshold is crossed.



Note When you change the RSVP bandwidth percentage configuration or when the bundle capacity changes due to bundle-member state change, TE accounts for the dark bandwidth when new bandwidth values are advertised.

Note The measured dark bandwidth can be increased or decreased based on a configurable adjustment factor.

When the dark bandwidth rate increases for a link, it will lower the BMR of that link, which might trigger preemption of the RSVP-TE LSPs. 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.

Point-to-Multipoint Traffic-Engineering

This section contains the following topics:

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 network.

Note

- For P2MP tunnels, a Cisco 8000 Series router supports the mid-point router function, and does not support source or receiver functions. To know how to configure a source or receiver (destination) router in a P2MP tunnel, refer the MPLS configuration guide for the corresponding platform.
- The FRR function is not supported for P2MP tunnels.

By using RSVP-TE extensions as defined in RFC 4875, multiple subLSPs are signaled for a given TE source. The P2MP tunnel is considered as a set of Source-to-Leaf (S2L) subLSPs that connect the TE source to multiple leaf Provider Edge (PE) nodes.

At the TE source, the ingress point of the P2MP-TE tunnel, IP multicast traffic is encapsulated with a unique MPLS label, which is associated with the P2MP-TE tunnel. The traffic continues to be label-switched in the P2MP tree. If needed, the labeled packet is replicated at branch nodes along the P2MP tree. When the labeled packet reaches the egress leaf (PE) node, the MPLS label is removed and forwarded onto the IP multicast tree across the PE-CE link.

To enable end-to-end IP multicast connectivity, RSVP is used in the MPLS-core for P2MP-TE signaling and PIM is used for PE-CE link signaling.

- All edge routers are running PIM-SSM or Source-Specific Multicast (SSM) to exchange multicast routing information with the directly-connected Customer Edge (CE) routers.
- In the MPLS network, RSVP P2MP-TE replaces PIM as the tree building mechanism, RSVP-TE grafts or prunes a given P2MP tree when the end-points are added or removed in the TE source configuration (explicit user operation).

These are the definitions for Point-to-Multipoint (P2MP) tunnels. Cisco 8000 Series routers only support the role of a mid-point.

Source

Configures the node in which Label Switched Path (LSP) signaling is initiated.

Mid-point

Specifies the transit node in which LSP signaling is processed (for example, not a source or receiver).

Receiver, Leaf, and Destination

Specifies the node in which LSP signaling ends.

Branch Point

Specifies the node in which packet replication is performed.

Source-to-Leaf (S2L) SubLSP

Specifies the P2MP-TE LSP segment that runs from the source to one leaf.

Point-to-Multipoint Traffic-Engineering Features

- P2MP RSVP-TE (RFC 4875) is supported. RFC 4875 is based on nonaggregate signaling; for example, per S2L signaling. Only P2MP LSP is supported.
- The **interface tunnel-mte** command identifies the P2MP interface type.
- P2MP tunnel setup is supported with label replication.
- 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.

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.

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.

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.

Point-to-Multipoint Label Switch Path

The Point-to-Multipoint Label Switch Path (P2MP LSP) has only a single root, which is the Ingress Label Switch Router (LSR). The P2MP LSP is created based on a receiver that is connected to the Egress LSR. The Egress LSR initiates the creation of the tree (for example, tunnel grafting or pruning is done by performing an individual sub-LSP operation) by creating the Forwarding Equivalency Class (FEC) and Opaque Value.



Note Grafting and pruning operate on a per destination basis.

The Opaque Value contains the stream information that uniquely identifies the tree to the root. To receive label switched multicast packets, the Egress Provider Edge (PE) indicates to the upstream router (the next hop closest to the root) which label it uses for the multicast source by applying the label mapping message.

The upstream router does not need to have any knowledge of the source; it needs only the received FEC to identify the correct P2MP LSP. If the upstream router does not have any FEC state, it creates it and installs the assigned downstream outgoing label into the label forwarding table. If the upstream router is not the root of the tree, it must forward the label mapping message to the next hop upstream. This process is repeated hop-by-hop until the root is reached.

By using downstream allocation, the router that wants to receive the multicast traffic assigns the label for it. The label request, which is sent to the upstream router, is similar to an unsolicited label mapping (that is, the upstream does not request it). The upstream router that receives that label mapping uses the specific label to send multicast packets downstream to the receiver. The advantage is that the router, which allocates the labels, does not get into a situation where it has the same label for two different multicast sources. This is because it manages its own label space allocation locally.

Path Option for Point-to-Multipoint RSVP-TE

P2MP tunnels are signaled by using the dynamic and explicit path-options in an IGP intra area. InterArea cases for P2MP tunnels are signaled by the verbatim path option.

Path options for P2MP tunnels are individually configured for each sub-LSP. Only one path option per sub-LSP (destination) is allowed. You can choose whether the corresponding sub-LSP is dynamically or explicitly routed. For the explicit option, you can configure the verbatim path option to bypass the topology database lookup and verification for the specified destination.

Both dynamic and explicit path options are supported on a per destination basis by using the **path-option (P2MP-TE)** command. In addition, you can combine both path options.

Explicit Path Option

Configures the intermediate hops that are traversed by a sub-LSP going from the TE source to the egress MPLS node. Although an explicit path configuration enables granular control 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.

Point-to-Multipoint Implicit Null

The Point-to-Multipoint (P2MP) implicit null feature enables the forwarding of unicast traffic over P2MP tunnels. This feature is enabled by default and requires no configuration.

In a P2MP tunnel, the tailend router signals the implicit null label to the midpoint router. If the given MPI leg of the P2MP tunnel is implicit null capable (where the penultimate router is capable to do penultimate hop popping), the FIB (Forwarding Information Base) creates two NRLDI (Non Recursive Load Distribution Index) entries, one for forwarding the IPv6 labeled packets, and the other for non-labeled IPv4 unicast traffic.

The headend and the tailend routers handle the unicast traffic arriving on the P2MP tunnel. The midpoint router forwards the unicast traffic to its bud and tailend routers.

The use of implicit null at the end of a tunnel is called penultimate hop popping (PHP). The FIB entry for the tunnel on the PHP router shows a "pop label" as the outgoing label.

In some cases, it could be that the packets have two or three or more labels in the label stack. Then the implicit null label used at the tailend router would signal the penultimate hop router to pop one label and send the labeled packet with one label less to the tailend router. Then the tailend router does not have to perform two label lookups. The use of the implicit null label does not mean that all labels of the label stack must be removed; only one label is "popped" off (remove the top label on the stack). In any case, the use of the implicit null label prevents the tailend router from performing two lookups.

Restriction - The P2MP implicit null feature may cause multicast traffic drop with implicit null label on the tailend routers. This is because the P2MP implicit null feature does not support forwarding of multicast traffic when no label is received on the tailend router.

Configuring MPLS-TE

MPLS-TE requires co-ordination among several global neighbor routers. RSVP, MPLS-TE and IGP are configured on all routers and interfaces in the MPLS traffic engineering network. Explicit path and TE tunnel interfaces are configured only on the head-end routers. MPLS-TE requires some basic configuration tasks explained in this section.

Building MPLS-TE Topology

Building MPLS-TE topology, sets up the environment for creating MPLS-TE tunnels. This procedure includes the basic node and interface configuration for enabling MPLS-TE. To perform constraint-based routing, you need to enable OSPF or IS-IS as IGP extension.

Before You Begin

Before you start to build the MPLS-TE topology, the following pre-requisites are required:

- 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.
- Enable RSVP on the port interface.

Example

This example enables MPLS-TE on a node and then specifies the interface that is part of the MPLS-TE. Here, OSPF is used as the IGP extension protocol for information distribution.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface hundredGigE 0/0/0/3
Router(config)# router ospf area 1
Router(config-ospf)# area 0
Router(config-ospf-ar)# mpls traffic-eng
Router(config-ospf-ar)# interface hundredGigE 0/0/0/3
Router(config-ospf-ar-if)# exit
Router(config-ospf)# mpls traffic-eng router-id 192.168.70.1
Router(config)# commit
```

Example

This example enables MPLS-TE on a node and then specifies the interface that is part of the MPLS-TE. Here, IS-IS is used as the IGP extension protocol for information distribution.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface hundredGigE 0/0/0/3
Router(config)# router isis 1
Router(config-isis)# net 47.0001.0000.0000.0002.00
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# mpls traffic-eng level 1
Router(config-isis-af)# exit
Router(config-isis)# interface hundredGigE 0/0/0/3
Router(config-isis-if)# exit
Router(config)# commit
```

Teardown and reestablishment of RSVP-TE tunnels

Table 23: Feature History Table

Feature Name	Release Information	Feature Description
Teardown and reestablishment of RSVP-TE tunnels	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Teardown and reestablishment of RSVP-TE tunnels	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Teardown and reestablishment of RSVP-TE tunnels	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
Teardown and reestablishment of RSVP-TE tunnels	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Teardown and reestablishment of RSVP-TE tunnels	Release 7.11.1	You can now teardown and reestablish the existing tunnels of headend, midend, or tailend router tunnels of an MPLS network for optimized distribution of the traffic across MPLS and RSVP-TE to improve network performance and enhance resource utilization. Previously, you could reestablish tunnels only at the headend router using the mpls traffic-eng-resetup command. The feature introduces these changes: **CLI**: mpls traffic-eng teardown **YANG Data Model**: Cisco-IOS-XR-mpls-te-act.yang (see GitHub, YANG Data Models Navigator)

In an MPLS-TE network which is configured with RSVP-TE, the headend, midend, and tailend router work together to establish and maintain tunnels or adjacencies for traffic engineering purposes. When the headend router boots up, it plays a critical role in MPLS-TE tunnel establishment using RSVP-TE signaling along the computed path with the tailend router.

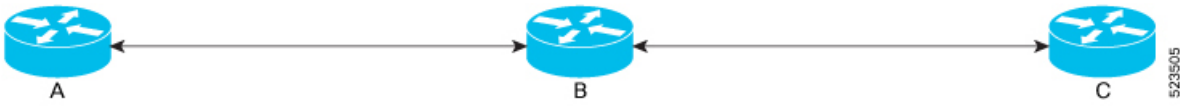
During a system reboot, if a tailend router comes up first, the Interior Gateway Protocol (IGP) adjacency establishes a path for the tailend with the adjacent network devices. However, if the headend router comes up and starts creating tunnels during this tailend path creation, it may result in a poor distribution of tunnels at the tailend. In such cases, it's necessary to tear down all the tunnels and recreate them to ensure optimal distribution and functioning of the RSVP-TE as per the network conditions.

From Release 7.11.1, you can teardown and reestablish tunnels across all headend, midend, and tailend routers using the **mpls traffic-eng teardown** command with **all, head, mid, tail** parameters.

Table 24: Network Routers and Description

Router Name	Router Function	Tunnel Tear-Down and Reestablish Command
Headend Router	The headend router is responsible for determining the tunnel's path. It initiates the signaling process, specifying the tunnel's parameters and requirements. It sets up and manages the MPLS-TE tunnels.	**mpls traffic-eng teardown head**
Midend Router	The midend router, or an intermediate router or network device, is located between the headend and the tailend along the tunnel's path. It's involved in the signaling process and ensures the proper forwarding of traffic along the established tunnel.	**mpls traffic-eng teardown mid**
Tailend Router	The tailend router is the terminating router or network device located at the other end of the tunnel. It receives and processes the traffic that traverses through the tunnel and ensures proper traffic delivery to the destination.	**mpls traffic-eng teardown tail**

Figure 13: RSVP-TE tunnel Teardown and Reestablishment Topology



When RSVP-TE tunnel teardown message is triggered and:

- Router B is configured as tailend and Router A is configured as headend: A **ResvTear upstream** message is sent to the headend router. This message informs the headend router to tear down the RSVP-TE tunnel and release the resources associated with the tunnel.
- Router B is configured as headend and Router C as tailend: A **PathTear downstream** message is sent to the tailend router. A headend router triggers the process of recomputing the tunnels. The headend router (Router B) initiates the process of recomputing the tunnels, which involves recalculating the path and parameters for establishing new tunnels.
- Router B is configured as midend where Router A and Router B as headend and tailend respectively: A **ResvTear upstream** message is sent to the headend router and a **PathTear downstream** message is sent to the tailend router. These messages inform the respective routers to tear down the RSVP-TE tunnel and release the associated resources.

You can tear down and set up all types of tunnels including, P2P, P2MP, numbered, named, flex LSPs, and auto tunnels.

Limitations

- A maximum 90 seconds are required for the tunnels to get reestablished once they are torn.
- Use the **mpls traffic-eng resetup** command to reestablish the tunnels only at the headend router.

Configure Tear down and Reestablishment of RSVP-TE Tunnels

Configuration Example

Use the **mpls traffic-eng teardown all** command to tear down and reestablish all the RSVP-TE tunnels in a network node. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown all
```

Use the **mpls traffic-eng teardown head** command to tear down and reestablish the RSVP-TE tunnels at the headend router. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown head
```

Note You can also use the **mpls traffic-eng resetup** command to reestablish tunnels only at the headend router in XR EXEC mode.

Use the **mpls traffic-eng teardown mid** command to tear down and reestablish the RSVP-TE tunnels at the midend router. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown mid
```

Use the **mpls traffic-eng teardown tail** command to tear down and reestablish the RSVP-TE tunnels at the tailend router. This command must be executed in XR EXEC mode.

```
Router# mpls traffic-eng teardown tail
```

Use the **show mpls traffic-eng tunnels summary** command to check RSVP-TE tunnel status after you run the teardown command.

```
Router# show mpls traffic-eng tunnels summary
Output received:
Thu Sep 14 10:48:45.007 UTC

        Path Selection Tiebreaker:  Min-fill (default)
              LSP Tunnels Process:  running
                     RSVP Process:  running
                       Forwarding:  enabled
          Periodic reoptimization:  every 3600 seconds, next in 2806 seconds
           Periodic FRR Promotion:  every 300 seconds, next in 9 seconds
      Periodic auto-bw collection:  5 minute(s) (disabled)

Signalling Summary:
    Head: 14006 interfaces, 14006 active signalling attempts, 14006 established
          14006 explicit, 0 dynamic
          14006 activations, 0 deactivations
          0 recovering, 0 recovered
    Mids: 2000
    Tails: 4003

Fast ReRoute Summary:
    Head:     14000 FRR tunnels, 14000 protected, 0 rerouted
    Mid:      2000 FRR tunnels, 2000 protected, 0 rerouted
    Summary:  16000 protected, 13500 link protected, 2500 node protected, 0 bw protected
    Backup:   6 tunnels, 4 assigned
    Interface: 10 protected, 0 rerouted

Bidirectional Tunnel Summary:
    Tunnel Head: 0 total, 0 connected, 0 associated, 0 co-routed
    Tunnel Tail: 0 total, 0 connected, 0 associated, 0 co-routed
```

```
LSPs Head:   0 established, 0 proceeding, 0 associated, 0 standby
LSPs Mid:    0 established, 0 proceeding, 0 associated, 0 standby
LSPs Tail:   0 established, 0 proceeding, 0 associated, 0 standby
```

Configuring Automatic Bandwidth

Automatic bandwidth allows you to dynamically adjust bandwidth reservation based on measured traffic. MPLS-TE automatic bandwidth monitors the traffic rate on a tunnel interface and resizes the bandwidth on the tunnel interface to align it closely with the traffic in the tunnel. MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every headend router.

Adjustment Threshold - It 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 signaled 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.

The following table specifies the parameters that can be configured as part of automatic bandwidth configuration.

Table 25: Automatic Bandwidth Parameters

Bandwidth Parameters	Description
Application frequency	Configures how often the tunnel bandwidths changed for each tunnel. The default value is 24 hours.
Bandwidth limit	Configures the minimum and maximum automatic bandwidth to set on a tunnel.
Bandwidth collection frequency	Enables bandwidth collection without adjusting the automatic bandwidth. The default value is 5 minutes.
Overflow threshold	Configures tunnel overflow detection.
Adjustment threshold	Configures the tunnel-bandwidth change threshold to trigger an adjustment.

Adjustment Threshold

Configuration Example

This example enables automatic bandwidth on MPLS-TE tunnel interface and configure the following automatic bandwidth variables.

- Application frequency
- Bandwidth limit

- Adjustment threshold
- Overflow detection

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# auto-bw
Router(config-if-tunte-autobw)# application 1000
Router(config-if-tunte-autobw)# bw-limit min 30 max 1000
Router(config-if-tunte-autobw)# adjustment-threshold 50 min 800
Router(config-if-tunte-autobw)# overflow threshold 100 limit 1
Router(config)# commit
```

Verification

Verify the automatic bandwidth configuration using the **show mpls traffic-eng tunnels auto-bw brief** command.

```
Router# show mpls traffic-eng tunnels auto-bw brief

  Tunnel    LSP  Last appl  Requested  Signalled    Highest    Application
  Name       ID   BW(kbps)   BW(kbps)   BW(kbps)   BW(kbps)      Time Left
  -------------- ------ ---------- ---------- ---------- ---------- --------------
  tunnel-te1      5          500         300        420          1h 10m
```

Configuring Automatic Capacity With Load-Interval Configuration

Table 26: Feature History Table

Feature Name	Release Information	Feature Description
Configuring Automatic Capacity With Load-Interval Configuration	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Configuring Automatic Capacity With Load-Interval Configuration	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
Configuring Automatic Capacity With Load-Interval Configuration	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Configuring Automatic Capacity With Load-Interval Configuration	Release 7.3.3	With this feature, you can enable the load-interval configuration for a main tunnel's clones, along with the automatic capacity feature.

The auto-bandwidth feature resizes MPLS-TE tunnels based on traffic loads. Multiple auto-bandwidth tunnels can be created for balancing traffic loads and redundancy.

The auto-capacity feature is an extension of the auto-bandwidth feature. With auto-capacity, for an auto-bandwidth enabled MPLS-TE tunnel, you can enable automatic creation and deletion of tunnels based on real-time capacity demands. These tunnels are called *clones*. For a main TE tunnel, you can specify the minimum and maximum number of clones, and allocate a nominal tunnel bandwidth value. Clones are automatically added to, or removed from, the main TE tunnel, based on the nominal bandwidth.

Consider the auto-capacity configuration example:

```
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels tunnel-te YOW2YZZ


Router(config-te-tun-name)# load-interval 90
Router(config-te-tun-name)# auto-bw
Router(config-mpls-te-tun-autobw)# auto-capacity
Router(config-mpls-te-tun-autobw)# commit
```

retries to establish the LSPs. The timeout range is 1 to 600 seconds.

The auto-capacity function is disabled by default. Since auto-bandwidth and auto-capacity functions are inter-related, these are the corresponding changes in behavior:

- When you enable auto-capacity, it is associated with a specific TE-tunnel, under the auto-bandwidth function. When you disable auto-bandwidth, auto-capacity is also disabled.
- If the load interval is enabled for the main tunnel, it is automatically applied to its clones too. For a main tunnel, if the auto-capacity feature is enabled but a load interval is not enabled, the clones' load interval value is set to a default of 300 seconds.

Splitting and Merging Tunnels

When there is a change in demand for bandwidth, MPLS-TE adds or reduces the number of tunnels and resizes the bandwidth of all the tunnels. It verifies these rules during this activity.

1. The number of tunnels between the headend-tailend router pair is within the specified range, and the bandwidth per tunnel is within the auto-bandwidth range.
2. The (*Bandwidth-per-tunnel*)*(*Number-of-Tunnels*) >= *Total-tunnel-bandwidth* requirement.

   While Rule 1 is enforced, MPLS-TE attempts to enforce Rule 2.
3. When the split requirement is met, and the maximum number of clones is not reached, at least one extra clone is added.

   When the merge requirement is met, and the minimum number of clones is not reached, at least one clone is removed
4. The nominal bandwidth value is used to balance the requirements of: (a) Number of tunnels and (b) Bandwidth for each tunnel. This helps in avoiding a merging or splitting instance at the next application event.

Configurations

```
/* Automatic Capacity Function */

Router# configure
Router(config)# mpls traffic-eng
```

The auto-capacity feature is only valid for the main tunnel **YOW2YZZ**, with reference to which, clones are created or removed.

```
Router(config-mpls-te)# named-tunnels tunnel-te YOW2YZZ
Router(config-te-tun-name)# auto-bw auto-capacity
```

MPLS-TE maintains the number of clones between 1 and 7. Including the main tunnel **YOW2YZZ**, the tunnel count range is between 2 and 8.

```
Router(config-te-tun-autocapacity)# max-clones 7
Router(config-te-tun-autocapacity)# min-clones 1
```

The nominal-bandwidth option is used for specifying the target bandwidth based on which MPLS-TE calculates the number of required tunnels.

```
Router(config-te-tun-autocapacity)# nominal-bandwidth 2000000
```

MPLS-TE also uses the merge-bandwidth and split-bandwidth values when implementing the auto-capacity feature.

```
Router(config-te-tun-autocapacity)# merge-bandwidth 1000000
Router(config-te-tun-autocapacity)# split-bandwidth 3000000
Router(config-te-tun-autocapacity)# commit
```

Verification

```
/* View the Auto-Capacity Feature Configuration */

Router# show mpls traffic-eng tunnels name YOW2YZZ

Name: YOW2YZZ     Ifhandle:0xf000014
  ..
  Config Parameters:
```

```
  ..
    Load-interval: 300 seconds  ..
Auto-Capacity: Enabled
  Minimum Clones: 1; Maximium Clones: 7
  Nominal BW: 2000000 kbps; Merge BW: 1000000 kbps; Split BW: 3000000 kbps
  Statistics:
    Splits: 0; Merges: 0
    Clones Created: 1; Clones Deleted: 0
    Clones High Watermark: 1
  Number of clones: 1
    Clone: YOW2YZZ-1
      Created: Thu Jan 27 13:55:06 2022; State: down
  ..
```

Configuring Auto-Bandwidth Bundle TE++

Table 27: Feature History Table

Feature Name	Release Information	Feature Description
MPLS-TE++ Support Over Numbered Tunnel-TE Interfaces	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
MPLS-TE++ Support Over Numbered Tunnel-TE Interfaces	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
MPLS-TE++ Support Over Numbered Tunnel-TE Interfaces	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
MPLS-TE++ Support Over Numbered Tunnel-TE Interfaces	Release 7.10.1	We have optimized network performance and enabled efficient utilization of resources for numbered tunnels based on real-time traffic by automatically adding or removing tunnels between two endpoints. This is made possible because this release introduces support for auto-bandwidth TE++ for numbered tunnels, expanding upon the previous support for only named tunnels, letting you define explicit paths and allocate the bandwidth to each tunnel. The feature introduces these changes: • **CLI:** The **auto-capacity** keyword is added to the **interface tunnel-te** command. • **YANG Data Model:** New XPaths for `Cisco-IOS-XR-mpls-te-cfg.yang` (see GitHub, YANG Data Models Navigator)

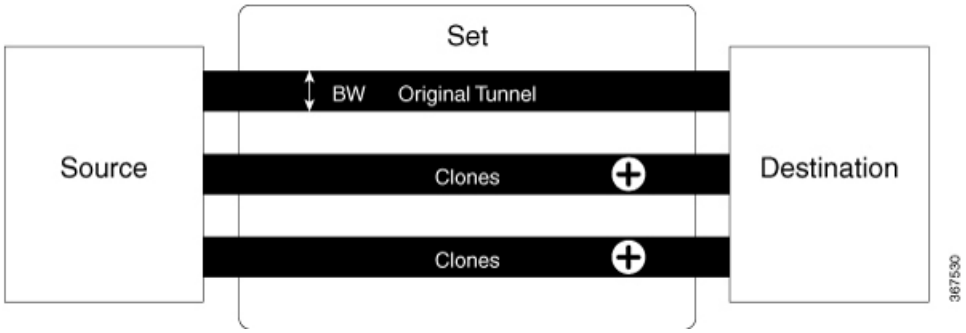
An MPLS-TE tunnel sets up labeled connectivity and provides dynamic bandwidth capacity between its endpoints. The auto-bandwidth function addresses the dynamic bandwidth capacity demands by resizing the

MPLS-TE tunnels based on the measured traffic loads. However, many customers require multiple auto-bandwidth tunnels between two endpoints for load balancing and redundancy. The auto-bandwidth bundle TE++ function is an extension of the auto-bandwidth feature, and provides this support. When the aggregate bandwidth between the endpoints changes, MPLS-TE creates new tunnels or removes existing tunnels to load balance the traffic.

When MPLS-TE automatically creates new tunnels to meet increasing bandwidth demands, they are called clones. The original tunnel and its clones collectively form a *set*. The clones inherit the properties of the main tunnel.You can specify an upper limit and lower limit on the number of clones.

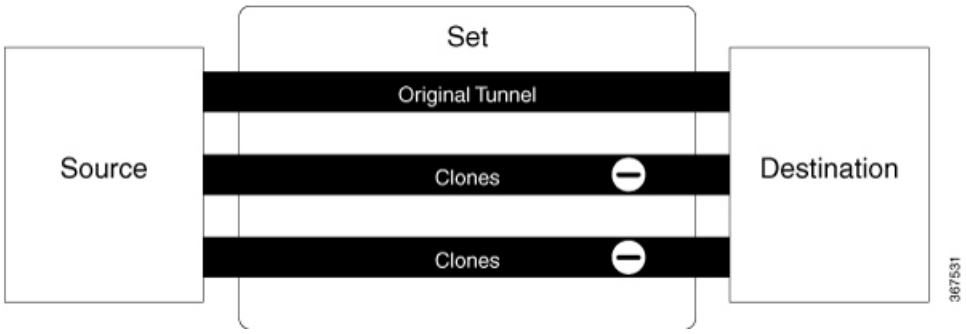
Splitting is the process of creating a new clone. When the bandwidth of a tunnel crosses the split bandwidth value, MPLS-TE creates a clone.

The following figure explains how MPLS-TE creates clones when the split bandwidth exceeds the configured value.



Merging is the process of removing a clone. If the bandwidth goes below the merge bandwidth value in a set of tunnels, MPLS-TE removes a clone.

The following figure explains how MPLS-TE removes clones when the bandwidth falls below the merge bandwidth value.



There are multiple ways to *load-share* the aggregate bandwidth demand among the tunnels in a set. An algorithm chooses the pair that satisfies the aggregate bandwidth requirements. You can configure a nominal bandwidth to guide the algorithm that determines the average bandwidth of the tunnels. If you don't configure, MPLS-TE uses the average of the split bandwidth and merge bandwidth values as the nominal bandwidth.

Restrictions and Guidelines

The following guidelines and restrictions apply for the auto-bandwidth bundle TE++ feature.

- The range for the lower limit on the number of clones is 0–63. The default value is 0. The upper limit range is 1–63. The default value is 63.

Configure Auto-Bandwidth Bundle TE++

Configure the following parameters:

- **min-clones**: Specifies the minimum number of clones that the original tunnel can create.
- **max-clones**: Specifies the maximum number of clones that the original tunnel can create.
- **nominal-bandwidth**: Specifies the average bandwidth for computing the number of tunnels to satisfy the overall demand.
- **split-bandwidth**: Specifies the bandwidth for splitting the original tunnel. If the tunnel bandwidth exceeds the configured split bandwidth, MPLS-TE creates clones.
- **merge-bandwidth**: Specifies the bandwidth for merging clones with the original tunnel. If the bandwidth goes below the merge bandwidth value, MPLS-TE removes the clones.

Configuration Example: Named MPLS-TE Tunnel

This example shows how to configure the auto-bandwidth bundle TE++ feature for a named MPLS-TE tunnel.

Here, the lower and upper limits on the number of clones are two and four, respectively. The bandwidth size for splitting and merging are 200 kbps and 100 kbps, respectively.

```
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels
Router(config-te-named-tunnels)# tunnel-te xyz
Router(config-te-tun-name)# auto-bw
Router(config-mpls-te-tun-autobw)# auto-capacity
Router(config-te-tun-autocapacity)# min-clones 2
Router(config-te-tun-autocapacity)# max-clones 4
Router(config-te-tun-autocapacity)# nominal-bandwidth 150
Router(config-te-tun-autocapacity)# split-bandwidth 200
Router(config-te-tun-autocapacity)# merge-bandwidth 100
```

Configuration Example: Numbered TE-Tunnel

This example shows how to configure the auto-bandwidth bundle TE++ feature for a numbered te-tunnel. The lower limit and the upper limit of clones are 3 and 10, respectively. The bandwidth size for splitting and merging are 3,000,000 kbps and 1,000,000 kbps, respectively.

```
Router(config)#interface tunnel-te 20
Router(config-if)#load-interval 90
Router(config-if)#auto-bw auto-capacity
Router(config-if-tunte-autocapacity)#max-clones 10
Router(config-if-tunte-autocapacity)#min-clones 3
Router(config-if-tunte-autocapacity)#nominal-bandwidth 2000000
Router(config-if-tunte-autocapacity)#merge-bandwidth 1000000
Router(config-if-tunte-autocapacity)#split-bandwidth 3000000
Router(config-if-tunte-autocapacity)#commit
```

Running Configuration

```
Router# show run interface tunnel-te20

interface tunnel-te20
 auto-bw
  auto-capacity
   merge-bandwidth 1000000
   split-bandwidth 3000000
   max-clones 10
   min-clones 3
   nominal-bandwidth 2000000
```

Verification

Here, you can verify that MPLS-TE has created the te-tunnel **tunnel-te20** and three clones.

```
Router# show mpls traffic-eng tunnels name tunnel-te20 detail

Name: tunnel-te20     Ifhandle:0xf00002c
  Signalled-Name: ios_t20
  Status:
    Admin:    up Oper: down   Path: not valid   Signalling: Down
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Wed May 31 11:45:58 2023 (00:04:35 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Delay-limit: disabled
    Delay-measurement: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (not enabled)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Load-interval: 300 seconds
    Auto-bw: enabled
      No BW Applied
      Currently no BW application allowed because: Tunnel is down
      Bandwidth Min/Max: 0-4294967295 kbps
      Application Frequency: 1440 min   Jitter: 0s   Time Left: 23h 55m 30s
      Collection Frequency: 5 min
      Samples Collected: 0   Next: 2m 5s
      Highest BW: 0 kbps   Underflow BW: 0 kbps
      Adjustment Threshold: 10%   10 kbps
      Overflow Detection disabled
      Underflow Detection disabled
      Resignal Last-bandwidth Disabled
    Auto-Capacity: Enabled
      Minimum Clones: 3; Maximium Clones: 10
      Nominal BW: 200000 kbps; Merge BW: 1000000 kbps; Split BW: 3000000 kbps
      Statistics:
        Splits: 0; Merges: 0
        Clones Created: 3; Clones Deleted: 0
        Clones High Watermark: 3
      Number of clones: 3
        Clone: tunnel-te20-1
          Created: Wed May 31 11:45:58 2023; State: down
          Signaled BW: 0 kbps; Current BW Demand: 0 kbps
        Clone: tunnel-te20-2
          Created: Wed May 31 11:45:58 2023; State: down
          Signaled BW: 0 kbps; Current BW Demand: 0 kbps
        Clone: tunnel-te20-3
          Created: Wed May 31 11:45:58 2023; State: down
          Signaled BW: 0 kbps; Current BW Demand: 0 kbps
    Self-ping: Disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
```

```
  Reason for the tunnel being down: No destination is configured
  SNMP Index: 105
  Binding SID: None
  Persistent Forwarding Statistics:
    Out Bytes: 0
    Out Packets: 0

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

Configuring Auto-Tunnel Backup

The MPLS Traffic Engineering Auto-Tunnel Backup feature enables a router to dynamically build backup tunnels on the interfaces that are configured with MPLS TE tunnels instead of building MPLS-TE tunnels statically.

The MPLS-TE Auto-Tunnel Backup feature has these benefits:

- Backup tunnels are built automatically, eliminating the need for users to pre-configure 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.

The TE attribute-set template that specifies a set of TE tunnel attributes, is locally configured at the headend of auto-tunnels. The control plane triggers the automatic provisioning of a corresponding TE tunnel, whose characteristics are specified in the respective attribute-set.

Configuration Example

This example configures Auto-Tunnel backup on an interface and specifies the attribute-set template for the auto tunnels. In this example, unused backup tunnels are removed every 20 minutes using a timer and also the range of tunnel interface numbers are specified.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# auto-tunnel backup
Router(config-mpls-te-if-auto-backup)# attribute-set ab
Router(config-mpls-te)# auto-tunnel backup timers removal unused 20
Router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
Router(config-mpls-te)# commit
```

Verification

This example shows a sample output for automatic backup tunnel configuration.

```
Router# show mpls traffic-eng tunnels brief

              TUNNEL NAME          DESTINATION       STATUS  STATE
               tunnel-te0            200.0.0.3          up  up
               tunnel-te1            200.0.0.3          up  up
               tunnel-te2            200.0.0.3          up  up
               tunnel-te50           200.0.0.3          up  up
             *tunnel-te60          200.0.0.3           up  up
                *tunnel-te70         200.0.0.3           up  up
                *tunnel-te80         200.0.0.3           up  up
```

Removing an AutoTunnel Backup

To remove all the backup autotunnels, perform this task.

Configuration Example

```
Router# clear mpls traffic-eng auto-tunnel backup unused all
```

Verification

Use the **show mpls traffic-eng auto-tunnel summary** command to verify MPLS-TE autotunnel information, including the ones removed.

Configuring Auto-Tunnel Mesh

The MPLS-TE auto-tunnel mesh (auto-mesh) feature allows you to set up full mesh of TE Point-to-Point (P2P) tunnels automatically with a minimal set of MPLS traffic engineering configurations. You can configure one or more mesh-groups and 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 can configure MPLS-TE auto-mesh type attribute-sets (templates) and associate them to mesh-groups. Label Switching Routers (LSRs) can create tunnels using the tunnel properties defined in this attribute-set.

Auto-Tunnel mesh configuration minimizes the initial configuration of the network. You can configure tunnel properties template and mesh-groups or destination-lists on TE LSRs that further creates full mesh of TE tunnels between those LSRs. 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.

Configuration Example

This example configures an auto-tunnel mesh group and specifies the attributes for the tunnels in the mesh-group.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# auto-tunnel mesh
Router(config-mpls-te-auto-mesh)# tunnel-id min 1000 max 2000
Router(config-mpls-te-auto-mesh)# group 10
Router(config-mpls-te-auto-mesh-group)# attribute-set 10
Router(config-mpls-te-auto-mesh-group)# destination-list dl-65
Router(config-mpls-te)# attribute-set auto-mesh 10
Router(config-mpls-te-attribute-set)# autoroute announce
Router(config-mpls-te-attribute-set)# auto-bw collect-bw-only
Router(config)# commit
```

Verification

Verify the auto-tunnel mesh configuration using the **show mpls traffic-eng auto-tunnel mesh** command.

```
Router# show mpls traffic-eng auto-tunnel mesh

Auto-tunnel Mesh Global Configuration:
  Unused removal timeout: 1h 0m 0s
  Configured tunnel number range: 1000-2000

Auto-tunnel Mesh Groups Summary:
  Mesh Groups count: 1
  Mesh Groups Destinations count: 3
```

```
  Mesh Groups Tunnels count:
    3 created, 3 up, 0 down, 0 FRR enabled

Mesh Group: 10 (3 Destinations)
  Status: Enabled
  Attribute-set: 10
  Destination-list: dl-65 (Not a prefix-list)
  Recreate timer: Not running
       Destination    Tunnel ID    State  Unused timer
  ----------------  -----------  -------  ------------
       192.168.0.2         1000     up   Not running
       192.168.0.3         1001     up   Not running
       192.168.0.4         1002     up   Not running
  Displayed 3 tunnels, 3 up, 0 down, 0 FRR enabled

Auto-mesh Cumulative Counters:
  Last cleared: Wed Oct  3 12:56:37 2015 (02:39:07 ago)
                      Total
  Created:                3
  Connected:              0
  Removed (unused):       0
  Removed (in use):       0
  Range exceeded:         0
```

Enable LDP over TE Automatically on Cloned Tunnels

Table 28: Feature History Table

Feature Name	Release Information	Feature Description
Enable LDP over TE Automatically on Cloned Tunnels	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Enable LDP over TE Automatically on Cloned Tunnels	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
Enable LDP over TE Automatically on Cloned Tunnels	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Enable LDP over TE Automatically on Cloned Tunnels	Release 7.5.3	You can now enable LDP automatically on all the cloned tunnels and associate the parent and cloned tunnels. This feature helps you to overcome the 1k tunnel limitation under LDP because the cloned tunnels are not accounted for in the global count. Earlier, LDP had to be enabled on all the cloned tunnels manually, which was time-consuming. This feature introduces the **clone-tunnels** command.

When you run tunnels end-to-end from a PE device to a remote PE, there's no need to enable LDP on tunnels. However, this approach requires a full mesh of tunnels which may lead to a high tunnel scale in large networks.

So in a large network, you may prefer to start a tunnel on a midpoint router. In this case, enable LDP over the tunnel so that a label path can be established end-to-end.

When LDP is enabled on named TE tunnels, and when the tunnels are cloned, currently there's no way to configure LDP automatically on all the cloned tunnels. You have to enable LDP on all the cloned tunnels manually, which might be time consuming.

Now you can enable LDP automatically for the named tunnel interfaces with minimal configuration steps. You don't need to configure hundreds of tunnels under LDP explicitly. Use the **clone-tunnel** command to enable LDP automatically on all the cloned tunnels.

Though LDP has 1K tunnel-interface limitation, cloned tunnels that are present is not considered in the global count.

For information about cloned tunnels, see Configuring Autobandwidth Bundle TE++.

Configure LDP over TE on Cloned Tunnels

You can enable LDP over TE automatically on cloned tunnels using the **clone-tunnel** configuration under **mpls-ldp address family ipv4**.

```
Router(config)#configure
Router(config)#mpls ldp
Router(config-ldp)#nsr
Router(config-ldp)#router-id 10.10.1.1
Router(config-ldp)#address-family ipv4
Router(config-ldp-af)#discovery targeted-hello accept
Router(config-ldp-af)#exit
Router(config-ldp)#interface tunnel-te n1
Router(config-ldp-if)#address-family ipv4
Router(config-ldp-if-af)#clone-tunnel
Router(config-ldp-if-af)#exit
Router(config-ldp-if)#interface tunnel-te n2
Router(config-ldp-if)#address-family ipv4
Router(config-ldp-if-af)#clone-tunnel
Router(config-ldp-if-af)#exit
Router(config-ldp-if)#interface tunnel-te n3
Router(config-ldp-if)#address-family ipv4
Router(config-ldp-if-af)#clone-tunnel
Router(config-ldp-if-af)#commit
```

Running Configuration

This section shows LDP over TE on cloned tunnels running configuration.

```
Router#show running-config mpls ldp
mpls ldp
log
  hello-adjacency
  neighbor
  nsr
  session-protection
!
nsr
router-id 10.10.1.1
address-family ipv4
  discovery targeted-hello accept
!
interface tunnel-te n1
  address-family ipv4
   clone-tunnel
  !
!
interface tunnel-te n2
  address-family ipv4
   clone-tunnel
  !
!
interface tunnel-te n3
  address-family ipv4
   clone-tunnel
  !
!
!
```

Verification

Verify that LDP is enabled on cloned tunnels.

```
Router#show mpls ldp interface
Interface n1 (0xf000014)
   VRF: 'default' (0x60000000)
   Enabled via config: LDP interface
Interface n1-1 (0xf001f84)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n1-2 (0xf001f8c)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n1-3 (0xf001fb4)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n1-4 (0xf001fbc)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n2 (0xf00001c)
   VRF: 'default' (0x60000000)
   Enabled via config: LDP interface
Interface n2-1 (0xf001f64)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n3 (0xf000024)
   VRF: 'default' (0x60000000)
   Enabled via config: LDP interface
Interface n3-1 (0xf001f74)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
Interface n3-2 (0xf001fc4)
   VRF: 'default' (0x60000000)
   Enabled via config: Cloned TE Tunnel
```

Attribute Set for Named Tunnels

Table 29: Feature History Table

Feature Name	Release Information	Feature Description
Attribute Set for Named Tunnels	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D

Feature Name	Release Information	Feature Description
Attribute Set for Named Tunnels	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M
Attribute Set for Named Tunnels	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Attribute Set for Named Tunnels	Release 7.5.3	You can now add an attribute set to the named tunnels under the path options eliminating the need to configure the TE-tunnel attributes individually. When you change the attribute set values for the path option, the same values are inherited to all the attached path options under named tunnels. The attribute set is a template that specifies a set of tunnel attributes, such as priority, affinity, signaled bandwidth, logging, policy class, record route and so on. Earlier, the attribute set was available only for numbered tunnels. The **attribute-set** keyword is added in the **path option (Named Tunnels)**.

In the traditional TE tunnel naming scheme, the tunnels are configured with IDs, where an ID is a 16-bit number. With the increased TE tunnel scale in the network, and with the 64K limit, there is a scarcity of unique tunnel IDs. So there came the need for named tunnels.

The named tunnel allows you to provision TE tunnels using a string name, which lets you differentiate various TE tunnels.

For a tunnel, you can configure one or more path options and each is identified by a unique name. You can configure the path option attributes through a template. This template is named as attribute set and configured under MPLS traffic-engineering tunnel. 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 TE attribute set template specifies a set of TE tunnel attributes and is configured at the headend of autotunnels. The control plane triggers the automatic provisioning of a corresponding TE tunnel, whose characteristics are specified in the respective attribute set.

Earlier, the attribute set was configurable only for numbered tunnels. With this feature, you can now apply an attribute set to a path option on a per-LSP basis for a named tunnel. The path option configuration is extended to take a path option attribute name. LSPs compute a path option using the set attributes as specified by the attribute set under the path option.

Configure Attribute Set for Named Tunnels

Perform the following task to configure an attribute set for the named tunnel:

- Define attribute set for path option.
- Associate the attribute set to the named tunnel.

```
/* Define attribute set for path option */
Router#configure
Router(config)#mpls traffic-eng
Router(config-mpls-te)#attribute-set path-option P1
Router(config-te-attribute-set)#signalled-bandwidth 10000 class-type 0
Router(config-te-attribute-set)# path-selection
Router(config-te-attrset-psel)#cost-limit 1000
Router(config-te-attrset-psel)#delay-limit 5000
Router(config-te-attrset-psel)#exclude AAA
Router(config-te-attrset-psel)#exit
Router(config-te-attribute-set)#affinity include YELLOW
Router(config-te-attribute-set)#affinity include-strict BLUE GREEN
Router(config-te-attribute-set)#affinity exclude RED
Router(config-te-attribute-set)#affinity exclude-all
Router(config-te-attribute-set)#commit
Router(config-te-attribute-set)#root

/* Associate the attribute set to the named tunnel */
Router(config)#mpls traffic-eng
Router(config-mpls-te)#named-tunnels
Router(config-te-named-tunnels)#tunnel-te cisco
Router(config-te-tun-name)#path-option 1
Router(config-po-name)#attribute-set P1
Router(config-po-name)#commit
Router(config-po-name)#root
Router(config)#exit
```

Running Configuration

This section shows the running configurations.

```
Router#show run
mpls traffic-eng
 attribute-set path-option P1
  signalled-bandwidth 10000 class-type 0
  path-selection
   cost-limit 1000
   delay-limit 5000
   exclude AAA
  !
  affinity include YELLOW
  affinity include-strict BLUE GREEN
  affinity exclude RED
  affinity exclude-all
 !
 named-tunnels
  tunnel-te cisco
   path-option 1
    attribute-set P1
   !
  !
 !
!
```

Verification

Verify that the attribute set is associated with the named tunnel.

```
Router#show mpls traffic-eng tunnel name cisco detail

Name: cisco  Destination: 0.0.0.0  Ifhandle:0xf000014
  Tunnel-ID: 32768
  Status:
    Admin:    up Oper: down   Path: not valid   Signalling: Down
    path option 1, preference unspecified,  type unspecified
      Path-option attribute: P1
        Number of affinity constraints: 4
           Include bit map           : 0x0
           Include ext bit map       :
               Length: 256 bits
               Value : 0x::
           Include affinity name     :
           Undefined affinity name   : YELLOW
           Include-strict bit map    : 0x0
           Include-strict ext bit map:
               Length: 256 bits
               Value : 0x::
           Include-strict name       :
           Undefined affinity name   : BLUE GREEN
           Exclude bit map           : 0x0
           Exclude ext bit map       :
               Length: 256 bits
               Value : 0x::
           Exclude affinity name     :
           Undefined affinity name   : RED
           Exclude all
```

Configure fast reroute

Fast reroute (FRR) provides link or node protection to LSPs enabling the traffic carried by LSPs that encounter a failed link or node to be rerouted around the failure. The reroute decision is controlled locally by the router connected to the failed link or node. The headend router on the tunnel is notified of the link or node failure through IGP or through RSVP. When it is notified of a link or node failure, the headend router attempts to establish a new LSP that bypasses the failure. This provides a path to reestablish links or nodes that fail, providing protection to data transfer. The path of the backup tunnel can be an IP explicit path, a dynamically calculated path, or a semi-dynamic path. For detailed conceptual information on fast reroute, see the MPLS-TE Features - Details topic.

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 or node to ensure that the link or node 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 or node.

Configuration Example

This example configures fast reroute on an MPLS-TE tunnel. Here, tunnel-te 2 is configured as the back-up tunnel. You can use the **protected-by** command to configure path protection for an explicit path that is protected by another path.

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# fast-reroute
Router(config-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# backup-path tunnel-te 2
Router(config)# interface tunnel-te 2
Router(config-if)# backup-bw global-pool 5000
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 192.168.92.125
Router(config-if)# path-option l explicit name backup-path protected by 10
Router(config-if)# path-option l0 dynamic
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng fast-reroute database** command to verify the fast reroute configuration.

```
Router# show mpls traffic-eng fast-reroute database

Tunnel head FRR information:
Tunnel      Out intf/label                    FRR intf/label      Status
----------  ----------------                  ----------------    -------
tt4000       HundredGigabitEthernet 0/0/0/3:34     tt1000:34        Ready
tt4001       HundredGigabitEthernet 0/0/0/3:35     tt1001:35        Ready
tt4002       HundredGigabitEthernet 0/0/0/3:36     tt1001:36        Ready
```

Configuring 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 the MPLS-TE tunnels.

In traditional TE, 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.

Cisco IOS XR Release 7.5.4 introduces support to configure the affinities to include any of the affinity names for the links. The link with any additional colors or a link without a color is also accepted.

Configuration Example

This example shows assigning a how to associate a tunnel with affinity constraints.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# affinity-map red 1
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# attribute-names red
Router(config)# interface tunnel-te 2
Router(config-if)# affinity include red
Router(config)# commit
```

Configuring Forwarding Path

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

Configuration Example

```
Router # configure
Router(config)# interface tunnel-te 1
Router(config-if)# forward-class 1
Router(config-if)# exit
Router(config)# commit
```

Configure an IETF DS-TE tunnel using MAM

Table 30: Feature History Table

Feature Name	Release Information	Feature Description
Configure an IETF DS-TE tunnel using MAM	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.

Configure an IETF DS-TE tunnel using MAM	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Configure an IETF DS-TE tunnel using MAM	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Configure an IETF DS-TE tunnel using MAM	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The router now supports assured bandwidth allocation using Maximum Allocation Model (MAM). It enhances network efficiency by allocating bandwidth based on the maximum reservation across all classes, thus optimizing usage. MAM ensures predictable performance and facilitates the reservation of guaranteed bandwidth for critical applications. This capability is vital for maintaining service quality in networks with varied traffic demands. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

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 Russian Doll Model (RDM) and Maximum Allocation Model (MAM), both with two bandwidth pools.

Configuration Example

This example configures an IETF DS-TE tunnel using MAM.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth mam max-reservable-bw 1000 bc0 600 bc1 400
Router(config-rsvp-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# ds-te mode ietf
```

```
Router(config-mpls-te)# ds-te bc-model mam
Router(config-mpls-te)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the IETF DS-TE tunnel using MAM configuration.

Configure an IETF DS-TE tunnel using RDM

Table 31: Feature History Table

Feature Name	Release Information	Feature Description
Configure an IETF DS-TE tunnel using RDM	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Configure an IETF DS-TE tunnel using RDM	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Configure an IETF DS-TE tunnel using RDM	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Configure an IETF DS-TE tunnel using RDM	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) The router can now implement bandwidth management through Russian Dolls Model (RDM), allowing hierarchical bandwidth sharing among different service classes. RDM maximizes resource utilization by enabling unused bandwidth to be shared dynamically, ensuring efficient use of network resources. This feature supports flexible bandwidth management, crucial for adapting to fluctuating traffic patterns and maintaining high service quality. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

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 Russian Doll Model (RDM) and Maximum Allocation Model (MAM), both with two bandwidth pools. In an IETF DS-TE network, identical bandwidth constraint models must be configured on all nodes.

Before you Begin

The following prerequisites are required to create a 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.

Configuration Example

This example configures an IETF DS-TE tunnel using RDM.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth rdm 100 150
Router(config-rsvp-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls-te)# ds-te mode ietf
Router(config-mpls-te)# exit
```

```
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10 class-type 1
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the IETF DS-TE tunnel using RDM configuration.

Configuring an MPLS Traffic Engineering Interarea Tunneling

Feature Name	Release Information	Description
MPLS Traffic Engineering Interarea Tunneling	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on the Cisco 8712-MOD-M routers.
MPLS Traffic Engineering Interarea Tunneling	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) You can simplify network management and reduce manual routing complexity by using MPLS TE interarea tunnels, which allow a single MPLS TE tunnel to span multiple IGP areas. This eliminates the need to configure separate tunnels for each area. The headend router specifies a loosely routed explicit path through Area Border Routers (ABRs). Each ABR then computes the path segment to the next ABR or the tunnel destination, enabling seamless traffic engineering across different areas using either IS-IS or OSPF protocols. *This feature is supported on: • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 8212-48FH-M • 8711-32FH-M

The MPLS TE Interarea Tunneling feature allows you to establish MPLS TE tunnels that span multiple Interior Gateway Protocol (IGP) areas and levels. This feature removes the restriction that required the tunnel headend and tailend routers both to be in the same area. The IGP can be either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).To configure an inter-area tunnel, you specify on the headend router a loosely routed explicit path for the tunnel label switched path (LSP) that identifies each area border router (ABR) the LSP should traverse using the next-address loose command. The headend router and the ABRs along the specified explicit path expand the loose hops, each computing the path segment to the next ABR or tunnel destination.

Configuration Example

This example configures an IPv4 explicit path with ABR configured as loose address on the headend router.

```
Router# configure
Router(config)# explicit-path name interarea1
Router(config-expl-path)# index1 next-address loose ipv4 unicast 172.16.255.129
Router(config-expl-path)# index 2 next-address loose ipv4 unicast 172.16.255.131
Router(config)# interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 172.16.255.2
Router(config-if)# path-option 10 explicit name interarea1
Router(config)# commit
```

MPLS-TE path protection

Path protection provides an end-to-end failure recovery mechanism 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. Failover is triggered by a RSVP error message sent to the LSP head end. Once the head end received this error message, it switches over to the secondary tunnel.

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. Both the explicit and dynamic path-options are supported for the MPLS-TE path protection feature. You should make sure that the same attributes or bandwidth requirements are configured on the protected option.

Table 32: Feature History Table

Feature Name	Release Information	Feature Description
MPLS-TE path protection	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.

MPLS-TE path protection	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
MPLS-TE path protection	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
MPLS-TE path protection	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) With path protection, which provides an end-to-end failure recovery mechanism for MPLS-TE tunnels, a secondary label switched path (LSP) is established in advance so that this secondary LSP acts as a failover LSP to carry forward the tunnel's traffic seamlessly from a failed protected LSP. *Previously this feature was supported on Q200 and Q100. This feature support is extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

Prerequisites

These are some prerequisites before you enable the MPLS-TE path protection feature:

- Ensure that your network supports MPLS-TE, Cisco Express Forwarding, and Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).
- Configure MPLS-TE on the routers.

Restrictions

Remember these restrictions when configuring the MPLS-TE path protection feature:

- User-specified path options are not allowed in the **Path-protection** configuration.
- **protected-by** is used specifically only for numbered tunnels and unavailable for named-tunnels.

Configure MPLS-TE path protection

This procedure provides the steps to configure the MPLS-TE path protection feature.

In this configuration, R1 is the headend router and R3 is the tailend router for the tunnel while R2 and R4 are mid-point routers. In this example, 6 explicit paths and 1 dynamic path are created for path protection. You can have upto 8 path protection options for a primary path.

Before you begin

The primary path option should be present to configure path protection.

Procedure

Step 1 **configure**

Example:

```
Router # configure
```

Enters the global configuration mode.

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

Example:

```
Router(config)# interface tunnel-te 0
```

Enters the interface configuration mode.

Step 3 **destination** *ip-address*

Example:

```
Router(config-if)# destination 192.168.3.3
```

Step 4 **ipv4 unnumbered Loopback** *interface-id*

Example:

```
Router(config-if)# ipv4 unnumbered Loopback 0
```

Specifies the loopback interface.

Step 5 **autoroute announce**

Example:

```
Router(config-if)# autoroute announce
```

Enables autoroute announce.

Step 6 **path-protection**

Example:

```
Router(config-if)# path-protection
```

Enables MPLS-TE path protection.

Step 7 **path-option** *value* **explicit name** *router-names* **protected-by** *value*

Example:

```
Router(config-if)# path-option 1 explicit name r1-r2-r3-00 protected-by 2
Router(config-if)# path-option 2 explicit name r1-r2-r3-01 protected-by 3
Router(config-if)# path-option 3 explicit name r1-r4-r3-01 protected-by 4
Router(config-if)# path-option 4 explicit name r1-r3-00 protected-by 5
Router(config-if)# path-option 5 explicit name r1-r2-r4-r3-00 protected-by 6
Router(config-if)# path-option 6 explicit name r1-r4-r2-r3-00 protected-by 7
```

Creates six explicit paths on the specified combination of routers, r1, r2, r3, and r4. Here r1 is the headend router, r2 is the tailend router, and r3 and r4 are midpoint routers.

Step 8 **path-option** *value* **dynamic**

Example:

```
Router(config-if)# path-option 7 dynamic
```

Creates a dynamic path.

Step 9 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode.

Step 10 **commit**

Example:

```
Router(config)# commit
```

Saves and applies the path protection configuration.

Verification

Use the **show mpls traffic-eng tunnels** command to verify the MPLS-TE path protection configuration.

```
Router# show mpls traffic-eng tunnels 0

Name: tunnel-te0  Destination: 192.168.92.125  Ifhandle:0x8007d34
  Signalled-Name: router
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
    path option 1,  type explicit r1-r2-r3-00 (Basis for Setup, path weight 2)
      Protected-by PO index: 2
    path option 2,  type explicit r1-r2-r3-01 (Basis for Standby, path weight 2)
      Protected-by PO index: 3
    path option 3,  type explicit r1-r4-r3-01
      Protected-by PO index: 4
    path option 4,  type explicit r1-r3-00
      Protected-by PO index: 5
    path option 5,  type explicit r1-r2-r4-r3-00
      Protected-by PO index: 6
```

```
    path option 6,  type explicit r1-r4-r2-r3-00
      Protected-by PO index: 7
    path option 7,  type dynamic
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Fri Oct 13 15:05:28 2017 (01:19:11 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Delay-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute:  enabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (not enabled)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Fast Reroute: Disabled, Protection Desired: None
    Path Protection: Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  History:
    Tunnel has been up for: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Current LSP:
      Uptime: 01:14:13 (since Fri Oct 13 15:10:26 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
    Standby Reopt LSP:
      Last Failure:
        LSP not signalled, identical to the [STANDBY] LSP
        Date/Time: Fri Oct 13 15:08:41 UTC 2017 [01:15:58 ago]
        First Destination Failed: 192.3.3.3
    Prior LSP:
      ID: 8 Path Option: 1
      Removal Trigger: path protection switchover
    Standby LSP:
      Uptime: 01:13:56 (since Fri Oct 13 15:10:43 UTC 2017)
  Path info (OSPF 1 area 0):
  Node hop count: 2
  Hop0: 192.168.1.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
  Standby LSP Path info (OSPF 1 area 0), Oper State: Up :
  Node hop count: 2
  Hop0: 192.168.2.2
  Hop1: 192.168.3.1
  Hop2: 192.168.3.2
  Hop3: 192.168.3.3
Displayed 1 (of 4001) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Configuring Next Hop Backup Tunnel

The backup tunnels that bypass only a single link of the LSP path are referred as Next Hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure. They protect LSPs, if a link along their path fails, by rerouting the LSP traffic to the next hop, thus bypassing the failed link.

Configuration Example

This example configures next hop backup tunnel on an interface and specifies the attribute-set template for the auto tunnels. In this example, unused backup tunnels are removed every 20 minutes using a timer and also the range of tunnel interface numbers are specified.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# auto-tunnel backup nhop-only
Router(config-mpls-te-if-auto-backup)# attribute-set ab
Router(config-mpls-te)# auto-tunnel backup timers removal unused 20
Router(config-mpls-te)# auto-tunnel backup tunnel-id min 6000 max 6500
Router(config)# commit
```

Configuring Point-to-Multipoint TE Tunnels

For P2MP tunnels, a Cisco 8000 Series router supports the mid-point router function, and does not support source or receiver functions. To know how to configure a source or receiver (destination) router in a P2MP tunnel, refer the MPLS configuration guide for the corresponding platform.

Configuring Point-to-Multipoint TE Auto-Tunnels

The P2MP-TE Auto-tunnels feature enables dynamic creation and management of P2MP auto-tunnels for the transport of VPLS traffic on Cisco IOS XR Software. The P2MP-TE auto-tunnel configuration is disabled by default. Use the **auto-tunnel p2mp tunnel-id** command to enable a P2MP-TE Auto-tunnel. This configures the tunnel ID range that can be allocated to P2MP auto-tunnels. This also determines the maximum number of P2MP auto-tunnels that can be created.

Configuration Example

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# auto-tunnel p2mp
Router(config-te-auto-p2mp)# tunnel-id min 10000 max 11000
Router(config-te-auto-p2mp)# commit
```

Enabling Soft-Preemption

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.

Configuration Example

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.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# soft-preemption
Router(config-soft-preemption)# timeout 100
Router(config-soft-preemption)# commit
```

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.

Configuration Example

When soft preemption is enabled on a tunnel, a path-modify message is sent for the current LSP, reopt LSP, path protection LSP, and current LSP in FRR active state, with the **soft preemption desired** property.

```
Router# configure
Router(config)# interface tunnel-te 10
Router(config-if)# soft-preemption
Router(config-if)# commit
```

Enabling Soft-preemption over FRR Backup Tunnels

Before enabling soft-preemption over FRR backup, ensure that you enable soft-preemption, and activate the FRR backup tunnel.

Configuration Example

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# soft-preemption frr-rewrite
Router(config-mpls-te)# commit
```

Configuring Pre-Standard DS-TE

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. MPLS DS-TE enables you to configure multiple bandwidth constraints on an MPLS-enabled interface. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint. Cisco IOS XR software supports two DS-TE modes: Pre-standard and IETF. Pre-standard 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. Pre-standard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces.

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

Before You Begin

The following prerequisites are required to configure a Pre-standard 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.

Configuration Example

This example configures a pre-standard DS-TE tunnel.

```
Router# configure
Router(config)# rsvp interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# bandwidth 100 150 sub-pool 50
Router(config-rsvp-if)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# signalled bandwidth sub-pool 10
Router(config)# commit
```

Verification

Use the **show mpls traffic-eng topology** command to verify the pre-standard DS-TE tunnel configuration.

Configuring SRLG Node Protection

Shared Risk Link Groups (SRLG) in MPLS traffic engineering refer to situations in which links in a network share common resources. These links have a shared risk, and that is when one link fails, other links in the group might fail too.

OSPF and 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.

MPLS-TE SRLG feature enhances backup tunnel path selection by avoiding using links that are in the same SRLG as the interfaces it is protecting while creating backup tunnels.

Configuration Example

This example creates a backup tunnel and excludes the protected node IP address from the explicit path.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigabitEthernet 0/0/0/3
Router(config-mpls-te-if)# backup-path tunnl-te 2
Router(config-mpls-te-if)# exit
Router(config)# interface tunnel-te 2
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# path-option 1 explicit name backup-srlg
Router(config-if)# destination 192.168.92.125
Router(config-if)# exit
Router(config)# explicit-path name backup-srlg-nodep
Router(config-if)# index 1 exclude-address 192.168.91.1
Router(config-if)# index 1 exclude-srlg 192.168.92.2
Router(config)# commit
```

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 signaled, they are verified dynamically in the next path verification cycle.

Creating an MPLS-TE Tunnel

Creating an MPLS-TE tunnel is a process of customizing the traffic engineering to fit your network topology. The MPLS-TE tunnel is created at the headend router. You need to specify the destination and path of the TE LSP.

To steer traffic through the tunnel, you can use the following ways:

- Static Routing
- Autoroute Announce
- Forwarding Adjacency

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.

Configuration Example

This example configures an MPLS-TE tunnel on the headend router with a destination IP address 192.168.92.125. The bandwidth for the tunnel, path-option, and forwarding parameters of the tunnel are also configured. You can use static routing, autoroute announce or forwarding adjacency to steer traffic through the tunnel.

Cisco IOS XR Release 7.5.4 introduces support to configure the timeout period before the headend router retries to establish the LSPs. The timeout range is 1 to 600 seconds.

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# destination 192.168.92.125
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# path-option 1 dynamic
Router(config-if)# retry-timer 300
Router(config-if)# autoroute announce | forwarding-adjacency
Router(config-if)# signalled-bandwidth 100
Router(config)# commit
```

Verification

Verify the configuration of MPLS-TE tunnel using the following command.

```
Router# show mpls traffic-engineering tunnels brief

            Signalling Summary:
```

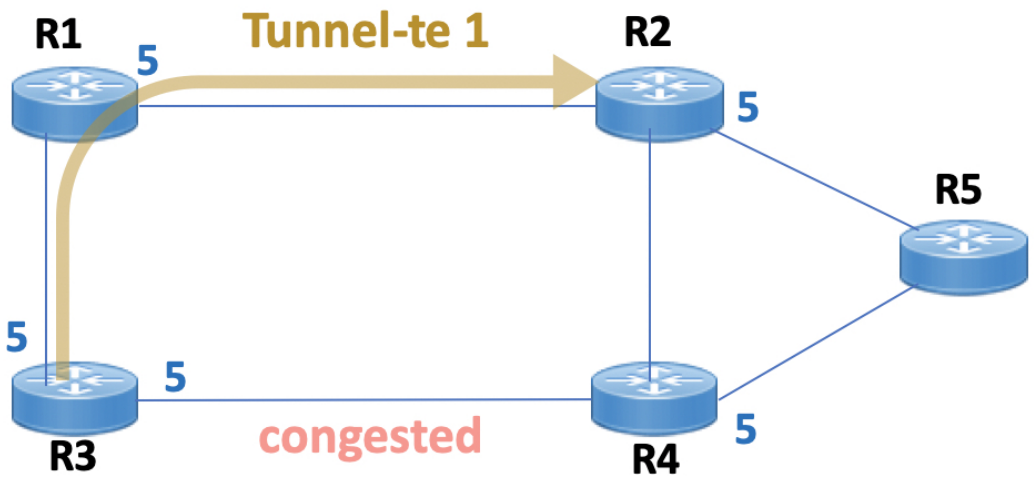
```
    LSP Tunnels Process:  running
           RSVP Process:  running
             Forwarding:  enabled
Periodic reoptimization:  every 3600 seconds, next in 2538 seconds
 Periodic FRR Promotion:  every 300 seconds, next in 38 seconds
Auto-bw enabled tunnels:  0 (disabled)
          TUNNEL NAME          DESTINATION      STATUS  STATE


          tunnel-te1        192.168.92.125      up       up
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Automatic Modification Of An MPLS-TE Tunnel's Metric

If the IGP calculation on a router results in an equal cost multipath (ECMP) scenario where next-hop interfaces are a mix of MPLS-TE tunnels and physical interfaces, you may want to ensure that a TE tunnel is preferred. Consider this topology:

Figure 14: MPLS-TE Tunnel



1. All links in the network have a metric of 5.
2. To offload a congested link between R3 and R4, an MPLS-TE tunnel is created from R3 to R2.
3. If the metric of the tunnel is also 5, traffic from R3 to R5 is load-balanced between the tunnel and the physical R3-R4 link.

To ensure that the MPLS-TE tunnel is preferred in such scenarios, configure the **autoroute metric** command on the tunnel interface. The modified metric is applied in the routing information base (RIB), and the tunnel is preferred over the physical path of the same metric. Sample configuration:

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# autoroute metric relative -1
```

The **autoroute metric** command syntax is **autoroute metric** {**absolute**|**relative**} *value*

- **absolute** enables the absolute metric mode, for a metric range between 1 and 2147483647.

- **relative** enables the relative metric mode, for a metric range between -10 and 10, including zero.

Note

- Since the **relative** metric is not saved in the IGP database, the advertised metric of the MPLS-TE tunnel remains 5, and doesn't affect SPF calculation outcomes on other nodes.
- Configuring Segment Routing and Autoroute Destination together is not supported. If autoroute functionality is required in an Segment Routing network, we recommend you to configure Autoroute Announce.

Traffic steering using autoroute announce

The autoroute announce feature in an MPLS Traffic Engineering (TE) network is a mechanism that:

- simplifies the integration of MPLS TE tunnels into the Interior Gateway Protocol (IGP) routing process, and
- allows an MPLS TE tunnel to automatically advertise itself as a forwarding adjacency (FA) to the IGP, making the tunnel appear like a regular link in the network topology.

Table 33: Feature History Table

Feature Name	Release	Feature Description
MPLS-TE IPv6-only autoroute announce	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.

Feature Name	Release	Feature Description
MPLS-TE IPv6-only autoroute announce	Release 25.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC:Q200]) ; Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) This feature allows you to disable IPv4 autoroute announce without turning off autoroute announce entirely. To achieve IPv6-only announcements over MPLS-TE tunnels, use the new **exclude-ipv4** option along with the **include-ipv6** option in the autoroute announce configuration. This feature introduces these changes: CLI: A new keyword, **exclude-ipv4**, has been added to the **autoroute announce** command.
Autoroute announce with IPv6 destination address for IS-IS	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Autoroute announce with IPv6 destination address for IS-IS	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is now supported on the Cisco 8711-48Z-M routers.

Feature Name	Release	Feature Description
Autoroute announce with IPv6 destination address for IS-IS	Release 25.2.1	Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q200, P100]). The autoroute announce feature now allows an MPLS-TE tunnel as the nexthop for IPv6 destinations thus ensuring a seamless IPv6 traffic engineering. IS-IS uses this IPv6 destination address when filtering prefixes. This feature introduces these changes: CLI: Two new keywords, **destination** and **ipv6**, were introduced in the **autoroute announce** command.

You can use the autoroute announce feature to steer traffic through the MPLS-TE tunnel for optimal network performance and efficient use of resources.

You can use the **autoroute announce** command in interface configuration mode to specify whether the Interior Gateway Protocol (IGP) should use the tunnel (if the tunnel is up) in its enhanced shortest path first (SPF) calculation. To return to the default behavior, use the **no** form of this command.

When more than one IGP is configured, the tunnel is announced as autoroute to the IGP that is used to compute the TE tunnel path.

When the **autoroute announce** command is configured, the route metric of the tunnel path to the destination equals the route metric of the shortest IGP path to that destination.

Starting from Cisco IOS XR Release 25.4.1, the autoroute announce feature restricts the usage of MPLS-TE tunnels by IGP to IPv6 prefixes. To enable the IPv6-only autoroute announce feature, configure the **autoroute announce exclude-ipv4** and **autoroute announce include-ipv6** commands.

Note IS-IS is the only IGP supporting IPv6 MPLS-TE tunnel announcements.

Configure autoroute announce

This example shows how to configure IGP to use the tunnel in its enhanced SPF calculation when the tunnel is up:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# autoroute announce
```

Use the **autoroute announce include-ipv6** to announce the MPLS-TE tunnel to IS-IS IGP for IPv6 routing.

This example shows how an MPLS-TE tunnel can be announced to IS-IS IGP for IPv6 routing:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 65534
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if-tunte-aa)# include-ipv6
```

This example shows how an MPLS-TE tunnel can be announced to IS-IS IGP for IPv6-only routing:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if-tunte-aa)# exclude-ipv4
RP/0/RP0/CPU0:router(config-if-tunte-aa)# include-ipv6
```

The **autoroute announce metric** configuration allows you to specify the MPLS-TE tunnel metric that the Interior Gateway Protocol (IGP) enhanced Shortest Path First (SPF) calculation uses.

The **autoroute announce metric** configuration overrides the autoroute metric configuration, if present.

Configure IPv6 routing over IPv4 MPLS-TE tunnels

Perform these steps to configure IPv6 routing over IPv4 MPLS-TE tunnels:

Procedure

Step 1 **configure**

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

Example:

```
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
```

Configures an MPLS-TE tunnel interface.

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

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv4 unnumbered Loopback 0
```

Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is the commonly-used interface type.

Step 4 **ipv6 enable**

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv6 enable
```

Enables IPv6 on interface.

Step 5 **signalled-bandwidth** *bandwidth*

Example:

```
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10
```

Sets the tunnel bandwidth requirement to be signalled in Kbps.

Step 6 **destination** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-if)#destination 192.168.0.1
```

Specifies the IPv4 destination address.

Step 7 Use one of these options:

- **autoroute announce include-ipv6**
- **forwarding-adjacency include-ipv6**

Example:

```
RP/0/RP0/CPU0:router(config-if)#autoroute announce include-ipv6
```

Or

```
RP/0/RP0/CPU0:router(config-if)#forwarding-adjacency include-ipv6
```

Announces the tunnel as an IPv6 autoroute, or as an IPv6 forwarding adjacency.

Step 8 **path-option** *preference-priority* **dynamic**

Example:

```
RP/0/RP0/CPU0:router(config-if)#path-option 1 dynamic
```

Sets the path option to dynamic and assigns the path ID.

Step 9 **commit**

Step 10 (Optional) **show mpls traffic-eng autoroute**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng autoroute

  Destination 192.168.0.2 has 1 tunnels in IS-IS ring level 1
    tunnel-te1 (traffic share 0, nexthop 192.168.0.2)
               (IPv4 unicast)
               (IPv6 unicast)
```

Verifies that the tunnel announces IPv6 autoroute information.

Step 11 (Optional) **show mpls traffic-eng forwarding-adjacency**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng forwarding-adjacency

 destination 192.168.0.1 has 1 tunnels

              tunnel-te10    (traffic share 0, next-hop 192.168.0.1)
                             (Adjacency Announced: yes, holdtime 0)
                             (IS-IS 100, IPv4 unicast)
                             (IS-IS 100, IPv6 unicast)
```

Verifies that the tunnel announces IPv6 forwarding adjacency information.

Configure IPv6 routing over IPv6 MPLS-TE tunnels

Perform these steps to configure IPv6 routing over IPv6 MPLS-TE tunnels:

Procedure

Step 1 **configure**

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

Example:

```
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
```

Configures an MPLS-TE tunnel interface.

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

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv4 unnumbered Loopback 0
```

Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is the commonly-used interface type.

Step 4 **ipv6 enable**

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv6 enable
```

Enables IPv6 on interface.

Step 5 **signalled-bandwidth** *bandwidth*

Example:

```
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10
```

Sets the tunnel bandwidth requirement to be signalled in Kbps.

Step 6 Use one of these options:

- **autoroute announce include-ipv6**
- **forwarding-adjacency include-ipv6**

Example:

```
RP/0/RP0/CPU0:router(config-if)#autoroute announce include-ipv6
```

Or

```
RP/0/RP0/CPU0:router(config-if)#forwarding-adjacency include-ipv6
```

Announces the tunnel as an IPv6 autoroute, or as an IPv6 forwarding adjacency.

Step 7 **destination ipv6** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-if)#destination ipv6 2000::1
```

Specifies the IPv6 destination address.

Step 8 **path-option** *preference-priority* **dynamic**

Example:

```
RP/0/RP0/CPU0:router(config-if)#path-option 1 dynamic
```

Sets the path option to dynamic and assigns the path ID.

Step 9 **commit**

Step 10 (Optional) **show mpls traffic-eng autoroute**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng autoroute

  Destination 192.168.0.2 has 1 tunnels in IS-IS ring level 1
    tunnel-te1 (traffic share 0, nexthop 192.168.0.2)
               (IPv4 unicast)
               (IPv6 unicast)
```

Verifies that the tunnel announces IPv6 autoroute information.

Step 11 (Optional) **show mpls traffic-eng forwarding-adjacency**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng forwarding-adjacency

 destination 192.168.0.1 has 1 tunnels

              tunnel-te10    (traffic share 0, next-hop 192.168.0.1)
                             (Adjacency Announced: yes, holdtime 0)
                             (IS-IS 100, IPv4 unicast)
                             (IS-IS 100, IPv6 unicast)
```

Verifies that the tunnel announces IPv6 forwarding adjacency information.

Configure IPv6-only autoroute announce for MPLS-TE tunnels

Perform these steps to configure IPv6-only autoroute announce over MPLS-TE tunnels:

Procedure

Step 1 Enable IPv6-only autoroute announce on MPLS-TE tunnels.

Example:

```
Router# config
Router(config)# interface tunnel-te 1
Router(config-if)# autoroute announce include-ipv6
Router(config-if-tunte-aa)# exclude-ipv4
Router(config-if-tunte-aa)# commit
```

Step 2 Verify the IPv6-only autoroute configuration.

Example:

```
Router# show mpls traffic-eng autoroutes
  Destination 192.168.0.2 has 1 tunnels in IS-IS ring level 1
    T1 (traffic share 0, nexthop 192.168.0.3,  metric 0)
    (IS-IS ring level-1, IPV6 Unicast)
    Signalled-Name: T1
```

Configure autoroute announce to prevent IP traffic steering over MPLS tunnel

To prevent the MPLS labelled segment routing (SR) traffic to be forwarded through the MPLS-TE tunnel, use the **autoroute announce exclude-traffic segment-routing** command.

This example shows how to configure the **exclude-traffic segment-routing** command that prevents the MPLS labelled SR prefixes from resolving over the MPLS-TE tunnel:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 3200
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if-tunte-aa)# exclude-traffic segment-routing
```

Prevention of IP traffic steering to destinations with segment routing prefix SID

Table 34: Feature History Table

Feature Name	Release Information	Feature Description
Prevention of IP traffic steering to destinations with segment routing prefix SID	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Prevention of IP traffic steering to destinations with segment routing prefix SID	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Prevention of IP traffic steering to destinations with segment routing prefix SID	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
Prevention of IP traffic steering over MPLS-TE tunnel	Release 24.4.1	Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q200, P100]). This feature allows to prevent the MPLS labelled traffic or IP traffic to destinations associated with a segment routing prefix SID from resolving over the MPLS-TE tunnel thus ensuring optimal hardware resource utilization. This feature enhances the existing configuration that prevents only the MPLS labelled segment routing traffic from steering into the MPLS-TE tunnel. The feature introduces these changes: **CLI**: The **all** keyword is introduced in the **autoroute announce exclude-traffic segment-routing** command.

Configure autoroute announce to prevent IP traffic steering to destinations with segment routing prefix SID

The **autoroute announce exclude-traffic segment-routing** configuration does not prevent steering of IP traffic to destinations associated with a segment routing prefix SID into the MPLS-TE tunnel.

To prevent the MPLS labelled traffic or IP traffic to destinations associated with a segment routing prefix SID from resolving over the MPLS-TE tunnel, use the **all** keyword introduced in Cisco IOS XR Release 24.4.1 in the **autoroute announce exclude-traffic segment-routing** command.

This example shows how to configure the **exclude-traffic segment-routing all** command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-te 3200
RP/0/RP0/CPU0:router(config-if)# autoroute announce
RP/0/RP0/CPU0:router(config-if-tunte-aa)# exclude-traffic segment-routing all
```

Enable per-application accounting for interfaces

The following configurations helps enable per-application accounting for MPLS RSVP-TE IPv4 traffic and segment routing with MPLS IPv4 traffic on the interfaces respectively.

Example 1: Enable per-application accounting for MPLS RSVP-TE IPv4 traffic on the interfaces

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# accounting
```

```
RP/0/RP0/CPU0:router(config-acct)# interfaces
RP/0/RP0/CPU0:router(config-acct-if)# mpls
RP/0/RP0/CPU0:router(config-acct-if-mpls)# ipv4 rspv-te
```

Example 2: Enable per-application accounting for segment routing MPLS IPv4 traffic on the interfaces

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# accounting
RP/0/RP0/CPU0:router(config-acct)# interfaces
RP/0/RP0/CPU0:router (config-acct-if)# segment-routing mpls ipv4
```

Configure dark bandwidth accounting

To enable RSVP-TE Dark Bandwidth Accounting feature, perform the following steps:

1. Enable per-interface aggregate SR counters.
2. Configure TE dark bandwidth accounting.

Before you begin

Make sure that the traffic accounting feature is enabled on the router.

SUMMARY STEPS

1. **configure**
2. **hw-module profile cef dark-bw enable**
3. **mpls traffic-eng**
4. **bandwidth-accounting**
5. **application interval** *seconds*
6. **application enforced**
7. **sampling-interval** *seconds*
8. **adjustment-factor** *percentage*
9. **flooding threshold up** *percentage* **down** *percentage*
10. **flooding sr-traffic** *percentage*
11. Use the **commit** or **end** command.
12. **mpls traffic-eng link-management bandwidth-accounting enforce all**
13. **clear mpls traffic-eng link-management bandwidth-accounting**
14. **show interface** *type_path* **accounting**
15. **show mpls traffic-eng link-management summary**
16. **show mpls traffic-eng link-management advertisements**
17. **show mpls traffic-eng link-management interfaces** [*type interface-path-id*] [**detail**] [**bandwidth-accounting**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**hw-module profile cef dark-bw enable**
	Example:
	`RP/0/RP0/CPU0:router(config)# hw-module profile cef dark-bandwidth enable`
	`RP/0/RP0/CPU0:router(config)# commit`
		Enables per-interface aggregate SR counters for all interfaces on the router.
		Note After you enter this command, you must re
		Caution This command should only be enabled on a has the same out label across all its paths. T network with homogenous SRGB and whe
		Do not use this command on a router with T extra labels to be imposed.
		In Cisco IOS XR release 7.5.2 and earlier, prefix SID is learned via ECMPs with diffe traffic loss would be observed for those pre longer applies.

Step 3	**mpls traffic-eng** **Example:** `RP/0/RP0/CPU0:router(config)# mpls traffic-eng`	Enters MPLS TE configuration mode.
Step 4	**bandwidth-accounting** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te)# bandwidth-accounting`	Enables RSVP-TE dark bandwidth accounting and enters bandwidth accounting configuration mode.
Step 5	**application interval** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# application interval 90`	Configures the length of the application interval in seconds. At the end of application interval, dark bandwidth rates are computed and applied to all RSVP-TE enabled interfaces. If the interval is reconfigured while the timer is running, the new value is compared to the time remaining for the

	Command or Action	Purpose
		running timer. The timer is adjusted so that the lower of these two values is used for this interval. The subsequent interval will use the newly configured value.  **Note** TE stores sample history for the current lowered, TE may discard the sample his Range is from 90 to 1800. The default value is 180.
Step 6	**application enforced** **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# application enforced`	Enables enforcement of the calculated BMRe rate. This configuration is mandatory for the RSVP dark bandwidth accounting feature to work properly.
Step 7	**sampling-interval** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# sampling-interval 30`	Configures the length of the sampling interval in seconds. The dark bandwidth rate is collected from the statistics collector process (statsD) at the end of each sampling interval for each TE link. If the interval is reconfigured while the timer is running, the new value is compared to the time remaining for the running timer. The timer is adjusted so that the lower of these two values is used for this interval. The subsequent interval will use the newly configured value. Range is from 10 to 600. The default is 60.
Step 8	**adjustment-factor** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# adjustment-factor 200`	Configures TE to over-book (>100%) or under-book (<100%) the effective maximum reservable bandwidth (BMRe). The measured dark-bandwidth will be scaled based on the adjustment factor. Range is from 0 to 200. The default value is 100.
Step 9	**flooding threshold up** *percentage* **down** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# flooding threshold up 30 down 30`	Configures the reserved bandwidth thresholds. When bandwidth crosses one of these thresholds, flooding is triggered. Range is from 0 to 100. The default value is 10.
Step 10	**flooding sr-traffic** *percentage* **Example:** `RP/0/RP0/CPU0:router(config-mpls-te-bw-account)# flooding sr-traffic 30`	Configures the flooding trigger for bandwidth accounting in segment routing traffic. When the bandwidth crosses the threshold value, flooding is triggered. Range is from 0 to 100. The default value is 10. This support is introduced in Cisco IOS XR Release 7.5.4.

	Command or Action	Purpose
Step 11	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 12	**mpls traffic-eng link-management bandwidth-accounting enforce all** **Example:** `RP/0/RP0/CPU0:router# mpls traffic-eng link-management bandwidth-accounting enforce all`	(Optional) Applies the measured rates immediately. When you apply measured rates immediately, the RSVP-TE bandwidth-accounting might flood the updated bandwidth values immediately. Applying measured rates immediately does not affect the periodic application of the bandwidth.
Step 13	**clear mpls traffic-eng link-management bandwidth-accounting** **Example:** `RP/0/RP0/CPU0:router# clear mpls traffic-eng link-management bandwidth-accounting`	(Optional) Erases the collected sample history and resets the application and sample timers.
Step 14	**show interface** *type_path* **accounting** **Example:** `RP/0/RP0/CPU0:router# show interface hundredGigE 0/0/0/26 accounting`	(Optional) Displays the per-interface SR accounting.
Step 15	**show mpls traffic-eng link-management summary** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management summary`	(Optional) Displays a summary of link management information, including bandwidth accounting information.
Step 16	**show mpls traffic-eng link-management advertisements** **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management advertisements`	(Optional) Displays local link information that MPLS-TE link management is currently flooding into the global TE topology.

	Command or Action	Purpose
Step 17	**show mpls traffic-eng link-management interfaces** [*type interface-path-id*] [**detail**] [**bandwidth-accounting**] **Example:** `RP/0/RP0/CPU0:router# show mpls traffic-eng link-management interfaces gig0/1/1/1 detail`	(Optional) Displays bandwidth accounting and utilization details and link management information.

To display the per-interface SR counters, use the **show interface** *type_path* **accounting** command:

```
RP/0/RP0/CPU0:router# show interface hundredGigE 0/0/0/26 accounting
Mon Feb  3 23:29:48.449 UTC
HundredGigE0/0/0/26
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  ARP                         3              222            3              126
  IPV6_ND                    11             1122           13             1112
  CLNS                       99           121910           94           116212
  SR_MPLS                     0                0         3126           581436
```

Note The SR_MPLS counter is an egress-only counter and includes all traffic from the following:

- IPv4 unlabelled - SR last-hop traffic after PHP
- IPv6 unlabelled - SR last-hop traffic after PHP
- SR label switched traffic

To display detailed SR bandwidth utilization, use the **show mpls traffic-eng link-management interface** *type_path* **detail** command:

```
Router# show mpls traffic-eng link-management interface hundredGigE 0/0/0/26 detail
bandwidth-accounting

  System Information::
      Links Count         : 16 (Maximum Links Supported 800)

  Link ID:: HundredGigE0/0/0/26 (26.1.1.1)
    Local Intf ID: 22
    Link Status:

      Link Label Type          : PSC
      Physical BW              : 1000000 kbits/sec
      BCID                     : RDM
      Max Reservable BW        : 529309 kbits/sec (reserved: 94% in, 94% out)
      Flooded Max Reservable BW: 529309 kbits/sec
      BC0 (Res. Global BW)     : 529309 kbits/sec (reserved: 94% in, 94% out)
      BC1 (Res. Sub BW)        : 0 kbits/sec (reserved: 100% in, 100% out)
      MPLS TE Link State       : MPLS TE on, RSVP on, admin-up
      IGP Neighbor Count       : 1
      Max Res BW (RDM)     : 900000 kbits/sec
      BC0 (RDM)            : 900000 kbits/sec
      BC1 (RDM)            : 0 kbits/sec
      Max Res BW (MAM)     : 0 kbits/sec
```

```
    BC0 (MAM)           : 0 kbits/sec
    BC1 (MAM)           : 0 kbits/sec

  Bandwidth Accounting: Segment-Routing
  Bandwidth Accounting Enforced: Yes
    Bandwidth Utilization Details:
      Sampling Interval               : 30 sec
      Application Interval            : 90 sec
      Adjustment Factor               : 200%
      Max Reservable BW Up Threshold  : 30
      Max Reservable BW Down Threshold: 30

Last Application at:  23:46:32 Mon 03 Feb 2020 (51 seconds ago)
        Segment-Routing BW Utilization      : 185346 kbits/sec
        Adjusted BW Utilization             : 370692 kbits/sec
        Enforced BW Utilization             : 370692 kbits/sec
      Next Application at:  19:42:43 Sun 30 Apr 2017 (in 38 seconds)
      Last Collection at :  19:41:42 Sun 30 Apr 2017 (23 seconds ago)
      Next Collection at :  19:42:11 Sun 30 Apr 2017 (in 6 seconds)

Bandwidth Samples (Kbps):
      Timestamp                     Segment-Routing
      19:40:12 Sun 30 Apr 2017               187961
      19:40:42 Sun 30 Apr 2017               180130
      19:41:12 Sun 30 Apr 2017               187949
```

To display a summary of link management information, including bandwidth accounting information, use the **show mpls traffic-eng link-management summary** command:

```
Router# show mpls traffic-eng link-management summary

  System Information::
      Links Count         : 14 (Maximum Links Supported 800)
      Flooding System     : enabled
      IGP Areas Count     : 1

  IGP Areas
  ----------

  IGP Area[1]:: IS-IS 0 level 2
      Flooding Protocol   : IS-IS
      Flooding Status     : flooded
      Periodic Flooding   : enabled (every 180 seconds)
      Flooded Links       : 7
      IGP System ID       : 0000.0000.0001
      MPLS TE Router ID   : 10.0.0.1
      IGP Neighbors       : 7

  Bandwidth accounting:
      Sampling interval: 30 seconds, Next in 29 seconds
      Application interval: 90 seconds, Next in 1 seconds
```

To display local link information that MPLS-TE link management is currently flooding into the global TE topology, use the **showmpls traffic-eng link-management advertisements** command:

```
Router# show mpls traffic-eng link-management advertisements

  Flooding Status             : Ready
  Last Flooding               : 470 seconds ago
  Last Flooding Trigger       : Link BW changed
  Next Periodic Flooding In   : 143 seconds
  Diff-Serv TE Mode           : Not enabled
  Configured Areas            : 1
```

```
IGP Area[1]:: IS-IS 0 level 2
    Flooding Protocol   : IS-IS
    IGP System ID       : 0000.0000.0001
    MPLS TE Router ID   : 10.0.0.1
    Flooded Links       : 5

    Link ID:: 0 (GigabitEthernet0/1/1/0)
        Link IP Address      : 10.12.110.1
        O/G Intf ID          : 22
        Neighbor             : ID 0000.0000.0002.00, IP 10.12.110.2
        TE Metric            : 10
        IGP Metric           : 10
        Physical BW          : 1000000 kbits/sec
        BCID                 : RDM
        Max Reservable BW    : 899999 kbits/sec
        Res Global BW        : 899999 kbits/sec
        Res Sub BW           : 0 kbits/sec
```

Configure Autoroute Tunnel as Designated Path

Table 35: Feature History Table

Feature Name	Release Information	Feature Description
Configure Autoroute Tunnel as Designated Path	Release 7.6.2	Simplify the path selection for a traffic class and split traffic among multiple TE tunnels to achieve many benefits such as security and service-level agreements. You can now exclusively specify an autoroute tunnel to forward traffic to a particular tunnel destination address without considering the IS-IS metric for traffic path selection. Earlier, MPLS-TE considered either the Forwarding Adjacency (FA) or Autoroute (AA) tunnel to forward traffic based only on IS-IS metric. The feature introduces the **mpls traffic-eng tunnel restricted** command.

MPLS-TE builds a unidirectional tunnel from a source to a destination using label switched path (LSP) to forward traffic.

To forward the traffic through MPLS tunneling, you can use autoroute, forwarding adjacency, or static routing:

- Autoroute (AA) functionality allows to insert the MPLS TE tunnel in the Shortest Path First (SPF) tree for the tunnel to transport all the traffic from the headend to all destinations behind the tail-end. AA is only known to the tunnel headend router.

- Forwarding Adjacency (FA) allows the MPLS-TE tunnel to be advertised as a link in an IGP network with the cost of the link associated with it. Routers outside of the TE domain can see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network.
- Static routing allows you to inject static IP traffic into a tunnel as the output interface for the routing decision.

Prior to this release, by default, MPLS-TE considers FA or AA tunnels to forward traffic based on the IS-IS metric. The lower metric is always used to forward traffic. There was no mechanism to forward traffic to a specific tunnel interface.

For certain prefixes to achieve many benefits such as security and service-level agreements, there might be a need to forward traffic to a specific tunnel interface that has a matching destination address.

With this feature, you can exclusively use AA tunnels to forward traffic to their tunnel destination address irrespective of IS-IS metric. Traffic steering is performed based on the prefixes and not metrics. Traffic to other prefixes defaults to the forwarding-adjacency (FA) tunnels.

To enable this feature, use the **mpls traffic-eng tunnel restricted** command.

Also, you may require more than one AA tunnel to a particular remote PE and use ECMP to forward traffic across AA tunnels. You can configure a loopback interface with one primary address and multiple secondary addresses on the remote PE, using one IP for the FA tunnel destination, and others for the AA tunnels destinations. Multiple IP addresses are advertised in the MPLS TE domain using the typed length value (TLV) 132 in IS-IS. A TLV-encoded data stream contains code related to the record type, the record length of the value, and value. TLV 132 represents the IP addresses of the transmitting interface.

Feature Behavior

When MPLS-TE tunnel restricted is configured, the following is the behavior:

- A complete set of candidate paths is available for selection on a per-prefix basis during RIB update as the first hop computation includes all the AA tunnels terminating on a node up to a limit of 64 and the lowest cost forwarding-adjacency or native paths terminating on the node or inherited from the parent nodes in the first hops set for the node.
- During per-prefix computation, AA tunnel first hops are used for traffic sent to their tunnel destination address even if FA tunnel or native first hops have a better metric. AA tunnel first-hops are not used for any other prefixes.
- ECMP is used when multiple AA tunnel first hops have the same destination address and metric.
- During per-prefix computation, AA tunnel first hops are used for traffic sent to their tunnel destination address, and for all other destinations on the tunnel tail node or behind it, even if a native path has a better metric.

  Adding **mpls traffic-eng tunnel preferred** configuration has no effect when the tunnel restricted is already configured.
- If there's no AA tunnel or if the tunnel is down, then native paths are used for all other destinations on the tunnel tail node or behind it.

  The route metric for a prefix reflects the chosen first-hop, not necessarily the lowest cost SPF distance to the node.

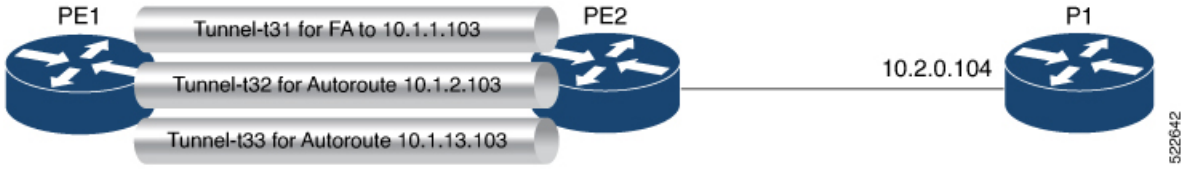
Restrictions for Configure Autoroute Tunnel as Designated Path

- The total number of interface addresses to the number that can be contained in 255 bytes is 63 for IPv4 and 15 for IPv6.
- When this feature is enabled, a maximum of 64 tunnels can terminate on the tail node.

Configure Autoroute Tunnel as Designated Path

Let's understand how to configure the feature using the following topology:

Figure 15: Topology



Consider the topology where PE1 has three MPLS tunnels connecting to PE2.

- Tunnel-t31: Forwarding adjacency (FA) is configured to the primary address of Loopback 0 on PE2 (10.1.1.103).
- Tunnel- t32: Autoroute announce (AA) is configured to a secondary address of Loopback 0 on PE2 (10.1.2.103).
- Tunnel-t33: Autoroute announce (AA) is configured to a secondary address of Loopback 0 on PE2 (10.1.3.103).

This feature is not enabled by default. When this feature is not enabled, traffic is load balanced over all AA tunnels towards the same remote PE provided the tunnel metric is the same:

```
Router# show routes
i L2 10.1.1.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.1.3.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/40] via 10.1.3.103, 00:00:30, tunnel-t33
 10.2.0.104/32 [115/50] via 10.1.2.103, 00:00:30, tunnel-t32
                   [115/50] via 10.1.3.103, 00:00:30, tunnel-t33
```

Configuration Example

You can configure the feature using the **mpls traffic-eng tunnel restricted** command.

```
RP/0/RSP0/CPU0:ios# configure
RP/0/RSP0/CPU0:ios(config)# router isis 1
RP/0/RSP0/CPU0:ios(config-isis)# address-family ipv4 unicast
RP/0/RSP0/CPU0:ios(config-isis-af# mpls traffic-eng tunnel restricted
```

Running Configuration

The following example shows the AA tunnel metric running configuration:

```
router isis 1
 address-family ipv4 unicast
  mpls traffic-eng tunnel restricted
 !
!
end
```

Verification

When you enable the feature, traffic towards a particular prefix is sent only over the tunnel that has that IP address as destination.

```
Router# show route
i L2 10.1.1.103/32 [115/40] via 10.1.1.103, 00:00:04, tunnel-t31
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:04, tunnel-t32
i L2 10.1.3.103/32 [115/40] via 10.1.3.103, 00:00:04, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.1.103, 00:00:04, tunnel-t31
i L2 10.2.0.104/32 [115/50] via 10.1.1.103, 00:00:04, tunnel-t31
```

When multiple restricted AA tunnels are created towards the same destination IP address, router load balances traffic across all those tunnels:

```
Router# show route
i L2 10.1.1.103/32 [115/40] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/40] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
i L2 10.1.2.103/32 [115/40] via 10.1.2.103, 00:00:08, tunnel-t32
                   [115/40] via 10.1.2.103, 00:00:30, tunnel-t34
i L2 10.1.3.103/32 [115/40] via 10.1.3.103, 00:00:08, tunnel-t33
i L2 10.2.0.103/32 [115/40] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/40] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
i L2 10.2.0.104/32 [115/50] via 10.1.1.101, 00:00:08, GigabitEthernet0/0/0/2
                   [115/50] via 10.1.3.101, 00:00:08, GigabitEthernet0/0/0/3
```

Configure MPLS over UDP

Table 36: Feature History Table

Feature Name	Release Information	Feature Description
Configure MPLS over UDP	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Configure MPLS over UDP	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D

Feature Name	Release Information	Feature Description
Configure MPLS over UDP	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Configure MPLS over UDP	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Configure MPLS over UDP	Release 7.5.3	You can now enable MPLS over UDP, where a dynamic tunnel is created with the BGP next hop that helps in traffic distribution along the path. Traffic distribution considers parameters such as IP source address, IP destination address, and UDP source port along different tunnels for hashing. This feature introduces the following new commands: • **set encapsulation-type mpls-udp** • **overlay-encapsulation mpls-udp**

There are different encapsulation methods to segregate traffic along the path, and one such method is MPLS over UDP encapsulation. In other encapsulation methods, such as MPLS, the load-balancing decision on core routers is based on MPLS label values. And for VPN, the top label is typically an IGP label that identifies the target PE. Hence the MPLS label is the same for all flows towards that PE.

MPLS over UDP considers IP source address, IP destination address, and UDP source port for distributing traffic effectively along different paths. This good variety of parameters enhances load-balancing efficiently, which is not available in other encapsulation methods. When you enable this feature, MPLS packets are encapsulated in UDP, and the tunnels are created dynamically with the BGP next hop.

Here is the encapsulation format for MPLS over UDP:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Source Port = Entropy      |       Dest Port = MPLS        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           UDP Length          |        UDP Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      MPLS Label Stack                         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
~                         Message Body                          ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

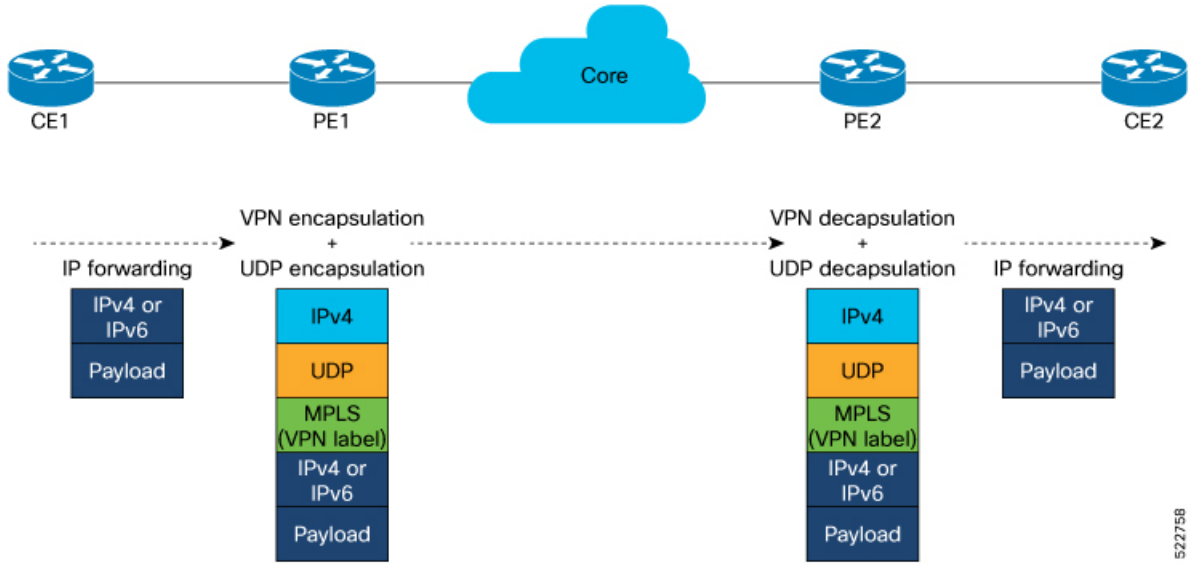
where,

- Source IP address is the tunnel source address.
- Destination IP address is the tunnel destination address, which is the BGP next-hop.
- UDP source port is the entropy value (allowed range is from 49152 through 65535).
- UDP destination port is 6635 used for MPLS payload.

Topology

Let's consider the following topology to understand MPLS over UDP.

Figure 16: MPLS over UDP



In this topology, traffic is sent from CE1 to CE2.

- When PE1 receives traffic from CE1.
- PE1 adds the MPLS label.
- PE1 adds a UDP flow label header. The UDP header contains a source and a destination port.
  - When the next-hop is defined as MPLS over UDP, a dynamic tunnel is created and the traffic is sent through this tunnel to the core router.
  - If next hop is not enabled with MPLS over UDP, regular IPv4 or IPv6 routing lookup is performed.
- The core router forwards the packet to PE2.
- PE2 decapsulates the packet and forwards it to the CE2 router.

You can define a set of decap-prefix lists and filter incoming packet source IP addresses using a set of specified decap-prefix. If the source IP address matches, then packets are terminated for the UDP tunnel. If the source IP address does not match, then packets are discarded without tunnel termination. When the decap-prefix list is not defined, all the incoming MPLS over UDP traffic is allowed regardless of their source IP addresses.

Restriction for Configuring MPLS over UDP

- Supported on routers with line cards based on Q200 Silicon.
- MPLS over UDP is not supported over IPv6 underlay.
- MPLS over UDP is not supported with permit ACEs in ACLs configured on underlay interfaces.

Configure MPLS over UDP

Perform the following tasks to configure MPLS over UDP:

- Configure interface NVE
- Enable MPLS over UDP
- Configure BGP
- Define prefix-set for encapsulation

Configuration Example

```
/* Configure interface NVE and enable MPLS over UDP. */
Router#configure
Router(config)# interface nve1
Router(config-if)# overlay-encapsulation mpls-udp
Router(config-if)# source-interface Loopback0
Router(config-if)# logging events link-status
Router(config-if)# commit
Router(config-if)# exit

/* Configure BGP */
Router(config)#router bgp 100
Router(config-bgp)#nsr
Router(config-bgp)#bgp router-id 10.1.1.1
```

```
Router(config-bgp)#bgp graceful-restart
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths ibgp 16
Router(config-bgp-af)#address-family vpnv4 unicast
Router(config-bgp-af)#nexthop route-policy MPLSoUDP-Encap-1
Router(config-bgp-af)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths ebgp 10
Router(config-bgp-af)#maximum-paths ibgp 16
Router(config-bgp-af)#address-family vpnv6 unicast
Router(config-bgp-af)#nexthop route-policy MPLSoUDP-Encap-1
Router(config-bgp-af)#neighbor 192.0.2.1
Router(config-bgp-nbr)#remote-as 100
Router(config-bgp-nbr)#update-source Loopback0
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#address-family vpnv4 unicast
Router(config-bgp-nbr-af)#address-family vpnv6 unicast
Router(config-bgp-nbr-af)#vrf vrf100
Router(config-bgp-vrf)#rd 100:1
Router(config-bgp-vrf)#address-family ipv4 unicast
Router(config-bgp-vrf-af)#label mode per-vrf
Router(config-bgp-vrf-af)#maximum-paths ebgp 16
Router(config-bgp-vrf-af)#maximum-paths ibgp 16
Router(config-bgp-vrf-af)#redistribute connected
Router(config-bgp-vrf-af)#address-family ipv6 unicast
Router(config-bgp-vrf-af)#label mode per-vrf
Router(config-bgp-vrf-af)#maximum-paths ebgp 16
Router(config-bgp-vrf-af)#maximum-paths ibgp 16
Router(config-bgp-vrf-af)#redistribute connected
Router(config-bgp-vrf-af)#commit


/* Define prefix-set for encapsulation*/
Router(config)#prefix-set BGP-NH-1
Router(config-pfx)# 192.0.2.1/32 le 32,
Router(config-pfx)# 192.0.3.1/32 le 32,
Router(config-pfx)# 192.0.4.1/32 le 32,
Router(config-pfx)# 192.0.5.1/32 le 32
Router(config-pfx)#exit
Router(config)#commit

Router(config)#route-policy MPLSoUDP-Encap-1
Router(config-rpl)#if next-hop in BGP-NH-1 then
Router(config-rpl-if)#set encapsulation-type mpls-udp
Router(config-rpl-if)#else
Router(config-rpl-else)#pass
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
Router(config)#commit
```

The following configuration example shows how to filter the source IP address for decapsulation. The incoming packet source IP that is part of the object-group is decapsulated:

```
Router#configure
Router(config)#object-group network ipv4 decap_2
Router(config-object-group-ipv4)#192.0.2.1/32
Router(config-object-group-ipv4)#192.0.3.1/32
Router(config-object-group-ipv4)#192.0.4.1/32
Router(config-object-group-ipv4)#192.0.5.1/32
Router(config-object-group-ipv4)#commit
Router(config-object-group-ipv4)#root
Router(config)#nve
```

```
Router(config-nve)#decap-prefix source ipv4 prefix-1
Router(config-nve-decap-src-prefix-ipv4)#object-group decap_2
Router(config-nve-decap-src-prefix-ipv4)#commit
RP/0/RP0/CPU0:PE1(config-nve-decap-src-prefix-ipv4)#root
RP/0/RP0/CPU0:PE1(config)#exit
```

Running Configuration

```
Router#show run

interface nve1
 overlay-encapsulation mpls-udp
 source-interface Loopback0
 logging events link-status
!

prefix-set BGP-NH-1
  192.0.2.1/32 le 32,
  192.0.3.1/32 le 32,
  192.0.4.1/32 le 32,
  192.0.5.1/32 le 32
end-set
!
route-policy MPLSoUDP-Encap-1
  if next-hop in BGP-NH-1 then
    set encapsulation-type mpls-udp
  else
    pass
  endif
end-policy
!
router bgp 100
 nsr
 bgp router-id 10.10.1.1
 bgp graceful-restart
 address-family ipv4 unicast
  maximum-paths ibgp 16
 !
 address-family vpnv4 unicast
  nexthop route-policy MPLSoUDP-Encap-1
 !
 address-family ipv6 unicast
  maximum-paths ebgp 10
  maximum-paths ibgp 16
 !
 address-family vpnv6 unicast
  nexthop route-policy MPLSoUDP-Encap-1
 !
 neighbor 192.0.2.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
 !
 vrf vrf100
  rd 100:1
  address-family ipv4 unicast
   label mode per-vrf
   maximum-paths ebgp 16
```

```
   maximum-paths ibgp 16
   redistribute connected
  !
  address-family ipv6 unicast
   label mode per-vrf
   maximum-paths ebgp 16
   maximum-paths ibgp 16
   redistribute connected
  !
 !
!
mpls traffic-eng
 attribute-set path-option P1
  signalled-bandwidth 10000 class-type 0
  path-selection
   cost-limit 1000
   delay-limit 5000
   exclude AAA
  !
  affinity include YELLOW
  affinity include-strict BLUE GREEN
  affinity exclude RED
  affinity exclude-all
 !
 named-tunnels
  tunnel-te cisco
   path-option 1
    attribute-set P1
   !
  !
 !
!
end
```

Running configuration for source adress filtering for decapsulation:

```
object-group network ipv4 decap_2
 192.0.2.1/32
 192.0.3.1/32
 192.0.4.1/32
 192.0.5.1/32
!
nve
 decap-prefix source ipv4 prefix-1
  object-group decap_2
 !
!
```

Verification

Verify that the MPLS over UDP is configured.

```
Router#show nve interface detail
Interface: nve1 State: Up Encapsulation: mpls-udp
Source Interface: Loopback0 (primary: 10.1.1.1)
Source Interface State: Up
NVE Flags: 0xa1, Admin State: Up, Interface Handle 0xf000014
UDP Port: 6635
Flags decode: ready, IM syn, SOC
LTEP Publish Info:
Last Del: NA
```

Verify that the tunnel type is MPLS UDP.

```
Router#show route vrf vrf100 10.1.3.1 detail
Routing entry for 10.1.3.1/32
Known via "bgp 100", distance 200, metric 0, type internal
Installed Aug 24 06:52:47.539 for 11:49:38
Routing Descriptor Blocks
192.0.2.1, from 192.0.2.1
Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000
Route metric is 0
Label: 0x10 (16)
Tunnel ID: None
Binding Label: None
Extended communities count: 0
Source RD attributes: 0x0000:100:1
NHID:0x0(Ref:0)
IP Tunnel Info: Auto create: 1 Tunnel Type: mpls-udp
MPLS eid:0x1
Route version is 0x1 (1)
No local label
IP Precedence: Not Set
QoS Group ID: Not Set
Flow-tag: Not Set
Fwd-class: Not Set
Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_REMOTE
Download Priority 3, Download Version 334
Route eid: 0x1
No advertising protos.
RP/0/RP0/CPU0:R1#
```

```
Router#show cef tunnel encap

Tunnel-type: mpls-udp, Dest addr: 192.0.2.1, Underlay Table: 0xe0000000
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000

Tunnel-type: mpls-udp, Dest addr: 192.0.3.1, Underlay Table: 0xe0000000
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000

Tunnel-type: mpls-udp, Dest addr: 192.0.4.1, Underlay Table: 0xe0000000
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000
```

The following show commands are used to verify the decapsulation:

```
Router#show cef global tunnel decap-nve

Overlay Table: 0xe0000000, Nve-If-ID: 1
LTEP: Tunnel-type: mpls-udp, Src addr: 10.1.1.1, Underlay Table: 0xe0000000

Router#show rib opaques ltep


Summary of LTEP opaque data in IPv4 RIB:

Opaque key: LTEP Overlay Table: 0xe0000000, IF ID: 1
Opaque data: LTEP Type: mpls-udp, NVE IFH: 0xf00001c, Source: 10.1.1.1, Underlay Table:
0xe00
00000, UDP Dest. Port: 6635

Router#show ofa objects iptnldecap object-count location 0/1/CPU0

Table [IPTNLDECAP] has 1 entries in DB
```

Improved scale for MPLS over UDP tunnels

Table 37: Feature History Table

Feature Name	Release Information	Feature Description
Improved scale for MPLS over UDP tunnels	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Improved scale for MPLS over UDP tunnels	Release 25.3.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100]; Centralized Systems (8600 [ASIC:Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) The **hw-module profile cef mplsoudp scale** command is replaced with **hw-module profile cef iptunnel scale** command. The output for **show hw-module profile cef** command is updated to display Iptunel scale.
Improved scale for MPLS over UDP tunnels	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) This feature is now supported on: • 8712-MOD-M
Improved scale for MPLS over UDP tunnels	Release 24.4.1	Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q200, P100]) To handle high traffic volume, you can enhance the MPLS over UDP tunnel scale up to 15284 tunnels using the **hw-module profile cef mplsoudp scale** command. The feature introduces these changes: **CLI**: • **hw-module profile cef mplsoudp scale**

MPLS over UDP is a networking technique that encapsulates MPLS (Multiprotocol Label Switching) packets within UDP (User Datagram Protocol) packets.

This allows the transport of Layer 3 (L3) traffic over MPLS over UDP tunnels. For more information, see Configure MPLS over UDP, on page 204.

For IOS XR releases prior to 24.4.1, MPLS over UDP supports a maximum of 1000 tunnels.

From IOS XR Release 24.4.1 onwards, you can configure MPLS over UDP feature to support a tunnel scale of up to 15284 tunnels using the **hw-module profile cef mplsoudp scale** command. This allows the router to handle high traffic volume and perform better load balancing.

Starting from Cisco IOS XR Release 25.3.1 onwards, the **hw-module profile cef mplsoudp scale** command is replaced with **hw-module profile cef iptunnel scale** command.

After you enable or disable the MPLS over UDP tunnel scale enhancement feature, you must reload the line cards using the **reload location all** command for the configuration to take effect.

Limitations and restrictions for improved scale for MPLS over UDP tunnels

- **Maximum Tunnels**
  - Supports up to 15284 tunnels.
- **ASIC Compatibility**
  - Only line cards and routers with the Q200 and P100 based Silicon One ASIC support the MPLS over UDP tunnel scale enhancement feature.
  - Line cards and routers with the Q100 based Silicon One ASIC don’t support the MPLS over UDP tunnel scale enhancement feature.
  - Don’t configure MPLS over UDP tunnel scale enhancement feature on the line cards and routers that have both Q100 and Q200 based Silicon One ASIC operating in compatible mode.
  - For Modular Systems (8800) with only Q200 based Silicon One ASIC, configure the Route Processor (RP) to Q200 mode before configuring MPLS over UDP tunnel scale enhancement feature.
- **Reload Requirement**
  - After you enable or disable the MPLS over UDP tunnel scale enhancement feature, you must reload the line cards with the **reload location all** command for the configuration to take effect.

Configure improved scale for MPLS over UDP tunnels

Perform these steps to configure improved scale for MPLS over UDP tunnels:

Before you begin

- For Modular Systems (8800) with only Q200 based Silicon One ASIC, configure the RP to Q200 mode. For more information, see Configuring the Compatibility Mode for Various ASIC Types.
- Configure the MPLS over UDP tunnel feature. For more information, see Configure MPLS over UDP, on page 204.

Procedure

Step 1 Configure the **hw-module profile cef iptunnel scale** command to increase the maximum scale for MPLS over LDP tunnels.

Example:

```
Router# configure
Router(config)# hw-module profile cef iptunnel scale
```

Step 2 Reload the line cards with the **reload location all** command to enable increased scale limits for MPLS over UDP tunnels.

```
Router(config)# reload location all
Router(config)# commit
```

Step 3 To view the configuration status of CEF hardware-modules, use the **show hw-module profile cef** command.

```
Router# show hw-module profile cef

Fri Aug 22 13:23:17.796 UTC
-----------------------------------------------------------------
Knob                          Status          Applied   Action
-----------------------------------------------------------------
CBF Enable                    Unconfigured    N/A       None
CBF forward-class-list        Unconfigured    N/A       None
BGPLU                         Unconfigured    N/A       None
LPTS ACL                      Unconfigured    N/A       None
Dark Bandwidth                Unconfigured    N/A       None
SR-OPT                        Unconfigured    N/A       None
IP Redirect Punt              Unconfigured    N/A       None
IPv6 Hop-limit Punt           Unconfigured    N/A       None
MPLS Per Path Stats           Unconfigured    N/A       None
SRv6 Per Path Stats           Unconfigured    N/A       None
Tunnel TTL Decrement          Unconfigured    N/A       None
High-Scale No-LDP-Over-TE     Unconfigured    N/A       None
Label over TE counters        Unconfigured    N/A       None
Highscale LDPoTE No SRoTE     Unconfigured    N/A       None
LPTS Pifib Entry Counters     Unconfigured    N/A       None
Unipath surpf                 Unconfigured    N/A       None
Source-based rtbh             Unconfigured    N/A       None
Vxlan ipv6 tunnel scale       Unconfigured    N/A       None
Encap Exact                   Unconfigured    N/A       None
Iptunnel scale                Configured      Yes       None
BGPLU over RSVPTE Enable      Unconfigured    N/A       None
MPLS Decap Stats              Unconfigured    N/A       None
```

Common Label for IPv4 and IPv6 Address Families

Table 38: Feature History Table

Feature Name	Release Information	Feature Description
Common MPLS Label for IPv4 and IPv6 Address Families	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Common MPLS Label for IPv4 and IPv6 Address Families	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M
Common MPLS Label for IPv4 and IPv6 Address Families	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
Common MPLS Label for IPv4 and IPv6 Address Families	Release 24.1.1	This release simplifies network operations and reduces the complexity of maintaining separate MPLS labels for IPv4 and IPv6 address families by enabling you to assign a single MPLS label to a Virtual Routing and Forwarding (VRF) instance common to both address families. You can allocate the label statically and dynamically. Previously, you could only allocate one label per address family. The feature introduces these changes: **CLI:** • The **per-vrf-46** keyword is introduced in the **label mode** command. • Modified the output of the following show commands: • **show bgp labels** • **show bgp process** • **show bgp vrf** **YANG Data Models:** `Cisco-IOS-XR-um-router-bgp-cfg.yang` (see GitHub, YANG Data Models Navigator)

You have the flexibility to enable the **per-vrf-46** mode either globally or at the VRF level. Dynamic allocation of **per-vrf-46** MPLS label is supported in global routing table (default VRF) and at the VRF level. Static allocation of **per-vrf-46** MPLS label is supported only at the VRF level. If a static configuration is enabled after a dynamic MPLS label assignment, the static label takes precedence over the dynamic label.

Furthermore, you can allocate the **per-vrf-46** label mode globally under the VPNv4 and VPNv6 unicast address families. The VRFs inherit the configured label mode unless a label mode is explicitly configured under a VRF.

To use the same label for advertising both IPv4 and IPv6 prefixes in a VRF or global routing table, you must configure the **per-vrf-46** label allocation mode explicitly under each address family. Enabling this mode in only one address family limits the allocated VPN label to that address family's advertisements.

Guidelines and Limitations for Using Common Label for IPv4 and IPv6 Address Families

The guidelines and restrictions for using the **per-vrf-46** label modes:

- The allocation of a static label is limited to a VRF. Dynamic allocation is supported for the global routing table and VRF.
- Unconfiguring the **per-vrf-46** label allocation mode from one address family only impacts that address family.
- If another address family has the **per-vrf-46** label allocation mode, the MPLS label remains reserved for use in that address family. When the last address family on a VRF has the **per-vrf-46** label allocation mode unconfigured, the allocated MPLS label gets released.
- In the case of static label allocation discrepancies, you must clear the discrepancy by running the **clear mpls static local-label discrepancy <label-value>** command.

Configure a Common Label for IPv4 and IPv6 Address Families

BGP configurations

Perform the following steps to configure **per-vrf-46** label mode:

- To enable the **per-vrf-46** label mode for global table IPv4 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv4 unicast
Router(configure-bgp-af)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for global table IPv6 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv6 unicast
Router(configure-bgp-af)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for global table VPNv4 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family vpnv4 unicast
Router(configure-bgp-af)#vrf all
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for global table VPNv6 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family vpnv6 unicast
Router(configure-bgp-af)#vrf all
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for VRF IPv4 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv4 unicast
Router(configure-bgp-af)#vrf INET
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

- To enable the **per-vrf-46** label mode for VRF IPv6 unicast:

```
Router(configure)#router bgp 65550
Router(configure-bgp)#address-family ipv6 unicast
Router(configure-bgp-af)#vrf INET
Router(configure-bgp-af-vrfall)#label mode per-vrf-46
```

MPLS Static Configurations

To allocate a static label to a VRF:

```
Router(configure)#mpls static
Router(config-mpls-static)#vrf Test
Router(config-mpls-static-vrf)#lsp Test-LSP
Router(config-mpls-static-vrf-lsp)#in-label 24100 allocate per-vrf-46
```

Verifications

Verfiy that the **per-vrf-46** label mode is enabled on your router.

- To verify the configurations:
  - RP/0/RP0/CPU0:tb13-r1#show bgp vpnv4 unicast process | i abel
    VRF all label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1#show bgp vpnv6 unicast process | i abel
    VRF all label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1#show bgp vrf INET ipv4 unicast process | i abel
    Label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1#show bgp vrf INET ipv6 unicast process | i abel
    Label alloc mode: per-vrf-46
  - RP/0/RP0/CPU0:tb13-r1# show bgp vrf INET ipv4 unicast labels | i 192.0.2.0/24
    *>192.0.2.0/24 198.51.100.0 24200 nolabel
  - RP/0/RP0/CPU0:tb13-r3#show bgp vrf INET ipv4 unicast labels | i 192.0.2.0/24
    *> 192.0.2.0/24 0.0.0.0 nolabel 24200
  - RP/0/RP0/CPU0:tb13-r1#show bgp vrf INET ipv6 unicast labels | i 2001:DB8::/32
    *>2001:DB8::/32 198.51.100.0 24200 nolabel
  - RP/0/RP0/CPU0:tb13-r3#show bgp vrf INET ipv6 unicast labels | i 2001:DB8::/32
    *> 2001:DB8::/32 :: nolabel 24200

Policy-based tunnel selection

Table 39: Feature History Table

Feature Name	Release Information	Feature Description
Policy-based tunnel selection	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.

Feature Name	Release Information	Feature Description
Policy-based tunnel selection	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Policy-based tunnel selection	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Policy-based tunnel selection	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Enhanced policy-based tunnel selection support for recursive prefixes	Release 7.5.4	We now support forwarding VPN traffic using policy based tunnel selection (PBTS) and ensure that traffic like L3VPN, IPv6 Provider Edge or IPv6 VPN Provider Edge (6PE or 6VPE) services flow as intended. This is possible because we have enhanced PBTS to support recursive prefix traffic.

Feature Name	Release Information	Feature Description
Policy-based tunnel selection	Release 7.5.3	You can now set the traffic preference based on Policy-Based Tunnel Selection (PBTS) to steer traffic to specific MPLS tunnels to optimize service distribution. It also aids in bandwidth utilization by steering traffic towards the same destination through different paths, independent of IGP metric. PBTS lets you direct traffic into specific TE tunnels based on class-based forwarding of IP (DSCP marking) or MPLS (EXP marking) packets to carry voice and data traffic through the MPLS network. This feature is supported only on routers with line cards based on Q200 Silicon. This feature introduces the following commands: • **forward-class** • **set forward-class** • **hw-module profile cef cbf forward-class-list** • **cef pbts class**

In a network, there are various traffic types such as data, voice, and video. All these traffic types are sent with equal importance to reach a destination through the MPLS tunnel. But different types of traffic needs different priority.

You can classify and prioritize traffic types using Quality of Service. In a packet, there are fields that show the traffic types and classification is performed based on the different fields of a packet. For example:

- IP Precedence value
- Differentiated Services Code Point (DSCP) values
- Source and destination IP address

After the traffic types are classified, QoS marking is performed at different levels to prioritize traffic. For IP traffic, IP precedence or DSCP values are used to mark the packets. For MPLS traffic, EXP values are used to mark these headers for packet marking.

For more Information, see Chapter *Mark Packets to Change Priority Settings* in *Modular QoS Configuration Guide for Cisco 8000 Series Routers*.

You can now use DSCP or EXP to classify and steer traffic to a specific MPLS tunnels using Policy-Based Tunnel Selection (PBTS).

PBTS selects the tunnel based on the classification criteria of the incoming packets, which are based on the experimental (EXP), differentiated services code point (DSCP), or type of service (ToS) field in the packet and forwards traffic to TE tunnels associated with that class-map.

A class-map is defined for various types of packet, and associates this class-map with a forward-class. The class-map defines the matching criteria for classifying a particular type of traffic, while the forward-class defines the forwarding path these packets should take. After a class-map is associated with a forwarding-class in the policy-map, all the packets that match the class-map are forwarded as defined in the policy-map. The egress TE tunnel interfaces route traffic based on the forwarding-class. For each forwarding-class, specify the TE interface explicitly or implicitly in the case of default value with the forward group.

Default-class for paths is always zero (0). If there's no MPLS TE tunnel for a given forward-class, then the default-class (0) is applied. If there's no default-class, then the packet is dropped.

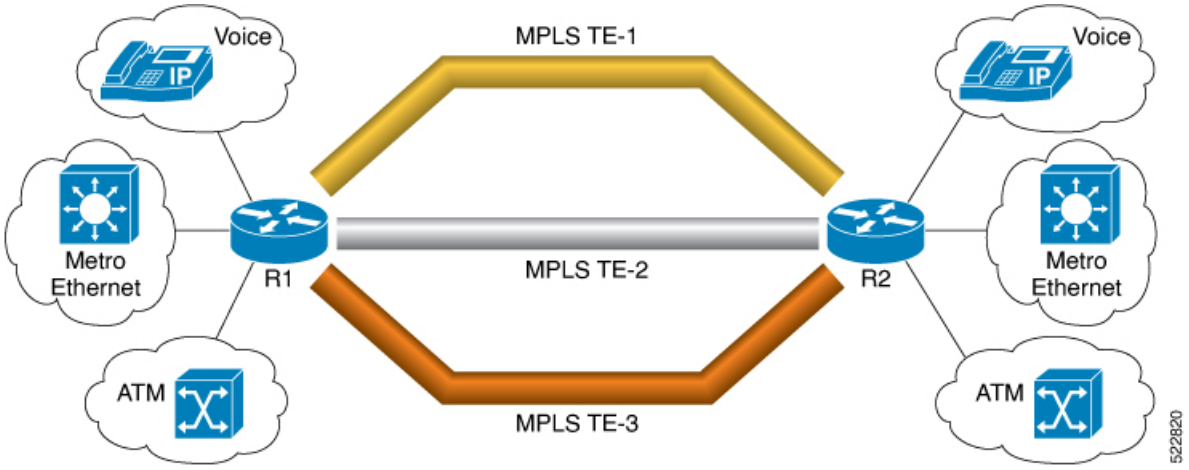
Table 40: Packet Classification

Packet Classification	
EXP	The MPLS experimental bits (EXP) field is a 3-bit field in the MPLS header that defines the QoS treatment on a node for a packet to classify and mark network traffic You can set the EXP value from 0 through 7 on a single MPLS TE tunnel. Value 0 is best effort forwarding and 7 is for highest priority. Router identifies the EXP field values in the topmost MPLS label as match criteria for a class map. By choosing different values for the MPLS EXP field, you can mark packets so that packets have the priority that they require during congestion.
DSCP	DiffServ Code Point (DSCP) is a 6-bit field that defines a class and drop precedence. Routers within your network can then use the newly marked DSCP precedence values to determine how the traffic should be treated. You can set the DSCP values for packets entering the network. You can set DSCP values from 0 through 63. Zero is used for best effort forwarding, 1 is used for medium effort forwarding, and 2 for high priority forwarding. The rest of the values are for specific applications such as VOIP calling forwarding, video conference forwarding, and so on.

Topology

Let's understand how the traffic types are classified and steered into a specific tunnel using the EXP marking for MPLS traffic in the following topology:

Figure 17: Policy-Based Tunnel Selection



In this topology, there are three MPLS-TE tunnels: TE1, TE2, and TE3.

These tunnels are associated with a specific forward-class and marked with different EXP marking to forward traffic.

The tunnels forward the traffic based on the forward-class and its associated EXP marking.

- MPLS TE1 is indicated in Gold color that carries voice traffic associated with forward-class 1 and EXP value 1.
- MPLS TE2 is indicated in Silver color that carries Metro Ethernet and ATM traffic associated with forward-class 2 and EXP value 2.
- MPLS TE3 is indicated in Bronze color that carries all other traffic associated with forward-class 0, which is a default class to take all other EXP traffic.

When PBTS is configured on R1and R2, traffic associated with EXP 1 is sent through the MPLS TE1. Similarly, traffic associated with EXP 2 is sent through the MPLS TE2.

Traffic associated other than EXP 1 and 2 is sent through the Bronze MPLS TE3.

Similarly, IP traffic is classified and prioritized using DSCP value.

Note When the TE tunnel associated with the forward-class goes down, you can use the **cef pbts class** command to redirect traffic to another forward-class.

When **cef pbts class any fallback-to any** is specified, then the lowest available forward-class in ascending order of 0-7 is chosen as fallback when paths to any class go down.

Guidelines and Limitations

- Only one forward-class can be associated with a TE tunnel at any time.
- Eight unique forwarding-class values are supported.

This feature supports the following prefixes:

- IPv4 and IPv6 prefixes for LDP over TE
- IPv4 and IPv6 prefixes for RSVP-TE
- Statically configured IPv4 or IPv6 prefix to LDP over TE tunnel destination
- Statically configured IPv4 or IPv6 prefix to TE tunnel destination

Restrictions

- This feature is supported only on routers with line cards based on Q200 Silicon.
- Statically configured MPLS prefix over LDP over TE tunnel is not supported.
- Statically configured MPLS prefix over TE tunnel is not supported.
- Recursive prefixes are not supported for LDP over TE tunnels that are enabled with PBTS.
- Recursive prefixes are not supported for TE tunnels that are enabled with PBTS.

All prefixes advertised by BGP are recursive, that is there's no directly associated next-hop interface. Router performs a lookup in the routing table to find the next-hop interface.

Starting from Cisco IOS XR Release 7.5.4, the Policy-Based Tunnel Selection feature supports the recursive prefixes for the following services:

- L3VPN, 6VPE, 6PE services and overlays over recursive prefixes through the TE tunnel.
- BGP prefixes over TE tunnel.
- Statically configured MPLS prefix over LDP through TE tunnel.
- Statically configured MPLS prefix over TE tunnel.
- Recursive prefixes for LDP over TE tunnels that are enabled with PBTS.
- Recursive prefixes for TE tunnels that are enabled with PBTS.

Configure Policy-Based Tunnel Selection

Perform the following tasks to configure PBTS on the edge routers:

- Classify traffic using IP DSCP or EXP marking
- Associate forward-class
- Apply service-policy to ingress interface
- (Optional) Configure forward-class PBTS using Hardware profile
- Create TE-tunnels to carry traffic based on priority
- Associate egress TE tunnels to forward-class

Configuration Example

The following example shows how to configure PBTS with EXP marking:

```
/* Classify traffic using EXP marking */
Router#configure
Router(config)#class-map match-any exp-1
Router(config-cmap)#match mpls experimental topmost 1
Router(config-cmap)#end-class-map

Router(config)#class-map match-any exp-2
Router(config-cmap)#match mpls experimental topmost 2
Router(config-cmap)#end-class-map

Router(config)#class-map match-any exp-3
Router(config-cmap)#match mpls experimental topmost 3
Router(config-cmap)#end-class-map
Router(config)#commit

/* Associate forward-class */
Router(config)#policy-map INGRESS-POLICY
Router(config-pmap)#class exp-1
Router(config-pmap-c)#set forward-class 1
Router(config-pmap-c)#exit

Router(config-pmap)#class exp-2
Router(config-pmap-c)#set forward-class 2
Router(config-pmap-c)#exit

Router(config-pmap)#class exp-3
Router(config-pmap-c)#set forward-class 3
Router(config-pmap-c)#exit
Router(config-pmap)#exit

/* Apply service-policy to ingress interface */
Router(config)#interface gigabitEthernet 0/0/0/1
Router(config-if)#service-policy input INGRESS-POLICY
Router(config-if)#ipv4 address 10.1.1.1 255.255.255.0
Router(config-if)#exit

/* (Optional) Configure forward-class PBTS using Hardware profile */
Router(config)#hw-module profile cef cbf forward-class-list 1 2 3

/* Create TE-tunnels to carry traffic based on priority */
Router(config)#interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)#signalled-bandwidth 1000
Router(config-if)#autoroute announce
Router(config-if)#destination 10.50.50.1
Router(config-if)#record-route

/* Associate egress TE tunnels to forward-class */
Router(config-if)#forward-class 1
Router(config-if)#path-option 1 dynamic
Router(config-if)#exit
```

The following example shows how to configure PBTS with DSCP marking:

```
Router#configure
Router(config)#class-map match-any AF41-Class
/* Classify traffic using IP DSCP or EXP marking*/
Router(config-cmap)#match dscp AF41
Router(config-cmap)#exit

/* Associate forward-class */
```

```
Router(config)#policy-map INGRESS-POLICY
Router(config-pmap)#class AF41-Class
Router(config-pmap-c)#set forward-class 1
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)#commit

/* Apply service-policy to ingress interface */
Router(config)#interface GigabitEthernet0/0/0/1
Router(config-if)#service-policy input INGRESS-POLICY
Router(config-if)#exit

/* (Optional) Configure forward-class PBTS using Hardware profile */
Router(config)#hw-module profile cef cbf forward-class-list 1
Router(config)#reload

/* Create TE-tunnels to carry traffic based on priority */
Router(config)#interface tunnel-te1
Router(config-if)#ipv4 unnumbered Loopback0
Router(config-if)#signalled-bandwidth 1000
Router(config-if)#autoroute announce
Router(config-if-tunte-aa)#destination 10.50.50.1
Router(config-if)#record-route

/* Associate egress TE tunnels to forward-class */
Router(config-if)#forward-class 1
Router(config-if)#path-option 1 dynamic
Router(config-if)#commit
Router(config)#exit
```

Running Configuration

This example shows the running show configuration for the EXP markings:

```
class-map match-any exp-1
 match mpls experimental topmost 1
 end-class-map
!
class-map match-any exp-2
 match mpls experimental topmost 2
 end-class-map
!
class-map match-any exp-3
 match mpls experimental topmost 3
 end-class-map
!
class-map match-any AF41-Class
 match dscp af41
 end-class-map
!
policy-map INGRESS-POLICY

 class exp-1
  set forward-class 1
 !
 class exp-2
  set forward-class 2
 !
 class exp-3
  set forward-class 3
 !
 class class-default
```

```
 !
 end-policy-map
!
interface tunnel-te1
 ipv4 unnumbered Loopback0
 signalled-bandwidth 1000
 autoroute announce
 !
 destination 10.50.50.1
 record-route
 forward-class 1
 path-option 1 dynamic
!
interface preconfigure GigabitEthernet0/0/0/1
 service-policy input INGRESS-POLICY
 ipv4 address 10.1.1.1 255.255.255.0
!
hw-module profile cef cbf forward-class-list 1 2 3
end
```

This example shows the running show configuration for the DSCP markings:

```
class-map match-any AF41-Class
 match dscp af41
 end-class-map
!
policy-map INGRESS-POLICY
 class AF41-Class
  set forward-class 1
 !
 class class-default
 !
 end-policy-map
!
interface tunnel-te1
 ipv4 unnumbered Loopback0
 signalled-bandwidth 1000
 autoroute announce
 !
 destination 10.50.50.1
!

interface preconfigure GigabitEthernet0/0/0/1
 service-policy input INGRESS-POLICY
!
hw-module profile cef cbf forward-class-list 1 2 3
end
```

Verification

Verify the PBTS path based on IP information:

```
Router#show cef 10.1.1.1/24

10.1.1.1/24, version 2236, internal 0x1000001 0x30 (ptr 0x940b61b0) [1], 0x600 (0x931456e8),
 0xa20 (0xa98eb318)
 Updated Oct 11 00:01:09.999
 Prefix Len 16, traffic index 0, precedence n/a, priority 3, encap-id 0x1001200000001
  gateway array (0x92fa97f0) reference count 6, flags 0x68, source lsd (5), 1 backups
                [3 type 5 flags 0x208401 (0x94c71960) ext 0x0 (0x0)]
  LW-LDI[type=5, refc=3, ptr=0x931456e8, sh-ldi=0x94c71960]
  gateway array update type-time 1 Oct 11 00:01:09.982
 LDI Update time Oct 11 00:01:09.992
```

```
LW-LDI-TS Oct 11 00:01:09.999
  via 202.158.0.2/32, tunnel-te1, 7 dependencies, weight 0, forward class 1 [flags 0x0]
   path-idx 0 NHID 0x0 [0xa933e300 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_4, 5 dependencies, weight 0, forward class 4 [flags 0x0]
   path-idx 1 NHID 0x0 [0xa933e9f0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_5, 7 dependencies, weight 0, forward class 5 [flags 0x0]
   path-idx 2 NHID 0x0 [0xa933ec40 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_6, 7 dependencies, weight 0, forward class 6 [flags 0x0]
   path-idx 3 NHID 0x0 [0xa933ee90 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, named_7, 7 dependencies, weight 0, forward class 7 [flags 0x0]
   path-idx 4 NHID 0x0 [0xa933f0e0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, tunnel-te0, 5 dependencies, weight 0, forward class 0 [flags 0x0]
   path-idx 5 NHID 0x0 [0xa933e0b0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, tunnel-te2, 7 dependencies, weight 0, forward class 2 [flags 0x0]
   path-idx 6 NHID 0x0 [0xa933e550 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}
  via 202.158.0.2/32, tunnel-te3, 7 dependencies, weight 0, forward class 3 [flags 0x0]
   path-idx 7 NHID 0x0 [0xa933e7a0 0x0]
   next hop 202.158.0.2/32
   local adjacency
    local label 24011      labels imposed {ImplNull}

   Weight distribution:
   slot 0, weight 0, normalized_weight 1, forward class 0
   slot 1, weight 0, normalized_weight 1, forward class 1
   slot 2, weight 0, normalized_weight 1, forward class 2
   slot 3, weight 0, normalized_weight 1, forward class 3
   slot 4, weight 0, normalized_weight 1, forward class 4
   slot 5, weight 0, normalized_weight 1, forward class 5
   slot 6, weight 0, normalized_weight 1, forward class 6
   slot 7, weight 0, normalized_weight 1, forward class 7

   PBTS class information:
     forward class 0: 1 paths, offset 0
     forward class 1: 1 paths, offset 1
     forward class 2: 1 paths, offset 2
     forward class 3: 1 paths, offset 3
     forward class 4: 1 paths, offset 4
     forward class 5: 1 paths, offset 5
     forward class 6: 1 paths, offset 6
     forward class 7: 1 paths, offset 7
   Load distribution: 0 1 2 3 4 5 6 7 (refcount 3)

   Hash  OK  Interface                 Address
```

```
    0     Y   tunnel-te0                point2point
    1     Y   tunnel-te1                point2point
    2     Y   tunnel-te2                point2point
    3     Y   tunnel-te3                point2point
    4     Y   named_4                   point2point
    5     Y   named_5                   point2point
    6     Y   named_6                   point2point
    7     Y   named_7                   point2point
```

Verify the PBTS path based on MPLS LDP info:

```
Router#show mpls ldp forwarding 10.1.1.1/24


Codes:
  - = GR label recovering, (!) = LFA FRR pure backup path
  {} = Label stack with multi-line output for a routing path
  G = GR, S = Stale, R = Remote LFA FRR backup
  E = Entropy label capability

Prefix          Label   Label(s)       Outgoing     Next Hop            Flags
                In      Out            Interface                        G S R E
--------------- ------- -------------- ------------ -------------------- -------
10.1.1.1/24     24011ImpNull        tt1          202.158.0.2
                        ImpNull        named_4      202.158.0.2
                        ImpNull        named_5      202.158.0.2
                        ImpNull        named_6      202.158.0.2
                        ImpNull        named_7      202.158.0.2
                        ImpNull        tt0          202.158.0.2
                        ImpNull        tt2          202.158.0.2
                        ImpNull        tt3          202.158.0.2
```

To view the PBTS information at CEF level for a label:

```
Router#show cef mpls local-label 24011 eoS detail

Label/EOS 24011/1, Label-type LDP, version 2236, internal 0x1000001 0x30 (ptr 0x94d4ae60)
[1], 0x600 (0x931456e8), 0xa20 (0xa98eb318)
 Updated Oct 11 00:01:09.996
 Prefix Len 21, traffic index 0, precedence n/a, priority 3, encap-id 0x1001200000001
  gateway array (0x92fa97f0) reference count 6, flags 0x68, source lsd (5), 0 backups
                [3 type 5 flags 0x208401 (0x94c71960) ext 0x0 (0x0)]
  LW-LDI[type=5, refc=3, ptr=0x931456e8, sh-ldi=0x94c71960]
  gateway array update type-time 1 Oct 11 00:01:09.979
 LDI Update time Oct 11 00:01:09.989
 LW-LDI-TS Oct 11 00:01:09.996
   via 202.158.0.2/32, tunnel-te1, 7 dependencies, weight 0, forward class 1 [flags 0x0]
    path-idx 0 NHID 0x0 [0xa933e300 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, named_4, 5 dependencies, weight 0, forward class 4 [flags 0x0]
    path-idx 1 NHID 0x0 [0xa933e9f0 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, named_5, 7 dependencies, weight 0, forward class 5 [flags 0x0]
    path-idx 2 NHID 0x0 [0xa933ec40 0x0]
    next hop 202.158.0.2/32
    local adjacency
     local label 24011      labels imposed {ImplNull}
   via 202.158.0.2/32, named_6, 7 dependencies, weight 0, forward class 6 [flags 0x0]
    path-idx 3 NHID 0x0 [0xa933ee90 0x0]
    next hop 202.158.0.2/32
```

```
 local adjacency
  local label 24011      labels imposed {ImplNull}
via 202.158.0.2/32, named_7, 7 dependencies, weight 0, forward class 7 [flags 0x0]
 path-idx 4 NHID 0x0 [0xa933f0e0 0x0]
 next hop 202.158.0.2/32
 local adjacency
  local label 24011      labels imposed {ImplNull}
via 202.158.0.2/32, tunnel-te0, 5 dependencies, weight 0, forward class 0 [flags 0x0]
 path-idx 5 NHID 0x0 [0xa933e0b0 0x0]
 next hop 202.158.0.2/32
 local adjacency
  local label 24011      labels imposed {ImplNull}
via 202.158.0.2/32, tunnel-te2, 7 dependencies, weight 0, forward class 2 [flags 0x0]
 path-idx 6 NHID 0x0 [0xa933e550 0x0]
 next hop 202.158.0.2/32
 local adjacency
  local label 24011      labels imposed {ImplNull}
via 202.158.0.2/32, tunnel-te3, 7 dependencies, weight 0, forward class 3 [flags 0x0]
 path-idx 7 NHID 0x0 [0xa933e7a0 0x0]
 next hop 202.158.0.2/32
 local adjacency
  local label 24011      labels imposed {ImplNull}

 Weight distribution:
 slot 0, weight 0, normalized_weight 1, forward class 0
 slot 1, weight 0, normalized_weight 1, forward class 1
 slot 2, weight 0, normalized_weight 1, forward class 2
 slot 3, weight 0, normalized_weight 1, forward class 3
 slot 4, weight 0, normalized_weight 1, forward class 4
 slot 5, weight 0, normalized_weight 1, forward class 5
 slot 6, weight 0, normalized_weight 1, forward class 6
 slot 7, weight 0, normalized_weight 1, forward class 7

 PBTS class information:
   forward class 0: 1 paths, offset 0
   forward class 1: 1 paths, offset 1
   forward class 2: 1 paths, offset 2
   forward class 3: 1 paths, offset 3
   forward class 4: 1 paths, offset 4
   forward class 5: 1 paths, offset 5
   forward class 6: 1 paths, offset 6
   forward class 7: 1 paths, offset 7
 Load distribution: 0 1 2 3 4 5 6 7 (refcount 3)

 Hash  OK  Interface                Address
 0     Y   tunnel-te0               point2point
 1     Y   tunnel-te1               point2point
 2     Y   tunnel-te2               point2point
 3     Y   tunnel-te3               point2point
 4     Y   named_4                  point2point
 5     Y   named_5                  point2point
 6     Y   named_6                  point2point
 7     Y   named_7                  point2point
```

Verify if PBTS policy-map is in use:

```
Router#show qos interface tenGigE 0/0/0/0/0 input

NOTE:- Configured values are displayed within parentheses
Interface TenGigE0/0/0/0/0 ifh 0xf000678  -- input policy
NPU Id:                        0
Total number of classes:       2
Interface Bandwidth:           10000000 kbps
Policy Name:                INGRESS-POLICY
```

```
Accounting Type:              Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   prec_7
Policer not configured for this class

Level1 Class                             =   class-default
Policer not configured for this class
```

Verify whether the traffic is taking the correct tunnel-te using the **show interfaces accounting** command. The value of the `pkts out` field increases if traffic is forwarding through the specified tunnel.

```
Router#show interfaces tunnel-te named_4 accounting
named_4
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  MPLS                        0                0        87496          9624560
```

Configure IPv6 routing over IPv4 MPLS-TE tunnels

Perform these steps to configure IPv6 routing over IPv4 MPLS-TE tunnels:

Procedure

Step 1 **configure**

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

Example:

```
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
```

Configures an MPLS-TE tunnel interface.

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

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv4 unnumbered Loopback 0
```

Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is the commonly-used interface type.

Step 4 **ipv6 enable**

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv6 enable
```

Enables IPv6 on interface.

Step 5 **signalled-bandwidth** *bandwidth*

Example:

```
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10
```

Sets the tunnel bandwidth requirement to be signalled in Kbps.

Step 6 **destination** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-if)#destination 192.168.0.1
```

Specifies the IPv4 destination address.

Step 7 Use one of these options:

- **autoroute announce include-ipv6**
- **forwarding-adjacency include-ipv6**

Example:

```
RP/0/RP0/CPU0:router(config-if)#autoroute announce include-ipv6
```

Or

```
RP/0/RP0/CPU0:router(config-if)#forwarding-adjacency include-ipv6
```

Announces the tunnel as an IPv6 autoroute, or as an IPv6 forwarding adjacency.

Step 8 **path-option** *preference-priority* **dynamic**

Example:

```
RP/0/RP0/CPU0:router(config-if)#path-option 1 dynamic
```

Sets the path option to dynamic and assigns the path ID.

Step 9 **commit**

Step 10 (Optional) **show mpls traffic-eng autoroute**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng autoroute

  Destination 192.168.0.2 has 1 tunnels in IS-IS ring level 1
    tunnel-te1 (traffic share 0, nexthop 192.168.0.2)
                (IPv4 unicast)
                (IPv6 unicast)
```

Verifies that the tunnel announces IPv6 autoroute information.

Step 11 (Optional) **show mpls traffic-eng forwarding-adjacency**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng forwarding-adjacency

 destination 192.168.0.1 has 1 tunnels

              tunnel-te10    (traffic share 0, next-hop 192.168.0.1)
                             (Adjacency Announced: yes, holdtime 0)
                             (IS-IS 100, IPv4 unicast)
                             (IS-IS 100, IPv6 unicast)
```

Verifies that the tunnel announces IPv6 forwarding adjacency information.

Configure IPv6 routing over IPv6 MPLS-TE tunnels

Perform these steps to configure IPv6 routing over IPv6 MPLS-TE tunnels:

Procedure

Step 1 **configure**

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

Example:

```
RP/0/RP0/CPU0:router(config)# interface tunnel-te 1
```

Configures an MPLS-TE tunnel interface.

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

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv4 unnumbered Loopback 0
```

Assigns a source address so that forwarding can be performed on the new tunnel. Loopback is the commonly-used interface type.

Step 4 **ipv6 enable**

Example:

```
RP/0/RP0/CPU0:router(config-if)#ipv6 enable
```

Enables IPv6 on interface.

Step 5 **signalled-bandwidth** *bandwidth*

Example:

```
RP/0/RP0/CPU0:router(config-if)# signalled-bandwidth 10
```

Sets the tunnel bandwidth requirement to be signalled in Kbps.

Step 6 Use one of these options:

- **autoroute announce include-ipv6**
- **forwarding-adjacency include-ipv6**

Example:

```
RP/0/RP0/CPU0:router(config-if)#autoroute announce include-ipv6
```

Or

```
RP/0/RP0/CPU0:router(config-if)#forwarding-adjacency include-ipv6
```

Announces the tunnel as an IPv6 autoroute, or as an IPv6 forwarding adjacency.

Step 7 **destination ipv6** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-if)#destination ipv6 2000::1
```

Specifies the IPv6 destination address.

Step 8 **path-option** *preference-priority* **dynamic**

Example:

```
RP/0/RP0/CPU0:router(config-if)#path-option 1 dynamic
```

Sets the path option to dynamic and assigns the path ID.

Step 9 **commit**

Step 10 (Optional) **show mpls traffic-eng autoroute**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng autoroute

  Destination 192.168.0.2 has 1 tunnels in IS-IS ring level 1
    tunnel-te1 (traffic share 0, nexthop 192.168.0.2)
               (IPv4 unicast)
               (IPv6 unicast)
```

Verifies that the tunnel announces IPv6 autoroute information.

Step 11 (Optional) **show mpls traffic-eng forwarding-adjacency**

Example:

```
RP/0/RP0/CPU0:router#show mpls traffic-eng forwarding-adjacency

 destination 192.168.0.1 has 1 tunnels

              tunnel-te10    (traffic share 0, next-hop 192.168.0.1)
                             (Adjacency Announced: yes, holdtime 0)
                             (IS-IS 100, IPv4 unicast)
                             (IS-IS 100, IPv6 unicast)
```

Verifies that the tunnel announces IPv6 forwarding adjacency information.

Event history for MPLS-TE tunnels

MPLS-TE tunnel event history provides network diagnostic records that

- chronologically document significant events and actions that occur within MPLS-TE headend tunnels
- enable detailed tracking of the tunnel lifecycle
- analyze tunnel lifecycle changes that also include RSVP signaling events, and
- facilitate monitoring, troubleshooting, and compliance by recording each event with timestamps and associated metadata.

Table 41: Feature History Table

Feature Name	Release	Feature Description
Event history for MPLS-TE headend tunnels	Release 25.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC:Q200]) ; Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) This feature introduces a granular and configurable event tracking mechanism for operational visibility into MPLS-TE tunnels. You can enable event history with the **mpls traffic-eng event-history tunnel** command and set the number of tracked events with the **mpls traffic-eng event-history tunnel event-count** command. This feature introduces these changes: CLI: • Three new keywords, **event-history**, **tunnel**,and **event-count**, are added to the **mpls traffic-eng** command. • Two new keywords, **event-history**, and **tunnels** are added to the **show mpls traffic-eng** command. • A new **clear mpls traffic-eng event-history tunnels** command has been added. • The **show mpls traffic-eng counters global** and **show mpls traffic-eng chunks** commands have been modified to include the event history information.

Cisco IOS XR Release 25.4.1 introduces the tunnel event history for MPLS-TE tunnels to track and analyse the timeline of events in the tunnel lifecycle, including RSVP signaling. This feature enables enhanced operational visibility for MPLS-TE tunnels by providing granular, configurable event tracking.

To enable the MPLS-TE tunnel event history feature, use the **mpls traffic-eng event-history tunnel** command. To modify the event count for the tunnel event history, use the **mpls traffic-eng event-history tunnel event-count** command. By default, there are 20 events in a tunnel. You can configure up to 1000 events per tunnel.

Benefits of tunnel event history

You can use the event history feature for MPLS-TE headend tunnels to

- log events in dedicated, per-tunnel buffers for fair and isolated event recording,
- compress duplicate events to minimize log size and clutter, and
- enhance debugging with accessible historical records for troubleshooting.

Restrictions for tunnel event history

The tunnel event history feature is:

- Disabled by default.
- Supported only on MPLS-TE headend tunnels.

Configure tunnel event history and event count

This procedure helps you to configure the tunnel event history and the event count.

Procedure

Step 1 Enable the tunnel event history.

Example:

```
Router# configure
Router(config)# mpls traffic-eng event-history tunnel
Router(config-te-evt-hist)# commit
```

Step 2 Specify the event count.

Example:

```
Router# configure
Router(config)# mpls traffic-eng event-history tunnel event-count 45
Router(config-te-evt-hist)# commit
```

The default number of events is 20. You can set up to 1000 events per tunnel.

Step 3 Verify the tunnel event history configuration.

Example:

This output shows the event history for all tunnels.

```
Router# show mpls traffic-eng event-history tunnels all
Thu Dec 18 07:30:41.116 UTC
Name: tunnel-te1
 Event history (oldest first):
 ID                   Time                 LSP  Event-Type  Rep  Event-Message
```

```
 01765952238563713907  Dec 17 06:17:50.600     2  Path          0  BW:300 kbps ERO: 10.1.3.103
10.3.5.105
 01765952238563713909  Dec 17 06:17:50.703     2  Resv          0  BW:300 kbps RRO:
 01765952238563713910  Dec 17 06:17:50.703     2  Oper State    0  New tunnel state: UP

Name: tunnel-T10
 Event history (oldest first):
 ID                    Time                  LSP  Event-Type  Rep  Event-Message
 01765952238563713933  Dec 17 07:15:28.220     5  Path Tear     0
 01765952238563713934  Dec 17 07:15:32.660     8  Path          0  BW:300 kbps ERO: 10.1.2.102
10.2.4.104
 01765952238563713935  Dec 17 07:15:32.763     8  Resv          0  BW:300 kbps RRO:
 01765952238563713936  Dec 17 07:15:32.803     6  Path Tear     0
 01765952238563713937  Dec 17 07:15:37.387     9  Path          0  BW:300 kbps ERO: 10.1.2.102
10.2.4.104
 01765952238563713938  Dec 17 07:15:37.480     9  Resv          0  BW:300 kbps RRO:
 01765952238563713939  Dec 17 07:15:37.517     7  Path Tear     0
 01765952238563713940  Dec 17 07:15:47.193    10  Path          0  BW:300 kbps ERO: 10.1.2.102
10.2.4.104
 01765952238563713941  Dec 17 07:15:47.283    10  Resv          0  BW:300 kbps RRO:
 01765952238563713942  Dec 17 07:15:47.320     8  Path Tear     0
 01765952238563713943  Dec 17 07:16:07.420     9  Path Tear     0
 01765952238563713944  Dec 17 07:16:15.573    11  Path          0  BW:300 kbps ERO: 10.1.2.102
10.2.4.104
 01765952238563713945  Dec 17 07:16:15.670    11  Resv          0  BW:300 kbps RRO:
 01765952238563713946  Dec 17 07:16:35.810    10  Path Tear     0
 01765952238563713947  Dec 17 07:22:49.863    12  Path          0  BW:300 kbps ERO: 10.1.2.102
10.2.4.104
 01765952238563713948  Dec 17 07:22:49.963    12  Resv          0  BW:300 kbps RRO:
 01765952238563713949  Dec 17 07:23:10.103    11  Path Tear     0
 01765952238563713950  Dec 17 07:23:49.240    13  Path          0  BW:300 kbps ERO: 10.1.2.102
10.2.4.104
 01765952238563713951  Dec 17 07:23:49.340    13  Resv          0  BW:300 kbps RRO:
 01765952238563713952  Dec 17 07:24:09.480    12  Path Tear     0

Name: tunnel-A
 Event history (oldest first):
 ID                    Time                  LSP  Event-Type  Rep  Event-Message
 01765952238563713856  Dec 17 06:17:38.293    2  Path         0  BW:0 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713877  Dec 17 06:17:38.413     2  Resv          0  BW:0 kbps RRO:
 01765952238563713894  Dec 17 06:17:38.463     2  Oper State    0  New tunnel state: UP

Name: tunnel-B
 Event history (oldest first):
 ID                    Time                  LSP  Event-Type  Rep  Event-Message
 01765952238563713857  Dec 17 06:17:38.293    2  Path         0  BW:0 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713873  Dec 17 06:17:38.413     2  Resv          0  BW:0 kbps RRO:
 01765952238563713891  Dec 17 06:17:38.457     2  Oper State    0  New tunnel state: UP

Name: tunnel-C
 Event history (oldest first):
 ID                    Time                  LSP  Event-Type  Rep  Event-Message
 01765952238563713858  Dec 17 06:17:38.293    2  Path         0  BW:0 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713878  Dec 17 06:17:38.413     2  Resv          0  BW:0 kbps RRO:
 01765952238563713895  Dec 17 06:17:38.463     2  Oper State    0  New tunnel state: UP

Name: SELFPING-101
 Event history (oldest first):
 ID                    Time                  LSP  Event-Type  Rep  Event-Message
 01769076278173248001  Jan 22 05:30:13.487    2  Path         0  BW:0 kbps ERO: 10.10.10.2 17.17.17.2
 17.17.17.3 22.22.22.3 22.22.22.4 192.168.0.4
```

```
 01769076278173248002  Jan 22 05:30:13.546     2  Resv          0  BW:0 kbps RRO:
 01769076278173248003  Jan 22 05:30:23.551     2  Self-Ping     0  Timed-out (started at Jan 22
05:01:30.769)
 01769076278173248004  Jan 22 05:30:23.554     2  Oper State    0  New tunnel state: UP

Name: SELFPING-103
 Event history (oldest first):
 ID                    Time                   LSP  Event-Type  Rep  Event-Message
 01769076278173248011  Jan 22 05:31:54.518    2  Path         0  BW:0 kbps ERO: 10.10.10.2 17.17.17.2
 17.17.17.3 22.22.22.3 22.22.22.4 192.168.0.4
 01769076278173248012  Jan 22 05:31:54.581     2  Resv          0  BW:0 kbps RRO:
 01769076278173248013  Jan 22 05:32:02.917     2  Self-Ping     0  Aborted (started at Jan 22
05:01:30.769)
 01769076278173248014  Jan 22 05:32:02.925     2  Path Tear     0
 01769076278173248015  Jan 22 05:32:02.936    3  Path         0  BW:0 kbps ERO: 11.11.11.2 17.17.17.2
 17.17.17.3 22.22.22.3 22.22.22.4 192.168.0.4
 01769076278173248016  Jan 22 05:32:02.987     3  Resv          0  BW:0 kbps RRO:
 01769076278173248017  Jan 22 05:32:06.994     3  Self-Ping     0  Succeeded (started at Jan 22
05:01:30.769)
 01769076278173248018  Jan 22 05:32:06.995     3  Oper State    0  New tunnel state: UP
```

Example:

This output shows the event history for a specific tunnel.

```
Router# show mpls traffic-eng event-history tunnels name tunnel-T10
Thu Dec 18 07:30:57.167 UTC
Name: tunnel-T10
 Event history (oldest first):
 ID                    Time                   LSP  Event-Type  Rep  Event-Message
 01765952238563713933  Jan 1 07:15:28.220     5  Path Tear     0
 01765952238563713934  Jan 1 07:15:32.660    8  Path         0  BW:300 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713935  Jan 1 07:15:32.763     8  Resv          0  BW:300 kbps RRO:
 01765952238563713936  Jan 1 07:15:32.803     6  Path Tear     0
 01765952238563713937  Jan 1 07:15:37.387    9  Path         0  BW:300 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713938  Jan 1 07:15:37.480     9  Resv          0  BW:300 kbps RRO:
 01765952238563713939  Jan 1 07:15:37.517     7  Path Tear     0
 01765952238563713940  Jan 1 07:15:47.193    10  Path         0  BW:300 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713941  Jan 1 07:15:47.283    10  Resv          0  BW:300 kbps RRO:
 01765952238563713942  Jan 1 07:15:47.320     8  Path Tear     0
 01765952238563713943  Jan 1 07:16:07.420     9  Path Tear     0
 01765952238563713944  Jan 1 07:16:15.573    11  Path         0  BW:300 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713945  Jan 1 07:16:15.670    11  Resv          0  BW:300 kbps RRO:
 01765952238563713946  Jan 1 12:23:05.711    10  Self-Ping     0  Succeeded (started at Jan  1
12:23:05.709)
 01765952238563713947  Jan 1 07:16:35.810    10  Path Tear     0
 01765952238563713948  Jan 1 07:22:49.863    12  Path         0  BW:300 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713949  Jan 1 07:22:49.963    12  Resv          0  BW:300 kbps RRO:
 01765952238563713950  Jan 1 07:23:10.103    11  Path Tear     0
 01765952238563713951  Jan 1 07:23:49.240    13  Path         0  BW:300 kbps ERO: 10.1.2.102 10.2.4.104

 01765952238563713952  Jan 1 07:23:49.340    13  Resv          0  BW:300 kbps RRO:
 01765952238563713953  Jan 1 07:24:09.480    12  Path Tear     0
 01765952238563713954  Jan 1 12:45:45.473    11  Self-Ping     0  Succeeded (started at Jan  1
12:45:45.472)
```

Here,

- ID—Specifies the unique ID for the event.

- Time—Specifies the timestamp when the event occurred.
- LSP—Specifies the LSP ID associated with the event.
- Event-Type—Specifies the type of the event. Some of the XR events are Path, Path Tear, Path Err, Resv, and so on.
- Rep—Specifies the number of times an event with the same data has occurred consecutively.
- Event-Message—Specifies information about specific events. For example, the event message string contains Record-route object (RRO) for a reservation (RESV) event.

Example:

This output shows the specified event count.

```
Router# show run mpls traffic-eng event-history
Thu Jan  1 19:26:00.479 UTC
mpls traffic-eng
 event-history
  tunnel
   event-count 45
  !
 !
!
```

Example:

This example shows the event history memory consumption.

```
Router# show mpls traffic-eng counters global

Wed Dec 3 14:09:38.405 UTC

Global Statistics:
Number of VIF : 2
Number of LSP : 2
Number of S2L : 2

Checkpoint request queue high watermark : 3
Checkpoint response queue high watermark : 16
Checkpoint reschedules : 28
Checkpoint enqueues (request queue) : 86
Checkpoint dequeues (request queue) : 86
Checkpoint enqueues (response queue) : 86
Checkpoint dequeues (response queue) : 86
Checkpoint write (VIF) : 32
Checkpoint read (VIF) : 0
Checkpoint write (DST) : 0
Checkpoint read (DST) : 0
Checkpoint write (S2L) : 4
Checkpoint read (S2L) : 0
Checkpoint write (Autobackup) : 0
Checkpoint read (Autobackup) : 0
Checkpoint write (PI backup) : 0
Checkpoint read (PI backup) : 0
Number of RSVP connection : 1
Number of RSVP disconnection : 0
Number of RSVP restart time change : 0
Number of LSD connection : 1
Number of LSD disconnection : 0
Number of LSD replay : 1
Number of LMRIB connection : 1
Number of LMRIB disconnection : 0
Number of LMRIB restart time change : 1
Number of LMRIB restart failed : 0
```

```
Number of LMRIB restart passed : 1
Number of LMRIB recovery failed : 0
Number of LMRIB recovery passed : 1
Number of RSI connection : 1
Number of RSI disconnection : 0
Event-history size (bytes)
 Tunnels (excluding chunks) : 16444912
```

Example:

This example displays the memory information for the tunnel event history.

```
Router# show mpls traffic-eng chunks

  Tue Dec  9 06:30:40.055 UTC
Pool name             InUse     Free      Allocs    Deallocs  Sib creat Sib trimm
-------------------- --------- --------- --------- --------- --------- ---------
mpls wavl Chunks             0     32000         0         0         0         0
mpls wavl Chunks            28      7972        28         0         0         0
mpls wavl Chunks           101      3899       101         0         0         0
mpls wavl Chunks            59      1941        59         0         0         0
mpls wavl Chunks            27       973        27         0         0         0
mpls wavl Chunks             9       491         9         0         0         0
mpls wavl Chunks             0       240         0         0         0         0
mpls wavl Chunks             0       120         0         0         0         0
TE S2L chunk                 0       200         0         0         0         0
TE S2L stats chunk           0      1024         0         0         0         0
TE LSP chunk                 0       200         0         0         0         0
TE collab chunk             15        25        15         0         0         0
TE FSM Q chunk               0       128         0         0         0         0
TE NextHop Route Po          0       128         0         0         0         0
TE Binding Sid Cont          0       128         0         0         0         0
TE CHKPT Q Elements          0       600        10        10         0         0
RSVP CHKPT LIST Chu          0      8192         0         0         0         0
TE Tunnel Service R          0       128         0         0         0         0
TE Tunnel Service R          0       128         0         0         0         0
RSVP API CTX                 0       156         0         0         0         0
RSVP API CTX stats           0        48         0         0         0         0
RSVP API CTX event           0      1248         0         0         0         0
RSVP API CTX event           0      1248         0         0         0         0
RSVP API CTX Histor          0      1248         0         0         0         0
TE S2L RW chunk              0       124         0         0         0         0
TE S2L RW chunk              0       128         0         0         0         0
TE Event History             0       200         0         0         0         0
VIF Pulse Context c          0      1024       668       668         0         0
```

Step 4 (Optional) Clear the tunnel event history for all tunnels, if required.

Example:

```
Router# clear mpls traffic-eng event-history tunnels all
```

Dynamic bandwidth update for MPLS-TE LSPs

In standard networking, when you need to increase the size or bandwidth of a reserved data path or an LSP (Label Switched Path), the system typically uses the IETF standard called make-before-break (MBB), where you need to create new LSPs, shift traffic to new LSPs, and tear down old LSPs.

However, in scenarios where both the old and new LSPs traverse the same path, you can increase the bandwidth value over an existing LSP.

Table 42: Feature History Table

Feature Name	Release Information	Feature Description
Increase bandwidth for existing MPLS-TE LSP instances	Release 26.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100])(select variants only*); Centralized Systems (8600 [ASIC:Q200]) ; Modular Systems (8800 [LC ASIC: Q100, Q200, P100])(select variants only*) This feature provides a more efficient method to increase the bandwidth of established LSPs (Label Switched Paths) thus eliminating the overhead of creating entirely new signaling instances. This feature allows the same LSP instance to be signaled with a new bandwidth value. The feature introduces these changes: CLI: • A new keyword, **in-place-bandwidth-update**, has been added to the **show mpls traffic-eng counters**. • A new keyword, **in-place-bandwidth-update**, has been added to the **clear mpls traffic-eng counters**.

Starting with Cisco IOS XR Release 26.1.1, in contrary to the traditional MBB operation, this feature allows to increase the bandwidth value of the same LSP instance. This feature is enabled by default on the mid-node and tail-end LSPs in MPLS-TE tunnels.

A new keyword, **in-place bandwidth-update**, has been added to the **show mpls traffic-eng counters** and **clear mpls traffic-eng counters** commands.

Benefits of existing LSP bandwidth update

Increasing the existing MPLS-TE LSP bandwidth provides the following benefits:

- Reduced signaling churn: By avoiding the creation and teardown of new LSPs for simple bandwidth adjustments, the load on the control plane is significantly reduced.
- Elimination of label reprogramming: There is no need to program new tunnel labels at transit LSRs if the path remains the same, leading to faster updates.

- Seamless traffic management: It eliminates the need for the ingress LER to shift traffic from an old LSP to a new one, avoiding potential micro-losses or the complexities of "LSP self-ping" and install timers.
- Optimized resource utilization: Benefits over-provisioned networks where bandwidth increase may succeed on the existing path.

Restrictions for existing MPLS-TE LSP bandwidth update

- The bandwidth update feature for existing MPLS-TE LSPs is supported only on transit (mid-point) LSRs and tail-end LERs.
- The bandwidth update feature is supported only for point-to-point (P2P) LSPs.
- The bandwidth update feature is not supported for GMPLS tunnels.
- If the bandwidth update for an existing MPLS-TE LSP fails, the headend node as per its defined behavior can perform one of the following actions:
  - Make-before-break (MBB)
  - Resignaling with a different bandwidth value through the same LSP
  - Tear down the existing LSP

Regular expressions in MPLS-TE tunnel names

In large scale MPLS-TE tunnel deployments, regular expressions (regex) can provide a scalable way to filter tunnels for visibility, run actions on group of tunnels, and avoid issuing commands one tunnel at a time.

Table 43: Feature History Table

Feature Name	Release Information	Feature Description
Regular expressions in MPLS-TE tunnel names	Release 26.2.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100])(select variants only*); Centralized Systems (8600 [ASIC:Q200]) ; Modular Systems (8800 [LC ASIC: Q100, Q200, P100])(select variants only*) This feature enhances operational scalability and usability by allowing regex-based tunnel name filtering for MPLS-TE show and execution commands, without changing configuration models, architecture, or security posture. This feature supports regular expression in all MPLS-TE show and execution commands using tunnel names.

Starting from Cisco IOS XR Release 26.2.1, you can use regular expressions in execution and show commands with MPLS-TE tunnel names to filter multiple tunnels at a time.

Regular expressions enable operators to quickly filter show command output for MPLS-TE tunnel names, allowing them to

- view all tunnels for a specific region, customer, or service
- reduce noise in large outputs, and
- troubleshoot issues more quickly during outages or convergence events

By using regular expressions, operators can identify all related MPLS-TE tunnels at once, streamlining operational tasks across large-scale deployments.

Benefits of using regular expressions in MPLS-TE tunnel names

- Manage all related tunnels at once in large scale deployments.
- Apply actions to multiple tunnels in a single command, such as reoptimizing paths, rerouting tunnels, and clearing auto-bandwidth state.
- Leverage existing naming conventions without changing tunnel configuration.
- Allow fine-grained selection preventing incorrect matches in large MPLS-TE networks.

Limitations of using regex in MPLS-TE tunnels

The gNMI, Netconf, and programmability interfaces do not support regular expressions.

Usage guidelines of regex in MPLS-TE tunnel names

Follow these guidelines to use regex with MPLS-TE tunnel names:

- Tunnel names support the dot character ('.'). However, using `.` as a regex in tunnel names does not match any single character. To filter tunnel names in this way, use regex patterns such as `b[0-9a-zA-Z_-]r` or `b(.)r`.

  `.*` and `.+` are exceptions that are considered as regular expressions matching any character N times.

- When your regex includes '(', ')', or '|', enclose the pattern in double quotes. Without double quotes, it will be reported as an unsupported character.

```
Router# mpls traffic-eng reoptimize name "(FROM-nyc.*)"
Router# mpls traffic-eng reoptimize name "(FROM-nyc|To-LA)"
```

- Regex patterns containing backslash (\) do not work because the IOS XR parser removes all backslash characters from the string before processing.
  - Use `[0-9]` instead of `\d`
  - Use `[A-Za-z0-9_]` instead of `\w`
  - Using a dot (.) matches literal `'.'`, and there is no need to use `\.`

    Example using `\w`:

```
Router# show mpls traffic-eng tunnels name fo.\w brief
TUNNEL NAME         DESTINATION      STATUS  STATE
       fo.w         192.168.0.3          up     up
```

    Example using `.`(dot):

```
Router# show mpls traffic-eng tunnels name fo.o brief

TUNNEL NAME         DESTINATION      STATUS  STATE
       fo.o         192.168.0.3          up     up

Router# show mpls traffic-eng tunnels name fo[.f]o brief

TUNNEL NAME         DESTINATION      STATUS  STATE
       fo.o         192.168.0.3          up     up
       fofo         192.168.0.3          up     up
```

- For backward compatibility, all patterns are automatically wrapped as `^pattern$`.
- To match substrings, explicitly include wildcards such as `.*pattern.*`.

  Example:

```
/* Matches only the tunnel name "foo". */
Router# show mpls traffic-eng tunnels name foo

/* Matches all tunnels containing "foo" */
Router# show mpls traffic-eng tunnels name .*foo.*
```

- The question mark ('?') is interpreted by the CLI for command help or auto-completion. Hence, avoid using `?` in regex.

Table 44: Regex summary

Typical regex	Parser behavior	Usage
`'.'`	Literal only	Use `'.'` or character class such as `[a-zA-Z0-9_-.]`
`'(', ')', '|'`	Not allowed	Not allowed unless inside double quotes
`\`	Stripped	Use explicit character classes
`?`	CLI help character	Do not use
`^$`	Auto-applied	Explicit use has no impact
Substring match	Disabled by default	Use `.*pattern.*`

By following these guidelines, you ensure your regex patterns for MPLS-TE tunnel names are parsed and executed as intended in show and execution commands.

Regular expressions in MPLS-TE tunnel name filtering commands

You can filter and manage MPLS-TE tunnels efficiently by specifying name patterns with regular expressions in CLI commands. Regular expressions allow you to match tunnels based on their naming patterns for more effective tunnel management.

Supported show commands and execution commands

Execution commands	Show commands
mpls traffic-eng reoptimize name <*tunn-name with regex*>	**show mpls traffic-eng tunnels name** <*tunn-name with regex*> **[affinity | auto-bw | brief | detail | fsm | name-lookup | signame | source | standby | tabular]**
mpls traffic-eng tunnels resetup name <*tunn-name with regex*>	**show mpls traffic-eng event-history tunnels name** <*tunn-name with regex*>
mpls traffic-eng tunnels reroute name <*tunn-name with regex*>	**show mpls traffic-eng autoroute name** <*tunn-name with regex*> **[ip | standby]**
mpls traffic-eng tunnels reroute name <*tunn-name with regex*>	**show mpls traffic-eng autoroute -destination name** <*tunn-name with regex*> **[ip | standby]**
clear mpls traffic-eng auto-bw name <*tunn-name with regex*>	**show mpls traffic-eng tunnels protection [frr] name** <*tunn-name with regex*> **[name-lookup | standby]**
	show mpls traffic-eng tunnels name <*tunn-name with regex*> **role [all | head | middle | tail]**

Some examples include:

- Execution commands:

```
Router# mpls traffic-eng reoptimize name FROM-nyc.*
Router# mpls traffic-eng tunnels reroute name .*TO-LA
```

- Show commands:

```
Router# show mpls traffic-eng tunnels name FROM-nyc.*
Router# show mpls traffic-eng tunnels name .*TO-LA
Router# show mpls traffic-eng tunnels name .*CUSTOMER-A.*
```

These examples demonstrate how to use regular expressions, such as '.*', to match tunnels that begin with , end with, or contain specific name patterns.

MPLS-TE Features - Details

MPLS TE Fast Reroute Link and Node Protection

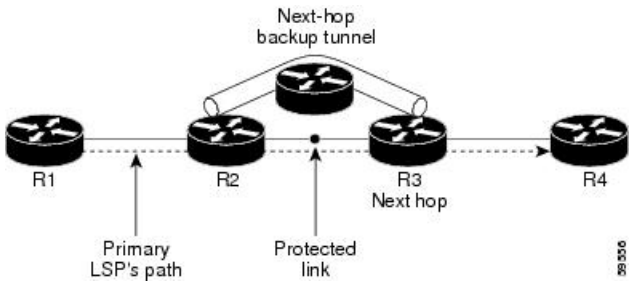
Fast Reroute (FRR) is a mechanism for protecting MPLS TE LSPs from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers try to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or node.

Note If FRR is greater than 50ms, it might lead to a loss of traffic.

Backup tunnels that bypass only a single link of the LSP's path provide link protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These tunnels are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure.

The following figure illustrates link protection.

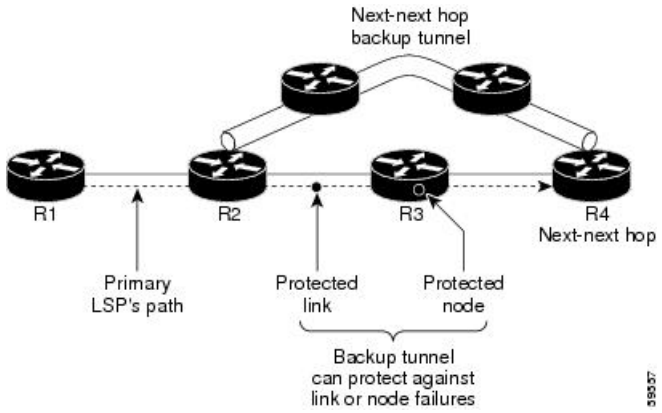
Figure 18: Link Protection



FRR provides node protection for LSPs. Backup tunnels that bypass next-hop nodes along LSP paths are called next-next-hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSP paths, bypassing the next-hop node. They protect LSPs if a node along their path fails by enabling the node upstream of the failure to reroute the LSPs and their traffic around the failed node to the next-next hop. NNHOP backup tunnels also provide protection from link failures, because they bypass the failed link and the node.

The following figure illustrates node protection.

Figure 19: Node Protection



Differentiated Services Traffic Engineering

MPLS Differentiated Services 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), you can configure multiple bandwidth constraints. These bandwidth constraints can be treated differently based on the requirement for the traffic class using that constraint.

Cisco IOS XR software supports two DS-TE modes: pre-standard and IETF. The pre-standard DS-TE mode uses the Cisco proprietary mechanisms for RSVP signaling and IGP advertisements. This DS-TE mode does not interoperate with third-party vendor equipment. Pre-standard DS-TE is enabled only after configuring the sub-pool bandwidth values on MPLS-enabled interfaces. Pre-standard DS-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 Pre-standard DS-TE mode.

IETF DS-TE mode uses IETF-defined extensions for RSVP and IGP. This mode inter-operates with third-party vendor equipment. IETF mode supports multiple bandwidth constraint models, including RDM and Maximum Allocation Bandwidth Constraint Model (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.

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.

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 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.

MPLS-TE Forwarding Adjacency

MPLS TE forwarding adjacency allows you to handle a TE label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network that is based on the Shortest Path First (SPF) algorithm. Both Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) are supported as the IGP. A forwarding adjacency can be created between routers regardless of their location in the network. The routers can be located multiple hops from each other.

As a result, a TE tunnel is advertised as a link in an IGP network with the tunnel's cost associated with it. Routers outside of the TE domain see the TE tunnel and use it to compute the shortest path for routing traffic throughout the network. 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 headend of any TE tunnels.

Automatic Bandwidth

Automatic bandwidth allows you to dynamically adjust bandwidth reservation based on measured traffic. MPLS-TE automatic bandwidth is configured on individual Label Switched Paths (LSPs) at every headend router. MPLS-TE automatic bandwidth monitors the traffic rate on a tunnel interface and 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 on either the largest average output rate that was noticed during a certain interval, or a configured maximum bandwidth value.

While re-optimizing the LSP with the new bandwidth, a new path request is generated. If the new bandwidth is not available, the last good LSP remains 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.

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 re-optimized. Then, the bandwidth samples are cleared to record the new largest output rate at the next interval. If a tunnel is shut down, and is later brought again, the adjusted bandwidth is lost, and the tunnel is brought back with the initially configured bandwidth. When the tunnel is brought back, the application period is reset.

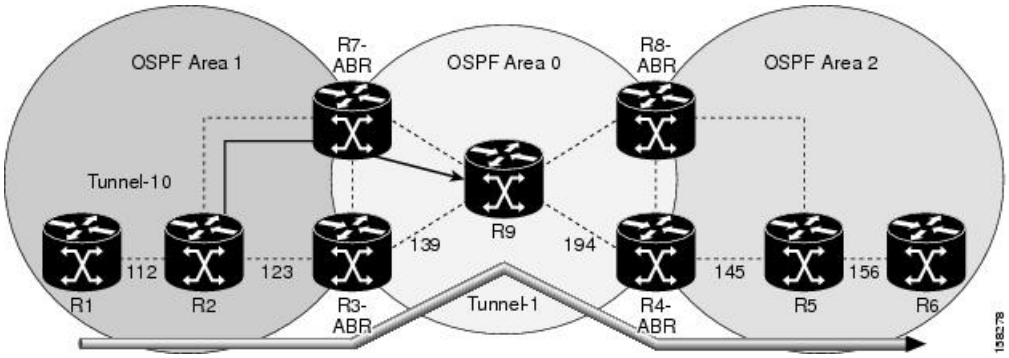
MPLS Traffic Engineering Interarea Tunneling

The MPLS-TE interarea tunneling feature allows you to establish TE tunnels spanning multiple Interior Gateway Protocol (IGP) areas and levels, thus 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. Customers running multiple IGP area backbones (primarily for scalability reasons) requires Multiarea and Interarea TE . 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.

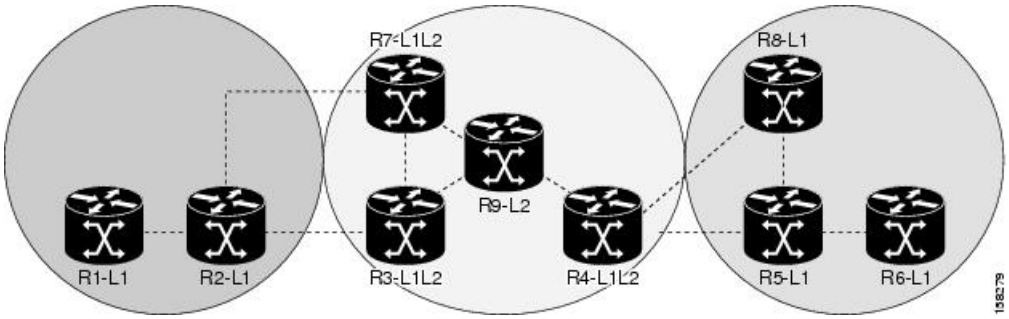
The following figure shows a typical interarea TE network using OSPF.

Figure 20: Interarea (OSPF) TE Network Diagram



The following figure shows a typical interlevel (IS-IS) TE Network.

Figure 21: Interlevel (IS-IS) TE Network Diagram



As shown in the topology, 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).

Loose hop optimization allows the re-optimization 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. Then it is 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 router). The same operation is performed by the last ABR connected to the tailend area to reach the tailend LSR.

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.

Configuring Performance Measurement

Network performance metrics such as packet loss, delay, delay variation, and bandwidth utilization is a critical measure for traffic engineering (TE) in service provider networks. These network performance metrics provide network operators information about the performance characteristics of their networks for performance evaluation and helps to ensure compliance with service level agreements. The service-level agreements (SLAs) of service providers depend on the ability to measure and monitor these network performance metrics. Network operators can use performance measurement (PM) feature to monitor the network metrics for links as well as end-to-end TE label switched paths (LSPs).

Path Calculation Metric Type

To configure the metric type to be used for path calculation for a given tunnel, use the **path-selection metric** command in either the MPLS-TE configuration mode or under the tunnel interface configuration mode.

The metric type specified per interface takes the highest priority, followed by the MPLS-TE global metric type.

Note If the delay metric is configured, CSPF finds a path with optimized*minimum* link delay metric. See the *Configuring Performance Measurement* chapter in the Segment Routing Configuration Guide for information on configuring interface performance delay measurement.

Configuration Example

The following example shows how to set the path-selection metric to use the IGP metric under a specific tunnel interface:

```
Router# configure
Router(config)# interface tunnel-te 1
Router(config-if)# path-selection metric igp
Router(config-if)# commit
```

The following example shows how to set the path-selection metric to use the delay metric under the MPLS-TE configuration mode:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# path-selection metric delay
Router(config-mpls-te)# commit
```

Path-Selection Delay Limit

Apply the **path-selection delay-limit** configuration to set the upper limit on the path aggregate delay when computing paths for MPLS-TE LSPs. After you configure the **path-selection delay-limit** value, if the sum of minimum-delay metric from all links that are traversed by the path exceeds the specified delay-limit, CSPF will not return any path. The periodic path verification checks if the delay-limit is crossed.

The **path-selection delay-limit** value can be configured at the global MPLS-TE, per-interface tunnel, and per path-option attribute set. The path-selection delay-limit per path-option attribute set takes the highest priority, followed by per-interface, and then the MPLS-TE global path-selection delay-limit values.

The delay limit range is a value from 1 to 4294967295 microseconds.

Note See the *Configuring Performance Measurement* chapter in the Segment Routing Configuration Guide for information on configuring interface performance delay measurement.

Configuration Example

The following example shows how to set the path-selection delay limit under a specific tunnel interface:

```
Router# configure
Router(config)# interface tunnel-te2000
Router(config-if)# path-selection metric delay
Router(config-if)# path-selection delay-limit 200
Router(config-if)# commit
```

The following example shows how to set the path-selection delay limit under a path-option attribute set:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# attribute-set path-option test
Router(config-te-attribute-set)# path-selection delay-limit 300
Router(config-te-attribute-set)# root
Router(config)# interface tunnel-te1000
Router(config-if)# path-option 10 dynamic attribute-set test
Router(config-if)# commit
```

The following example shows how to set the path-selection delay limit under the global MPLS-TE configuration mode:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# path-selection metric delay
Router(config-mpls-te)# path-selection delay-limit 150
Router(config-mpls-te)# commit
```

Additional References

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

Related Documents

Related Topic	Document Title
MPLS-TE commands	*MPLS Traffic Engineering Commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

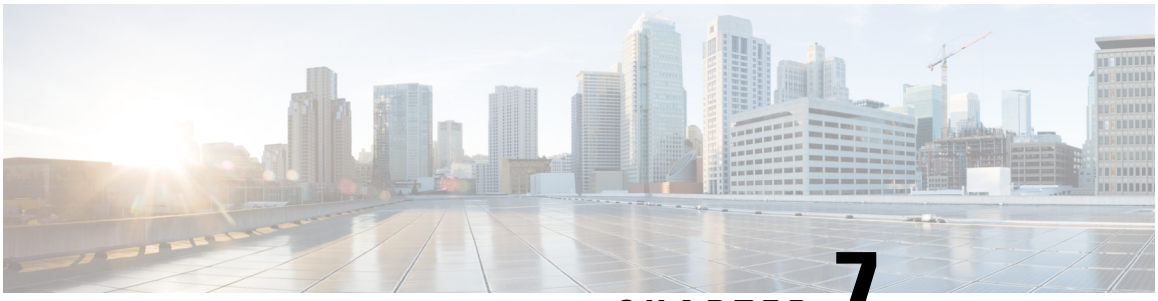
MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 4124	*Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*, F. Le Faucheur, Ed. June 2005. (Format: TXT=79265 bytes) (Status: PROPOSED STANDARD)
RFC 4125	*Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*, F. Le Faucheur, W. Lai. June 2005. (Format: TXT=22585 bytes) (Status: EXPERIMENTAL)
RFC 4127	*Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*, F. Le Faucheur, Ed. June 2005. (Format: TXT=23694 bytes) (Status: EXPERIMENTAL)

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 7

RSVP-TE Transport Solution with Flex LSP

Table 45: Feature History Table

Feature Name	Release Information	Feature Description
RSVP-TE Transport Solution with Flex LSP	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC:K100]) (select variants only*) *This feature is now supported on Cisco 8404-SYS-D routers.
RSVP-TE Transport Solution with Flex LSP	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700[ASIC:K100]); Modular Systems (8800 [LC ASIC: P100]) (select variants only*). *This feature is now supported on: • 8212-32FH-M • 8711-32FH-M • 88-LC1-12TH24FH-E • 8712-MOD-M
RSVP-TE Transport Solution with Flex LSP	Release 7.11.1	You can configure bidirectional label-switched paths dynamically through Resource Reservation Protocol-Traffic Engineering (RSVP-TE) using flex LSP. With this configuration, you can dynamically alter settings for LSPs such as QoS, bandwidth, fast reroute, and path selection to tailor the associated bidirectional LSP to network requirements and traffic patterns.

Flex LSP is the combination of static bidirectional MPLS-TP and dynamic MPLS-TE. Flex LSP provides bidirectional label switched paths (LSPs) set up dynamically through Resource Reservation Protocol–Traffic Engineering (RSVP-TE). It does not support non-co routed LSPs.

Flex 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 a co-routed associated bidirectional TE tunnel.

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.

- Signaling Methods and Object Association for Flex LSPs, on page 254
- Associated Bidirectional Co-routed LSPs, on page 254
- Restrictions for Flex LSP, on page 255
- Key Features supported in Flex LSP, on page 255
- Flex LSP Scale Details , on page 257
- How to Configure Co-routed Flex LSPs, on page 257
- RSVP-TE Transport Solution with Flex LSP Headend, on page 263
- RSVP-TE transport solution using PCE, on page 274

Signaling Methods and Object Association for Flex 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:

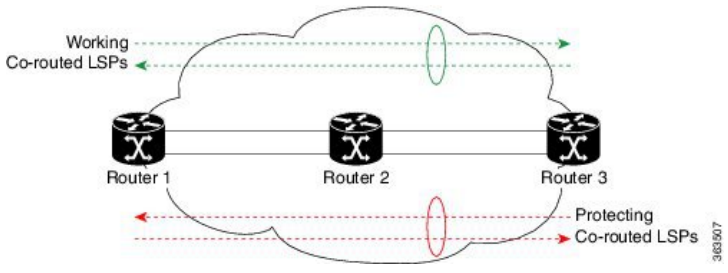
- No unidirectional LSP exists, and both must be established.
- Both unidirectional LSPs exist, but the association must be established.
- One unidirectional LSP exists, but the reverse associated LSP must be established.

Associated Bidirectional Co-routed LSPs

This section provides an overview of associated bidirectional co-routed LSPs. Establishment of MPLS TE-LSP involves computation of a path between a head-end node to a tail-end node, signaling along the path, and modification of intermediate nodes along the path. The signaling process ensures bandwidth reservation (if signaled bandwidth is lesser than 0 and programming of forwarding entries).

Path computation is performed by the head-end nodes of both the participating LSPs using Constrained Shortest Path First (CSPF). CSPF is the 'shortest path (measured in terms of cost) that satisfies all relevant LSP TE constraints or attributes, such as required bandwidth, priority and so on.

Associated Bidirectional 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 green) 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 green working LSP to Router 1.
- Router 3 sets up protecting LSP to Router 1 (in red) after performing bidirectional CSPF and sends reverse ERO to Router 1. Node Router 1 uses the received reverse ERO to set up reverse red protecting LSP to Router 3.

Restrictions for Flex LSP

- Exp-null over Flex-LSP is not supported.
- 50 msec convergence is not guaranteed without WRAP protection. WRAP protection is mandatory to achieve 50 msec convergence for remote failures.
- TE NSR and IGP NSR are mandatory for RSP switchover.
- VPLS over Flex-LSP is not supported.
- Non-co routed Flex LSP is not supported.
- Sub interface shut will not guarantee 50 msec convergence.
- MPLS forwarding table stats is not supported.
- 1000 tunnels are supported with wrap protection and path protection.

Key Features supported in Flex LSP

Following list outlines key features supported:

- **Protection -**

  Following Protection features are supported:

  - **Lockout:** Using Lockout feature, user can perform lockout protection on a selected LSP and can switch traffic to protecting LSP, if the selected LSP carries the traffic. LSP on locked out interface remains up but no traffic flows on the locked out path. To configure lockout under an MPLS-TE enabled interface:

    ```
    RP/0/RP0/CPU0:router# configure
    RP/0/RP0/CPU0:router(config)# mpls traffic-eng
    RP/0/RP0/CPU0:router(config-mpls-te)# interface tenGigE0/1/0/1
    RP/0/RP0/CPU0:router(config-mpls-te-if)# fault-oam lockout
    ```

  - **Wrap Protection:** Using Wrap Protection, each LSP signals unique wrap label for head-end to identify lookback traffic and sends it over protect LSP. To configure Wrap Protection:

    ```
    RP/0/RP0/CPU0:router# configure
    RP/0/RP0/CPU0:router(config)# interface tunnel-te1
    ```

```
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered Loopback0
RP/0/RP0/CPU0:router(config-if)# destination 49.49.49.2
RP/0/RP0/CPU0:router(config-if)# path-option 10 explicit name PATH1-2-3
RP/0/RP0/CPU0:router(config-if)# bidirectional association id 100 source-address
49.49.49.2
RP/0/RP0/CPU0:router(config-if)# bidirectional association association type co-routed
RP/0/RP0/CPU0:router(config-if)# wrap-protection
RP/0/RP0/CPU0:router(config-if)# fault-oam
```

- **MPLS-OAM:** MPLS-OAM supports single segment pseudowire going over the associated bidirectional TE tunnels. This support includes pseudowires signaled dynamically, statically, or using a mix of both modes. To configure MPLS-OAM, use the following command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# mpls oam
RP/0/RP0/CPU0:router(config-oam)# echo reply-mode control-channel allow-reverse-lsp
```

  Following features are supported:

  - **LSP Ping:** Using LSP ping, use can enable on demand ping. It supports IP encapsulation for both request and reply messages. It also performs reverse path verification.

    Following is the example of LSP Ping configuration:

```
RP/0/RP0/CPU0:router# ping mpls traffic-eng tunnel-te 1 reply mode control-channel

Tue May 21 11:04:12.211 EDT
Sending 5, 100-byte MPLS Echos to tunnel-te1,
      timeout is 2 seconds, send interval is 0 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

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/4/5 ms
```

  - **Traceroute:** Traceroute supports IP-encapsulation but not for echo reply. Following is the example of Traceroute configuration:

```
RP/0/RP0/CPU0:router# traceroute mpls traffic-eng tunnel-te 1

Tue May 21 11:06:16.056 EDT
Tracing MPLS TE Label Switched Path on tunnel-te1, timeout is 2 seconds

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

Type escape sequence to abort.
```

```
  0 10.10.10.1 MRU 1500 [Labels: 16005 Exp: 0]
L 1 10.10.10.2 MRU 1500 [Labels: implicit-null Exp: 0] 11 ms
! 2 13.13.13.4 4 ms
```

Flex LSP Scale Details

Scale details for Flex LSP:

Table 46: Supported LSPs for FLex LSP

Flex LSP with wrap protection	Head/Tail Node: 18750 LSPs Mid Node: 12500 LSPs
Flex LSP without wrap protection	Head/Tail Node: 18750 LSPs Mid Node: 75000 LSPs

How to Configure Co-routed Flex 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.

The configuration includes the following steps:

1. Enable basic MPLS Traffic Engineering on hostname PE1 and RSVP Configuration.
2. Configure Flex LSP.
3. Enable Wrap Protection.
4. Enable Fault OAM.
5. Map pseudowire to a specific Flex LSP tunnel.

Configuring Co-routed Flex LSPs

- 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.

Note Up to 1000 Flex LSP tunnels are supported.

Before you begin

Procedure

1. Enable basic MPLS Traffic Engineering on hostname PE1 and RSVP Configuration:

Configure MPLS-TE;

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface TenGigE0/9/0/0/12.1
```

Configure RSVP:

```
Router# configure
Router(config)# rsvp
Router(config-rsvp)# signalling hello graceful-restart refresh interval 3000
Router(config-rsvp)# interface TenGigE0/8/0/0/102.1 bandwidth 1000000
Router(config-rsvp)# interface TenGigE0/5/0/4.1 bandwidth 1000000
```

2. Configure Flex LSP:

```
Router# configure
Router(config)# interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 49.49.49.2
Router(config-if)# path-option 10 explicit name PATH1-2-3
Router(config-if)# bidirectional
Router(config-if-bidir)# association id 100 source-address 49.49.49.2
Router(config-if-bidir-co-routed)# association type co-routed
```

3. Wrap Protection:

For Wrap Protection:

```
Router# configure
Router(config)# interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 49.49.49.2
Router(config-if)# path-option 10 explicit name PATH1-2-3
Router(config-if)# bidirectional
Router(config-if-bidir)# association id 100 source-address 49.49.49.2
Router(config-if-bidir)# association type co-routed
Router(config-if-bidir-co-routed)# wrap-protection
```

4. Enable Fault OAM

```
Router# configure
Router(config)# interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 49.49.49.2
Router(config-if)# path-option 10 explicit name PATH1-2-3
Router(config-if)# bidirectional
Router(config-if-bidir)# association id 100 source-address 49.49.49.2
Router(config-if-bidir)# association type co-routed
Router(config-if-bidir-co-routed)# fault-oam
```

5. Map pseudowire to a specific Flex LSP tunnel:

```
Router# configure
Router(config)# l2vpn
```

```
Router(config-l2vpn)# pw-class foo
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc-mpls)# preferred-path interface tunnel-te 1
Router(config-l2vpn-pwc)# exit
Router(config-l2vpn)# exit
Router(config-l2vpn)# xconnect group gold
Router(config-l2vpn-xc)# p2p cust_one
Router(config-l2vpn-xc-p2p)# neighbor ipv4 49.49.49.2 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# pw-class foo
```

Verifying the Co-routed Flex LSP Configuration

To verify the co-routed LSP, use the **show mpls traffic-eng tunnels detail** command.

```
Router# show mpls traffic-eng tunnels 7001 detail

Name: tunnel-te7001  Destination: 104.0.0.1  Ifhandle:0x8000aa4
  Signalled-Name: Cisco8K-R11_t7001
  Status:
    Admin:    up Oper:   up (Uptime 136y10w)

    path option 1,  type explicit path01 (Basis for Setup, path weight 30 (reverse 30))
      Protected-by PO index: 2
    path option 2,  type explicit path02 (Basis for Standby, path weight 100010 (reverse
100010))
      Protected-by PO index: 1
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 10 kbps  CT0
    Creation Time: Wed Jan 11 03:08:36 2017 (136y10w ago)
  Config Parameters:
    Bandwidth:       10 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    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: 86, Source: 192.0.0.0
    Reverse Bandwidth: 10 kbps (CT0), Standby: 10 kbps (CT0)
    LSP Wrap Protection: Enabled

    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  Fault-OAM Info:
    Last Fault Msg: Clear
  SNMP Index: 25
  Binding SID: None
  Path Protection Info:
    Standby Path: User defined [explicit path option: 2],
    Last Switchover:
      136y10w ago, From LSP 14 To LSP 16
      No subcause recorded
```

```
    Reopt time remaining: 0 seconds
  Number of Switchovers 1, Standby Ready 3 times, Standby Reopt 0 times
  Lockout Info:
    Locked Out: NO
    Locked out LSP ID: 0
    Lockout Originated By: None
  LSP Wrap Protection: Enabled
    LSP Wrap Label: 24182
History:
  Reopt. LSP:
    Last Failure:
      LSP not signalled, identical to the [CURRENT] LSP
      Date/Time: Tue Jan 10 21:42:41 UTC 2017 [00:03:42 ago]
  Standby Reopt LSP:
    Last Failure:
      LSP not signalled, identical to the [STANDBY] LSP
      Date/Time: Tue Jan 10 21:42:41 UTC 2017 [00:03:42 ago]
      First Destination Failed: 104.0.0.1
  Prior LSP:
    ID: 14 Path Option: 1
    Removal Trigger: path protection switchover
Current LSP Info:
  Instance: 18, Signaling Area: IS-IS 100 level-2
  Uptime: 136y10w (since Wed Jan 11 03:09:56 UTC 2017)
  Outgoing Interface: TenGigE0/4/0/2.1, Outgoing Label: 24157
  Router-IDs: local      102.0.0.1
              downstream 107.0.0.1
  Soft Preemption: None
  SRLGs: not collected
  Path Info:
    Outgoing:
      Explicit Route:
        Strict, 1.27.1.2
        Strict, 3.67.1.2
        Strict, 3.67.1.1
        Strict, 1.46.1.2
        Strict, 1.46.1.1
        Strict, 104.0.0.1

    Record Route: Disabled
    Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                        Soft Preemption Desired: Not Set
    Reverse Associated LSP Information:
      Signaled Name: Cisco8K-R10_t7001
      Tunnel: 7001, Source: 104.0.0.1, Dest: 102.0.0.1, LSP: 9, State: Up
    Association:
      Association Type: Single Sided Bidirectional LSPs
      Association ID: 86, Source: 192.0.0.0
    Extended Association:
      Global source: 0
      Extended ID:
        0x66000001 (102.0.0.1)
        0x12 (0.0.0.18)
    Protection:
      Secondary (S): 0, Protecting (P): 0, Notification (N): 0, Oper (O): 0
      Link Flags: Any, LSP Flags: 1:N Protection with Extra-Traffic
    Reverse Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Reverse ERO:
      Explicit Route:
        Strict, 1.46.1.1
        Strict, 1.46.1.2
        Strict, 3.67.1.1
        Strict, 3.67.1.2
```

```
            Strict, 1.27.1.2
            Strict, 1.27.1.1
            Strict, 102.0.0.1

      Resv Info: None
        Record Route: Disabled
        Fspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Standby LSP Info:
      Instance: 19, Signaling Area: IS-IS 100 level-2
      Uptime: 136y10w (since Wed Jan 11 03:10:04 UTC 2017), Oper State: Up
      Outgoing Interface: TenGigE0/4/0/11.1, Outgoing Label: 24176
      Router-IDs: local      102.0.0.1
                  downstream 109.0.0.1
      Soft Preemption: None
      SRLGs: not collected
      Path Info:
        Outgoing:
          Explicit Route:
            Strict, 1.29.1.2
            Strict, 1.49.1.2
            Strict, 1.49.1.1
            Strict, 104.0.0.1

        Record Route: Disabled
        Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
        Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                            Soft Preemption Desired: Not Set
        Reverse Associated LSP Information:
          Signaled Name: Cisco8K-R10_t7001
          Tunnel: 7001, Source: 104.0.0.1, Dest: 102.0.0.1, LSP: 10, State: Up
        Association:
          Association Type: Single Sided Bidirectional LSPs
          Association ID: 86, Source: 192.0.0.0
        Extended Association:
          Global source: 0
          Extended ID:
            0x68000001 (104.0.0.1)
            0xa (0.0.0.10)
        Protection:
          Secondary (S): 0, Protecting (P): 1, Notification (N): 0, Oper (O): 0
          Link Flags: Any, LSP Flags: 1:N Protection with Extra-Traffic
      Resv Info: None
        Record Route: Disabled
        Fspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Persistent Forwarding Statistics:
      Out Bytes: 20272384
      Out Packets: 79189

LSP Tunnel 104.0.0.1 7001 [9] is signalled, Signaling State: up
  Tunnel Name: Cisco8K-R10_t7001 Tunnel Role: Tail
  InLabel: TenGigE0/4/0/2.1, 24164
  Signalling Info:
    Src 104.0.0.1 Dst 102.0.0.1, Tun ID 7001, Tun Inst 9, Ext ID 104.0.0.1
    Router-IDs: upstream   107.0.0.1
                local      102.0.0.1
    Bandwidth: 10 kbps (CT0) Priority:  7  7 DSTE-class: 0
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Incoming Address: 1.27.1.1
      Incoming:
      Explicit Route:
        Strict, 1.27.1.1
        Strict, 102.0.0.1
```

```
      Record Route: Disabled
      Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
      Reverse Associated LSP Information:
        Signaled Name: Cisco8K-R11_t7001
        Tunnel: 7001, Source: 102.0.0.1, Dest: 104.0.0.1, LSP: 18, State: Up
      Association:
        Association Type: Single Sided Bidirectional LSPs (Tie breaking slave)
        Association ID: 86, Source: 192.0.0.0
      Extended Association:
        Global source: 0
        Extended ID:
          0x66000001 (102.0.0.1)
          0x12 (0.0.0.18)
      Protection:
        Secondary (S): 0, Protecting (P): 0, Notification (N): 0, Oper (O): 0
        Link Flags: Any, LSP Flags: 1:N Protection with Extra-Traffic
    Resv Info: None
      Record Route: Disabled
      Fspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits

LSP Tunnel 104.0.0.1 7001 [10] is signalled, Signaling State: up
  Tunnel Name: Cisco8K-R10_t7001 Tunnel Role: Tail
  InLabel: TenGigE0/4/0/11.1, 24463
  Signalling Info:
    Src 104.0.0.1 Dst 102.0.0.1, Tun ID 7001, Tun Inst 10, Ext ID 104.0.0.1
    Router-IDs: upstream   109.0.0.1
                local      102.0.0.1
    Bandwidth: 10 kbps (CT0) Priority:  7  7 DSTE-class: 0
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Incoming Address: 1.29.1.1
      Incoming:
      Explicit Route:
        Strict, 1.29.1.1
        Strict, 102.0.0.1

      Record Route: Disabled
      Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
      Reverse Associated LSP Information:
        Signaled Name: Cisco8K-R11_t7001
        Tunnel: 7001, Source: 102.0.0.1, Dest: 104.0.0.1, LSP: 19, State: Up
      Association:
        Association Type: Single Sided Bidirectional LSPs (Tie breaking slave)
        Association ID: 86, Source: 192.0.0.0
      Extended Association:   Fspec: avg rate=8K, burst=1K, peak rate=8K
```

To verify the forwarding interface, use the **show mpls forwarding tunnels detail** command.

```
Router# show mpls forwarding tunnels 7001 detail

Tunnel        Outgoing    Outgoing     Next Hop        Bytes
Name          Label       Interface                    Switched
------------- ----------- ------------ --------------- ------------
tt7001            24157       Te0/4/0/2.1  1.27.1.2        0
     Updated: Jan 10 21:40:04.966
     Version: 17852, Priority: 2
     Label Stack (Top -> Bottom): { 24157 }
     Local Label: 24354
```

```
   NHID: 0x0, Encap-ID: INVALID, Path idx: 0, Backup path idx: 0, Weight: 0
   MAC/Encaps: 18/22, MTU: 1500
   Packets Switched: 0

Interface Name: tunnel-te7001, Interface Handle: 0x08000aa4, Local Label: 24354
Forwarding Class: 0, Weight: 0
Packets/Bytes Switched: 79189/20272384
```

RSVP-TE Transport Solution with Flex LSP Headend

Table 47: Feature History Table

Feature Name	Release Information	Feature Description
RSVP-TE Transport Solution with Flex LSP Headend	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
RSVP-TE Transport Solution with Flex LSP Headend	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
RSVP-TE Transport Solution with Flex LSP Headend	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) You can now configure up to four backup paths for both primary and secondary paths. This is an improvement over previous configurations that supported only two backup paths. This enhance the network's ability to handle failures gracefully and optimize performance You can now configure headend router or a centralized Path Computation Element (PCE) server to perform flex LSP path computation. For more information about PCE, see RSVP-TE transport solution using PCE. *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
RSVP-TE Transport Solution with Flex LSP Headend	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100]); Centralized Systems (8600 [ASIC:Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) You can enable the router to compute • a primary LSP path, • a secondary LSP path, • a dynamic restore path serves as a backup to the primary LSP path, and • another dynamic restore path acts as a backup to the secondary LSP path. During the primary LSP path failure, the headend router computes a new dynamic restore primary path by reusing the links from the failed primary LSP path, and during the secondary LSP path or backup path or protecting path failure, the headend router computes a new dynamic restore path by reusing the links from the failed secondary LSP path. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM **CLI:** The **protected-by** keyword is enhanced to include two backup paths in the **path-option** command.

RSVP-TE (Resource Reservation Protocol-Traffic Engineering) is a protocol used to reserve resources across an MPLS (Multiprotocol Label Switching) network to ensure Quality of Service (QoS) for specific traffic flows.

Flex LSP (Flexible Label Switched Path) is an enhancement of RSVP-TE that allows for more flexible and efficient selection of co-routed bidirectional LSP path selection.

Co-routed bidirectional LSP paths refer to LSP paths where traffic flows from headend to tailend and vice versa over the same physical path.

You can enable the router to configure these co-routed bidirectional LSP paths:

- **Primary flex LSP path**: This is an explicit path computed by the router between a headend node and a tailend node. This path serves as the main route for traffic.

- **Secondary flex LSP path**: In addition to the primary path, the router computes a secondary flex LSP path. This path acts as a backup to the primary path.
- **Dynamic restore paths**:
  - **Primary backup**: The headend router computes a dynamic LSP path using Constrained Shortest Path First (CSPF). This path serves as a backup to the primary flex LSP path.
  - **Secondary backup**: Another dynamic restore path is configured as a backup for the secondary flex LSP path. It provides additional redundancy and ensures continuous traffic flow even if the secondary path becomes unavailable.

For more information, see RSVP-TE Transport Solution with Flex LSP.

Once the failure is resolved, the router restores the previously used primary or secondary LSP paths.

Before Release 24.4.1, you can enable the router to configure

- a secondary LSP path as a backup to the primary LSP path, and
- dynamic restore path as a backup to the secondary LSP path.

From Release 24.4.1 onwards, you can enable the router to configure

- a secondary LSP path and a dynamic restore primary path as backups to the primary LSP path, and
- a dynamic restore path as a backup to the secondary LSP path.

From Release 25.1.1 onwards, you can configure a centralized PCE server to perform flex LSP path computation for primary, secondary, and restore paths. For more information, see RSVP-TE transport solution using PCE, on page 274.

How flex LSP headend traffic flow works

Summary

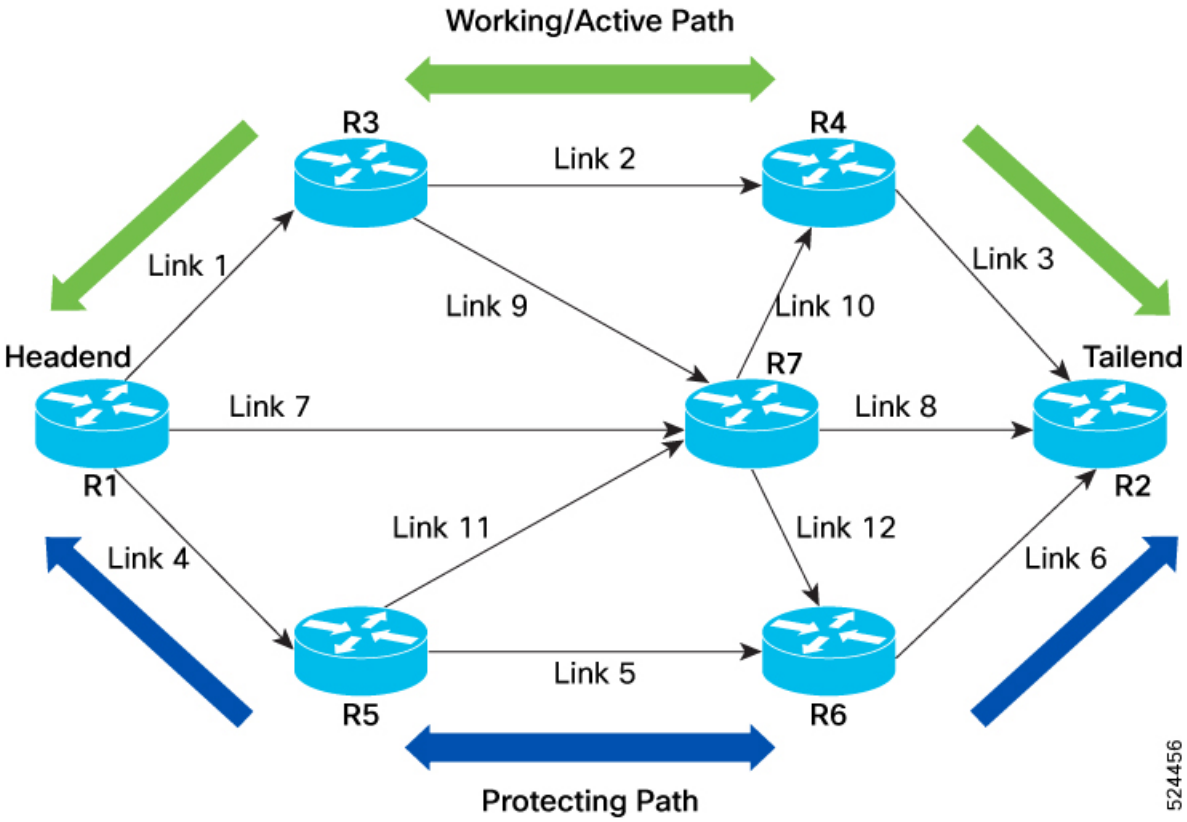
The key components that are involved in flex LSP headend traffic flow are:

- R1: Headend router
- R2: Tailend router
- R3, R4, R5, R6, and R7: Transit routers
- Path 10 (Link 1-2-3): Primary LSP path or Working or Active path
- Path 20 (Link 4-5-6): Secondary LSP path or Protecting path
- Path 30 (Link 1-9-10-3): Dynamic restore primary path
- Path 40 (Link 4-11-12-6): Dynamic restore path
- Path 50 (Link 7-8): Dynamic LSP path
- All links are 100G
- Weight:
  - Link 1 to Link 6: 10000 weights

- Link 7 to Link 12: 100 weights

Workflow

Figure 22: Flex LSP Headend Traffic Flow



These stages describe how flex LSP headend traffic flow works.

Before Release 24.4.1,

When	Then...
Link 2 of the primary LSP path is broken	the headend router activates the secondary LSP Path 20.
both primary and secondary LSP paths are broken	the headend router computes a new dynamic path using CSPF, without prioritizing the primary or secondary paths.
for example, Link 2 and Link 5 are broken	the headend router computes a new dynamic LSP Path 50, selecting the shortest path compared to Path 30 or Path 40.

From Release 24.4.1 onwards,

When	Then...
Link 2 of the primary LSP path is broken	the headend router activates the secondary LSP Path 20.
both primary and secondary LSP paths are broken	the headend router computes a new dynamic restore primary path, prioritizing the primary LSP path.
for example, Link 2 and Link 5 are broken	the headend router prioritizes Link 1 and assigns a new dynamic primary Path 30.
primary, secondary, and dynamic primary LSP paths are broken	the headend router computes a new dynamic restore path, prioritizing the secondary LSP path.
for example, Link 2, Link 5, and Link 10 are broken	the headend router prioritizes Link 4 and assigns a new dynamic restore Path 40.

Flex LSP headend scale details

By default, 1000 RSVP-TE tunnels are enabled on the edge routers. If you want to scale up the edge routers to support more than 1000 RSVP-TE tunnels

1. Scale up the edge routers to support maximum of 8000 RSVP-TE tunnels with **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command.
2. Reload all the LCs in the chassis with the **reload location all** command to enable the maximum of 8000 RSVP-TE tunnels.
3. Enable the RSVP-TE transport solution with flex LSP headend feature.

Below are the scale details for flex LSP headend.

Table 48: Supported LSPs for flex LSP headend

Supported LSPs for flex LSP headend	Scale Coverage
Flex LSP with path protection	Head/Tail Node: 4000 LSPs Mid Node: 8000 LSPs
Flex LSP without path protection	Head/Tail Node: 4000 LSPs Mid Node: 8000 LSPs

Limitations and restrictions for flex LSP headend

The flex LSP headend has these limitations and restrictions:

- This feature does not support the exp-null over Flex-LSP feature.
- This feature does not guarantee 50 msec convergence without WRAP protection. WRAP protection is mandatory to achieve 50 msec convergence for remote failures.
- TE NSR and IGP NSR are mandatory for RSP switchover.

- This feature does not support VPLS over Flex-LSP.
- This feature does not support non-co routed Flex LSP.
- Sub interface shut does not guarantee 50 msec convergence.
- This feature does not support MPLS forwarding table stats on mid nodes.
- This feature supports 4000 tunnels with and without wrap protection and path protection.
- This feature does not support the following features:
  - Bidirectional Forwarding Detection (BFD) over Flex-LSP
  - Multicast over Flex-LSP
  - Policy-Based Traffic Steering (PBTS)
  - Policy-Based Routing (PBR)
  - Access Control Lists (ACL)
  - L3VPN over Flex-LSP

Configure the co-routed flex LSPs headend

Perform these steps to configure co-routed flex LSPs headend:

Before you begin

- You must have symmetric source and destination TE router IDs for bidirectional LSPs to be associated.
- Tunnels attributes must be configured identically on both sides of co-routed bidirectional LSP.
- Before enabling the Flex LSP feature, scale up the RSVP-TE tunnels on the edge routers.

Procedure

Step 1 Scale up the edge routers with the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command to support a maximum of 8000 RSVP-TE tunnels.

Example:

```
Router# configure
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
Router(config)# commit
```

Step 2 You must reload all the LCs in the chassis with the **reload location all** command to enable the 8000 RSVP-TE tunnels.

Example:

```
Router# reload location all
```

Step 3 Configure MPLS-TE on Provider Edge (PE) router with **mpls traffic-eng** command to enable the MPLS traffic engineering on PE.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# interface HundredGigE0/9/0/0
```

Step 4 Configure RSVP on PE with **rsvp** command to enable the RSVP on PE.

```
Router# configure
Router(config)# rsvp
Router(config-rsvp)# interface HundredGigE0/0/0/37 bandwidth percentage 90
Router(config-rsvp)# interface HundredGigE0/1/0/40 bandwidth percentage 90
```

Step 5 Configure flex LSP with **association type co-routed** command to enable bidirectional co-routed LSPs.

```
Router# configure
Router(config)# interface tunnel-te7001
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 192.0.2.1
Router(config-if)# bidirectional
Router(config-if-bidir)# association id 1 source-address 192.0.2.0
Router(config-if-bidir-co-routed)# association type co-routed
```

Step 6 Enable wrap protection with **wrap-protection** command to enable LSP signals unique wrap label for headend.

```
Router(config-if-bidir-co-routed)# wrap-protection
```

Step 7 Configure path protection for a mpls-te tunnel with the **path-protection** command to create explicit and dynamic path.

```
Router(config-if-bidir-co-routed)# path-protection
```

Step 8 Configure working co-routed LSP path, protecting co-routed LSP path, working dynamic path, and protecting dynamic path using **path-option** command to enable the primary, secondary, and dynamic backup LSP paths.

From Release 25.1.1, You can configure up to four protected-by paths for the working and protecting paths, which serve as backup paths for the primary and secondary paths. In this configuration, only two protected-by paths are considered.

```
Router# configure
Router(config)# interface tunnel-te7001
Router(config-if)# path-option 10 explicit name r1-r2-working protected-by 20 40
Router(config-if)# path-option 20 explicit name r1-r2-protecting protected-by 10 30
Router(config-if)# path-option 30 dynamic attribute-set r1-r2-restoreworking protected-by 40
Router(config-if)# path-option 40 dynamic attribute-set r1-r2-restoreprotecting protected-by 30
```

Step 9 Map Pseudowire to a specific Flex LSP tunnel with **pw-class** command to ensure the Pseudowire traffic is transported over the MPLS TE tunnel.

```
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# pw-class pw1
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc-mpls)# preferred-path interface tunnel-te1
Router(config-l2vpn-pwc)# exit
Router(config-l2vpn)# exit
Router(config-l2vpn)# xconnect group gold
Router(config-l2vpn-xc)# p2p cust_one
Router(config-l2vpn-xc-p2p)# neighbor ipv4 192.2.2.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# pw-class pw1
```

Step 10 Configure explicit working dynamic path using **attribute-set** command to specify criteria for the working dynamic path computation.

```
Router# configure
Router(config)# mpls traffic
```

```
Router(config-if-mpls-te)# biased-to explicit-path r1-r2-working
Router(config-te-attribute-set)# commit
```

Step 11 Configure explicit protecting dynamic path using **attribute-set** command to specify criteria for the protecting dynamic path computation.

```
Router# configure
Router(config)# mpls traffic
Router(config-if-mpls-te)# attribute-set path-option r1-r2-restoreprotecting
Router(config-te-attribute-set)# commit
```

Step 12 Verify the configured co-routed flex LSP with **show mpls traffic-eng tunnels detail** command.

Example:

The following example dispays the co-routed flex LSP configuration with interface tunnel 7001 and destination address 104.0.0.1:

```
Router# show mpls traffic-eng tunnels 7001 detail

Name: tunnel-te7001  Destination: 104.0.0.1  Ifhandle:0x8000aa4
  Signalled-Name: Cisco8K-R11_t7001
  Status:
    Admin:    up Oper:   up (Uptime 136y10w)

    path option 1,  type explicit path01 (Basis for Setup, path weight 30 (reverse 30))
      Protected-by PO index: 2
    path option 2,  type explicit path02 (Basis for Standby, path weight 100010 (reverse 100010))
      Protected-by PO index: 1
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 10 kbps  CT0
    Creation Time: Wed Jan 11 03:08:36 2017 (136y10w ago)
  Config Parameters:
    Bandwidth:       10 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    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: 86, Source: 192.0.0.0
    Reverse Bandwidth: 10 kbps (CT0), Standby: 10 kbps (CT0)
    LSP Wrap Protection: Enabled

    Reoptimization after affinity failure: Enabled
    Soft Preemption: Disabled
  Fault-OAM Info:
    Last Fault Msg: Clear
  SNMP Index: 25
  Binding SID: None
  Path Protection Info:
    Standby Path: User defined [explicit path option: 2],
    Last Switchover:
      136y10w ago, From LSP 14 To LSP 16
```

```
    No subcause recorded
    Reopt time remaining: 0 seconds
  Number of Switchovers 1, Standby Ready 3 times, Standby Reopt 0 times
  Lockout Info:
    Locked Out: NO
    Locked out LSP ID: 0
    Lockout Originated By: None
  LSP Wrap Protection: Enabled
    LSP Wrap Label: 24182
History:
  Reopt. LSP:
    Last Failure:
      LSP not signalled, identical to the [CURRENT] LSP
      Date/Time: Tue Jan 10 21:42:41 UTC 2017 [00:03:42 ago]
  Standby Reopt LSP:
    Last Failure:
      LSP not signalled, identical to the [STANDBY] LSP
      Date/Time: Tue Jan 10 21:42:41 UTC 2017 [00:03:42 ago]
      First Destination Failed: 104.0.0.1
  Prior LSP:
    ID: 14 Path Option: 1
    Removal Trigger: path protection switchover
Current LSP Info:
  Instance: 18, Signaling Area: IS-IS 100 level-2
  Uptime: 136y10w (since Wed Jan 11 03:09:56 UTC 2017)
  Outgoing Interface: TenGigE0/4/0/2.1, Outgoing Label: 24157
  Router-IDs: local      102.0.0.1
              downstream 107.0.0.1
  Soft Preemption: None
  SRLGs: not collected
  Path Info:
    Outgoing:
      Explicit Route:
        Strict, 1.27.1.2
        Strict, 3.67.1.2
        Strict, 3.67.1.1
        Strict, 1.46.1.2
        Strict, 1.46.1.1
        Strict, 104.0.0.1

    Record Route: Disabled
    Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                        Soft Preemption Desired: Not Set
    Reverse Associated LSP Information:
      Signaled Name: Cisco8K-R10_t7001
      Tunnel: 7001, Source: 104.0.0.1, Dest: 102.0.0.1, LSP: 9, State: Up
    Association:
      Association Type: Single Sided Bidirectional LSPs
      Association ID: 86, Source: 192.0.0.0
    Extended Association:
      Global source: 0
      Extended ID:
        0x66000001 (102.0.0.1)
        0x12 (0.0.0.18)
    Protection:
      Secondary (S): 0, Protecting (P): 0, Notification (N): 0, Oper (O): 0
      Link Flags: Any, LSP Flags: 1:N Protection with Extra-Traffic
    Reverse Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Reverse ERO:
      Explicit Route:
        Strict, 1.46.1.1
        Strict, 1.46.1.2
        Strict, 3.67.1.1
```

```
            Strict, 3.67.1.2
            Strict, 1.27.1.2
            Strict, 1.27.1.1
            Strict, 102.0.0.1

      Resv Info: None
        Record Route: Disabled
        Fspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Standby LSP Info:
      Instance: 19, Signaling Area: IS-IS 100 level-2
      Uptime: 136y10w (since Wed Jan 11 03:10:04 UTC 2017), Oper State: Up
      Outgoing Interface: TenGigE0/4/0/11.1, Outgoing Label: 24176
      Router-IDs: local      102.0.0.1
                  downstream 109.0.0.1
      Soft Preemption: None
      SRLGs: not collected
      Path Info:
        Outgoing:
          Explicit Route:
            Strict, 1.29.1.2
            Strict, 1.49.1.2
            Strict, 1.49.1.1
            Strict, 104.0.0.1

        Record Route: Disabled
        Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
        Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                            Soft Preemption Desired: Not Set
        Reverse Associated LSP Information:
          Signaled Name: Cisco8K-R10_t7001
          Tunnel: 7001, Source: 104.0.0.1, Dest: 102.0.0.1, LSP: 10, State: Up
        Association:
          Association Type: Single Sided Bidirectional LSPs
          Association ID: 86, Source: 192.0.0.0
        Extended Association:
          Global source: 0
          Extended ID:
            0x68000001 (104.0.0.1)
            0xa (0.0.0.10)
        Protection:
          Secondary (S): 0, Protecting (P): 1, Notification (N): 0, Oper (O): 0
          Link Flags: Any, LSP Flags: 1:N Protection with Extra-Traffic
      Resv Info: None
        Record Route: Disabled
        Fspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
    Persistent Forwarding Statistics:
      Out Bytes: 20272384
      Out Packets: 79189

  LSP Tunnel 104.0.0.1 7001 [9] is signalled, Signaling State: up
    Tunnel Name: Cisco8K-R10_t7001 Tunnel Role: Tail
    InLabel: TenGigE0/4/0/2.1, 24164
    Signalling Info:
      Src 104.0.0.1 Dst 102.0.0.1, Tun ID 7001, Tun Inst 9, Ext ID 104.0.0.1
      Router-IDs: upstream   107.0.0.1
                  local      102.0.0.1
      Bandwidth: 10 kbps (CT0) Priority:  7  7 DSTE-class: 0
      Soft Preemption: None
      SRLGs: not collected
      Path Info:
        Incoming Address: 1.27.1.1
        Incoming:
        Explicit Route:
          Strict, 1.27.1.1
```

```
        Strict, 102.0.0.1

      Record Route: Disabled
      Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
      Reverse Associated LSP Information:
        Signaled Name: Cisco8K-R11_t7001
        Tunnel: 7001, Source: 102.0.0.1, Dest: 104.0.0.1, LSP: 18, State: Up
      Association:
        Association Type: Single Sided Bidirectional LSPs (Tie breaking slave)
        Association ID: 86, Source: 192.0.0.0
      Extended Association:
        Global source: 0
        Extended ID:
          0x66000001 (102.0.0.1)
          0x12 (0.0.0.18)
      Protection:
        Secondary (S): 0, Protecting (P): 0, Notification (N): 0, Oper (O): 0
        Link Flags: Any, LSP Flags: 1:N Protection with Extra-Traffic
    Resv Info: None
      Record Route: Disabled
      Fspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits

LSP Tunnel 104.0.0.1 7001 [10] is signalled, Signaling State: up
  Tunnel Name: Cisco8K-R10_t7001 Tunnel Role: Tail
  InLabel: TenGigE0/4/0/11.1, 24463
  Signalling Info:
    Src 104.0.0.1 Dst 102.0.0.1, Tun ID 7001, Tun Inst 10, Ext ID 104.0.0.1
    Router-IDs: upstream   109.0.0.1
                local      102.0.0.1
    Bandwidth: 10 kbps (CT0) Priority:  7  7 DSTE-class: 0
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Incoming Address: 1.29.1.1
      Incoming:
      Explicit Route:
        Strict, 1.29.1.1
        Strict, 102.0.0.1

      Record Route: Disabled
      Tspec: avg rate=10 kbits, burst=1000 bytes, peak rate=10 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                          Soft Preemption Desired: Not Set
      Reverse Associated LSP Information:
        Signaled Name: Cisco8K-R11_t7001
        Tunnel: 7001, Source: 102.0.0.1, Dest: 104.0.0.1, LSP: 19, State: Up
      Association:
        Association Type: Single Sided Bidirectional LSPs (Tie breaking slave)
        Association ID: 86, Source: 192.0.0.0
      Extended Association:   Fspec: avg rate=8K, burst=1K, peak rate=8K
```

Step 13 Verify the forwarding interface with the **show mpls forwarding tunnels detail** command.

Example:

```
Router# show mpls forwarding tunnels 7001 detail

Tunnel        Outgoing    Outgoing     Next Hop        Bytes
Name          Label       Interface                    Switched
------------- ----------- ------------ --------------- ------------
tt7001            24157       Te0/4/0/2.1  1.27.1.2        0
     Updated: Jan 10 21:40:04.966
```

```
   Version: 17852, Priority: 2
   Label Stack (Top -> Bottom): { 24157 }
   Local Label: 24354
   NHID: 0x0, Encap-ID: INVALID, Path idx: 0, Backup path idx: 0, Weight: 0
   MAC/Encaps: 18/22, MTU: 1500
   Packets Switched: 0

Interface Name: tunnel-te7001, Interface Handle: 0x08000aa4, Local Label: 24354
Forwarding Class: 0, Weight: 0
Packets/Bytes Switched: 79189/20272384
```

RSVP-TE transport solution using PCE

Table 49: Feature History Table

Feature Name	Release Information	Feature Description
RSVP-TE transport solution with flex LSP headend and PCE	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.

Feature Name	Release Information	Feature Description
RSVP-TE transport solution using PCE	Release 25.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100])(select variants only*); Centralized Systems (8600 [ASIC:Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100])(select variants only*) You can now enable a centralized Path Computation Element (PCE) server to perform LSP path computations for primary, secondary, and restore paths. The PCE efficiently tracks and manages network resources, handling multiple LSP requests and complex network environments. It optimizes LSP routing by dynamically adapting to network changes, reducing crank-back, and enhancing traffic engineering. *This feature is now supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

MPLS Traffic Engineering Path Computation Element (TE-PCE) extends the existing IOS XR Path Computation Element Protocol (PCEP) functionality to provide stateless PCE capabilities, supporting operations on both the MPLS data plane and the IPv4 control plane. For more information, see RFC 5440.

PCE characteristics:

- **Stateless operation**: The PCE operates in a stateless manner, meaning it does not maintain ongoing state information about the LSPs or paths it computes. It provides paths on demand based on real-time network conditions.
- **Request and reply Model**: The headend router sends a Path Computation Request (PCReq) to the PCE and receives a Path Computation Reply (PCRep) in return, containing the computed path details.
- **Topology learning**: The PCE learns about the network topology from the Interior Gateway Protocol (IGP).
- **Bandwidth management**: The PCE maintains information on in-progress bandwidth to reduce the need for computing the path again in the network.
- **Path disjointness**: The PCE can compute disjoint paths, ensuring that two or more LSPs do not share the network resources. This capability enhances network resilience by providing diverse path options that can originate from the same or different headend routers.

The Cisco IOS XRv 9000 is the recommended platform to act as the TE-PCE. For more information, see Cisco IOS XRv 9000 Router Installation and Configuration Guide.

Path definition

- **Explicit path**: A manually configured path that specifies a predetermined sequence of nodes or links for traffic passage from the headend to the tailend node.
- **Dynamic path**: A path automatically computed by either the PCE server or headend router using routing protocols and algorithms to adapt to real-time network conditions.

Before Release 25.1.1, the headend router were used to perform flex LSP path computation for these paths:

- **Primary path**: Depending on network requirements, you could configure an explicit path or enable the headend router to compute a dynamic path.
- **Secondary path**: Depending on network requirements, you could configure an explicit path or enable the headend router to compute a dynamic path.
- **Restore path for primary**: Headend router computes a new dynamic path by reusing the links from the failed primary path.
- **Restore path for secondary**: Headend router computes a new dynamic path by reusing the links from the failed secondary path.

For more information, see RSVP-TE Transport Solution with Flex LSP Headend, on page 263.

From Release 25.1.1, a centralized TE-PCE is introduced to perform the flex LSP paths computation for these paths:

- **Primary path**: Depending on network requirements, you can configure an explicit path or enable the PCE or headend router to compute a dynamic path.
- **Secondary path**: Depending on network requirements, you can configure an explicit path or enable the PCE or headend router to compute a dynamic path.
- **Restore path for primary**: The PCE server computes a new dynamic path by reusing links from the failed primary path.
- **Restore path for secondary**: The PCE server computes a new dynamic path by reusing links from the failed secondary path.

If the PCE connection is broken, the headend router performs flex LSP path computation.

Table 50: Difference between flex LSP path computation using headend and flex LSP path computation using PCE

Aspects	Flex LSP path computation using headend	Flex LSP path computation using PCE
Path computation	Performed by headend router.	Centralized computation by the PCE.
State maintenance	Stateful: Maintains state at mid-point nodes.	Stateless: Does not maintain an ongoing database of LSPs.

Aspects	Flex LSP path computation using headend	Flex LSP path computation using PCE
Bandwidth management	Prevents overbooking by tracking “in-progress” bandwidth based on reservations at mid-point nodes.	Prevents overbooking by computing bandwidth on real-time.
Topology updates	Synchronizes path and bandwidth information upon receiving new topology updates.	Computes based on the latest topology updates, without storing past states.
Network view	Limited to the information available at each node.	Utilizes a global view of the network, enabling optimized path computation.
Flexibility	Requires manual configuration for complex path requirements.	Offers flexibility through centralized computation, accommodating various constraints and requirements.
Redundancy handling	Need additional mechanisms for redundancy and failover management.	Incorporates redundancy and failover within its centralized computation model.

How PCEP session works

The headend or tailend routers which tries to connect to the PCE are referred to as Path Computation Client (PCC).

Connect the PCC to PCE server using the **mpls traffic pce server ipv4** command.

1. The PCC maintains the PCEP session until connection is active.
2. If the connection between PCC and PCE is broken, the PCC performs these steps to retry connecting to PCE:
   a. The PCC sends the PC request and waits to receive the PC reply from PCE.
   b. If the PCE does not respond to the PCC within this timeout, the PCC skips path-options that require the PCE and moves on to the next available path-option.

      By default, the PCC timeout for a PC request is 10 seconds. You can modify this timeout to any value between 5 to 100 seconds using the request-timeout command.
   c. If there is no more available path-options, then the PCC will retry the PCE path-option.

How PCE handles multiple PC request

The PCE performs these steps to compute paths concurrently when it receives multiple PC request:

When…	Then…
PCE receives the first LSP request with a specified disjoint-group ID	the PCE computes the first LSP, encoding the shortest path from the first source to the first destination.

When…	Then…
PCE receives the second LSP request with the same disjoint-group ID	the PCE utilizes the information from both requests to compute two disjoint paths concurrently.

Advantages of PCE

These are the advantages of RSVP-TE transport solution using PCE:

- You can connect multiple PCE servers to the headend and tailend router to enhance network redundancy.
- The PCC continues to verify PCE-computed paths even if the connection to the PCE is lost, ensuring that tunnels remain unaffected.
- The PCE computes co-routed bidirectional paths automatically when the reverse-bandwidth constraint is present, removing the need for special indication for co-routed and bidirectional computation.

Limitations and restrictions for RSVP-TE transport solution using PCE

The PCE has these limitations and restrictions:

- For the flex LSP headend limitations, see RSVP-TE Transport Solution with Flex LSP Headend, on page 263.
- In addition to the traditional path request parameters such as source and destination in the endpoints object as per RFC 5440 section 7.6, the path computation service supports the following:
  - Co-routed bidirectional dynamic computation
  - Forward and reverse asymmetrical bandwidth constraints
  - Affinity constraints
  - Priority for the bandwidth constraints
  - Diversity with an existing path
  - SRLG Diversity: on or off
  - Bias path to influence the computation to use certain links in the topology

How RSVP-TE transport solution with PCE works

Summary

You can configure primary and secondary paths as explicit or dynamic path based on the network requirements. For explaining this feature, we are considering these key components:

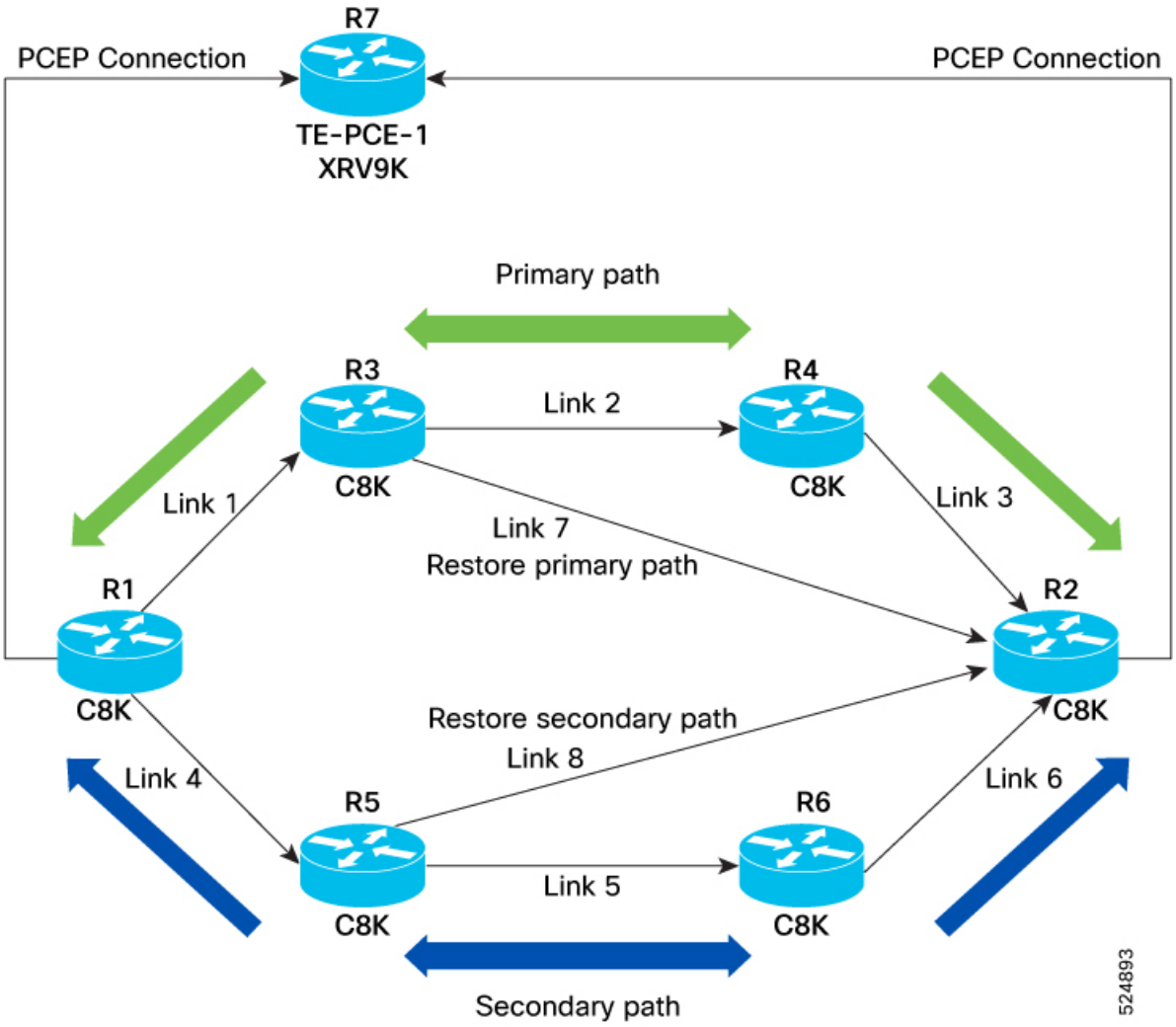
- **Headend router**: R1
- **Tailend router**: R2
- **Mid-point routers**: R3, R4, R5, and R6

- **PCE server**: R7 TE-PCE1, the Cisco IOS XRv 9000 platform serving as the centralized path computation element.
- This feature provides the flexibility to connect multiple PCEs to the headend and tailend router for network redundancy.
- To explain the workflow, we are prioritizing the PCE to perform flex LSP path computation, If the primary and secondary paths are broken, and If the PCE connection is broken then the headend router perform flex LSP path computation.
- **Paths**:
  - Path 1 (Link 1-2-3): Working or active or primary path, explicitly configured path.
  - Path 2 (Link 4-5-6): Protecting or standby or secondary path, explicitly configured path.
  - Path 3 (Link 1-7): Restore primary path, dynamically calculated through the PCE server or headend router.
  - Path 4 (Link 4-8): Restore secondary path, dynamically calculated through the PCE server or headend router.

  The headend router and PCE use the shortest path first (SPF) algorithm to compute flex LSP paths, allowing them to potentially compute identical paths.
- **Link Capacities**: All links are 100G
- **Weights**:
  - Link 1 to 6 have weights of 10,000
  - Link 7 and 8 have weights of 800

Workflow

Figure 23: Flex LSP path computation using PCE



These scenarios explain how PCE perfoms the flex LSP path computation:

When...	Then...
a primary path breaks	**Active path**: The secondary path becomes active, which is Path 2. **Standby path**: The PCE computes a restore primary path, which is Path 3, by reusing the links from the failed primary path.

When...	Then...
both primary and secondary path breaks	**Active path**: The restore primary path becomes active, which is Path 3. **Standby path**: The PCE computes a restore secondary path, which is Path 4, by reusing the links from the failed secondary path.
the broken link comes up again	the primary or secondary path becomes active • after reoptimization or • you can perform manual reoptimization using the **mpls traffic-eng reoptimize** command. **Reoptimization**: The broken primary or secondary path becomes active after the configured reoptimization time limt. By default, the reoptimization time limit is 3600 seconds, but you can configure it between 0 to 604800 seconds using the **reoptimize** command under MPLS traffic engineering.

In a scenario where the headend and tailend routers lose connection with the PCE, the headend router performs the flex LSP path computation. For more information, see How flex LSP headend traffic flow works, on page 265.

Configure the RSVP-TE transport solution with flex LSP headend and PCE

Perform these steps on headend and tailend routers to configure the RSVP-TE transport solution with flex LSP headend and PCE:

Before you begin

- **Symmetric Configuration**: Ensure symmetric source and destination TE router IDs for bidirectional LSPs to be associated.
- **Tunnel Attributes**: Configure tunnel attributes identically on both sides of co-routed bidirectional LSPs.
- **Path Configuration**: You can configure primary and secondary paths as explicit or dynamic path based on the network requirements. For explaining this configuration, we are considering these key components:
  - **Path 10**: Explicit primary path protected by Paths 20, 50, and 60.
  - **Path 20**: Explicit secondary path protected by Paths 10, 30, and 40.
  - **Path 30**: Restore primary path dynamically computed by the PCE server.
  - **Path 40**: Restore primary path dynamically computed by the headend router.
  - **Path 50**: Restore secondary path dynamically computed by the PCE server.
  - **Path 60**: Restore secondary path dynamically computed by the headend router.
  - **Path Restoration Priority**:

- If both primary and secondary paths are broken, the PCE is prioritized to compute restored primary and secondary paths.
- If the connection of the headend and tailend routers with the PCE is broken, the headend router computes restored primary and secondary paths.

- The headend and tailend routers, along with the PCE, use the Shortest Path First (SPF) algorithm to compute flex LSP paths, potentially resulting in identical paths.

- Configure these setups:
  - **Bundle ethernet interfaces**: Configure bundle ethernet interfaces for efficient link aggregation.
  - **Loopback interface**: Set up a loopback interface for reliable routing.
  - **L2VPN configuration**: Define Pseudowire (PW) classes and point-to-point (P2P) cross-connects within an xconnect group.
  - **RSVP configuration**: Configure RSVP on PE devices to reserve bandwidth.
  - **Administrative weight**: Assign administrative weights to interfaces for path preference.
  - **MPLS LDP configuration**: Configure MPLS Label Distribution Protocol (LDP) on network devices for label switching.

Procedure

Step 1 If your network requires more than 1,000 tunnels, perform these steps:

a) Scale up the routers with the **hw-module profile cef te-tunnel highscale-no-ldp-over-te** command to support a maximum of 8,000 RSVP-TE tunnels.

Example:

```
Router# configure
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
Router(config)# commit
```

b) Reload all the LCs in the chassis with the **reload location all** command to enable the 8000 RSVP-TE tunnels.

Example:

```
Router# reload location all
```

Step 2 Configure primary path as explicit or dynamic path based on the network requirements. In this configuration, the primary LSP path is explicit.

Example:

```
Router# configure
Router(config)# explicit-path name R1-R2-Working-1
Router(config-exp-path)# index 1 next-address strict ipv4 unicast 101.1.3.17
Router(config-exp-path)# index 2 next-address strict ipv4 unicast 101.1.3.18
Router(config-exp-path)# index 3 next-address strict ipv4 unicast 101.3.4.17
Router(config-exp-path)# index 4 next-address strict ipv4 unicast 101.3.4.18
Router(config-exp-path)# index 5 next-address strict ipv4 unicast 101.2.4.6
Router(config-exp-path)# index 6 next-address strict ipv4 unicast 101.2.4.5
Router(config-exp-path)# exit
```

Step 3 Configure secondary path as explicit or dynamic path based on the network requirements. In this configuration, the secondary LSP path is explicit.

Example:

```
Router# configure
Router(config)# explicit-path name R1-R2-Protecting-1
Router(config-exp-path)# index 1 next-address strict ipv4 unicast 101.1.5.97
Router(config-exp-path)# index 2 next-address strict ipv4 unicast 101.1.5.98
Router(config-exp-path)# index 3 next-address strict ipv4 unicast 101.5.6.1
Router(config-exp-path)# index 4 next-address strict ipv4 unicast 101.5.6.2
Router(config-exp-path)# index 5 next-address strict ipv4 unicast 101.2.6.6
Router(config-exp-path)# index 6 next-address strict ipv4 unicast 101.2.6.5
Router(config-exp-path)# exit
```

Step 4 Configure a MPLS TE tunnel with bidirectional co-routed LSP, and enable wrap protection.

Example:

```
Router# configure
Router(config)# interface tunnel-te51
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# ipv6 enable
Router(config-if)# autoroute announce include-ipv6
Router(config-if)# destination 100.2.1.1
Router(config-if)# bidirectional
Router(config-if-bidir)# association id 51 source-address 100.1.1.1
Router(config-if-bidir)# association type co-routed
Router(config-if-bidir-co-routed)# wrap-protection
Router(config-if-bidir-co-routed)# exit
Router(config-if-bidir)# exit
```

Step 5 Configure path protection, path option for primary path, secondary path, and dynamic restore paths for primary and secondary.

You can configure up to four protected-by paths for the working and protecting paths, which serve as backup paths for the primary and secondary paths. In this configuration, only three protected-by paths are considered.

Example:

```
Router(config-if)# path-protection
Router(config-if)# path-option 10 explicit name R1-R2-Working-1 protected-by 20 50 60
Router(config-if)# path-option 20 explicit name R1-R2-Protecting-1 protected-by 10 30 40
Router(config-if)# path-option 30 dynamic attribute-set R1-R2-RestoreWorking-pce-1 protected-by 50
Router(config-if)# path-option 40 dynamic attribute-set R1-R2-RestoreWorking-distributed-1
protected-by 60
Router(config-if)# path-option 50 dynamic attribute-set R1-R2-RestoreProtecting-pce-1 protected-by
 30
Router(config-if)# path-option 60 dynamic attribute-set R1-R2-RestoreProtecting-distributed-1
protected-by 40
```

Step 6 Perform the these steps to configure PCEP connection:

a) Configure PCE server to setup a centralized TE-PCE.

The Cisco IOS XRv 9000 is the recommended platform to act as the TE-PCE. For more information, see Cisco IOS XRv 9000 Router Installation and Configuration Guide.

Example:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# pce
Router(config-mpls-te-pce)# server ipv4 100.7.1.1
Router(config-mpls-te-pce)# commit
```

b) Configure PCE peer on headend and tailend routers to enable the PCEP connection.

By default, the PCC timeout for a PC request is 10 seconds. You can modify this timeout to any value between 5 to 100 seconds using the **request-timeout** command.

Example:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# pce
Router(config-mpls-te-pce)# peer ipv4 100.7.1.1
Router(config-mpls-te-pce)# request-timeout 100
Router(config-mpls-te-pce)# exit
Router(config-mpls-te)# commit
```

Step 7 Configure attribute set for explicit primary and secondary LSP path.

Example:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# attribute-set path-option R1-R2-RestoreWorking-distributed-1
Router(config-te-attribute-set)# biased-to explicit-path R1-R2-Working-1
Router(config-te-attribute-set)# commit
Router(config-mpls-te)# attribute-set path-option R1-R2-RestoreProtecting-distributed-1
Router(config-te-attribute-set)# biased-to explicit-path R1-R2-Protecting-1
Router(config-te-attribute-set)# commit
```

Step 8 Configure attribute set for primary and secondary LSP path with PCE.

Example:

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# attribute-set path-option R1-R2-RestoreWorking-pce-1
Router(config-te-attribute-set)# pce
Router(config-te-attribute-set)# biased-to explicit-path R1-R2-Working-1
Router(config-te-attribute-set)# commit
Router(config-mpls-te)# attribute-set path-option R1-R2-RestoreProtecting-pce-1
Router(config-te-attribute-set)# pce
Router(config-te-attribute-set)# biased-to explicit-path R1-R2-Protecting-1
Router(config-te-attribute-set)# commit
```

Step 9 Run the **sh mpls traffic-eng pce peer** command on the headend router to verify the PCEP connection with headend.

Example:

```
Router# sh mpls traffic-eng pce peer

Address Precedence State Learned From
--------------- ------------ ------------ --------------------
100.7.1.1 255 Up Static config
```

Step 10 Run the **show mpls traffic-eng tunnels <tunnel id> detail** command on the headend router, to verify MPLS TE tunnel with explicit paths and dynamic restore paths for primary and secondary routes computed by the PCE server or headend routers.

These examples explain scenarios of how the PCE server compute dynamic restore paths for primary and secondary routes during a failure:

a) In this example, Path 10 is the active path, and Path 20 is the standby path.

Example:

```
Router# show mpls traffic-eng  tunnels  51 detail
Name: tunnel-te51  Destination: 100.2.1.1  Ifhandle:0x7800a684
  Signalled-Name: R1_t51
  Status:
    Admin:    up Oper:   up (Uptime 00:05:21)
    path option 10,  type explicit R1-R2-Working-1 (Basis for Setup, path weight 30000 (reverse
 30000))
      Accumulative metrics: TE 30000 (reverse 30000) IGP 30 (reverse 30) Delay 900000 (reverse
 900000)
      Accumulative biased metrics: TE 30000 (reverse 30000) IGP 30 (reverse 30) Delay 900000
(reverse 900000)
      Protected-by PO index: 20,50,60
    path option 20,  type explicit R1-R2-Protecting-1 (Basis for Standby, path weight 30000
(reverse 30000))
      Accumulative metrics: TE 30000 (reverse 30000) IGP 30 (reverse 30) Delay 900000 (reverse
 900000)
      Accumulative biased metrics: TE 30000 (reverse 30000) IGP 30 (reverse 30) Delay 900000
(reverse 900000)
      Protected-by PO index: 10,30,40
    path option 30,  type dynamic pce
      Protected-by PO index: 50
      Path-option attribute: R1-R2-RestoreWorking-pce-1
        PCE enabled
        Biased-to: explicit-path R1-R2-Working-1
    path option 40,  type dynamic
      Protected-by PO index: 60
      Path-option attribute: R1-R2-RestoreWorking-distributed-1
        Biased-to: explicit-path R1-R2-Working-1
    path option 50,  type dynamic pce
      Protected-by PO index: 30
      Path-option attribute: R1-R2-RestoreProtecting-pce-1
        PCE enabled
        Biased-to: explicit-path R1-R2-Protecting-1
    path option 60,  type dynamic
      Protected-by PO index: 40
      Path-option attribute: R1-R2-RestoreProtecting-distributed-1
        Biased-to: explicit-path R1-R2-Protecting-1
*
*
  Current LSP Info:
    Role: Working
    Instance: 11, Signaling Area: PCE computed path: server 100.7.1.1
    In-use path-option: 10
    Uptime: 00:05:21 (since Wed Mar 19 12:03:16 PDT 2025)
    Outgoing Interface: HundredGigE0/8/0/10, Outgoing Label: 24050
    Router-IDs: local      100.1.1.1
                downstream 100.3.1.1
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 101.1.3.18
          Strict, 101.3.4.18
          Strict, 101.2.4.5
          Strict, 100.2.1.1

*
*

  Standby LSP Info:
```

```
    Role: Protecting
    Instance: 13, Signaling Area: PCE computed path: server 100.7.1.1
    In-use path-option: 20
    Uptime: 00:04:52 (since Wed Mar 19 12:03:45 PDT 2025), Oper State: Up
    Outgoing Interface: HundredGigE0/8/0/0, Outgoing Label: 24051
    Router-IDs: local      100.1.1.1
                downstream 100.5.1.1
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 101.1.5.98
          Strict, 101.5.6.2
          Strict, 101.2.6.5
          Strict, 100.2.1.1
```

The above sample displays only a part of the actual output; the actual output displays more details.

b) In this example:

- The explicit primary path, Path 10, is broken.
- Path 20 becomes the active path, protected by Paths 10, 30, and 40.
- The PCE computes a restored primary path, which is Path 30, as a standby path by reusing the links from the original primary path.

Example:

```
Router# show mpls traffic-eng  tunnels  51 detail
Name: tunnel-te51  Destination: 100.2.1.1  Ifhandle:0x7800a684
  Signalled-Name: R1_t51
  Status:
    Admin:    up Oper:   up (Uptime 00:09:56)

   path option 20,  type explicit R1-R2-Protecting-1 (Basis for Setup, path weight 30000 (reverse
 30000))
      Accumulative metrics: TE 30000 (reverse 30000) IGP 30 (reverse 30) Delay 900000 (reverse
 900000)
      Accumulative biased metrics: TE 30000 (reverse 30000) IGP 30 (reverse 30) Delay 900000
(reverse 900000)
      Protected-by PO index: 10,30,40
    path option 30,  type dynamic pce (Basis for Standby, path weight 100 (reverse 100))
      Accumulative metrics: TE 10100 (reverse 10100) IGP 20 (reverse 20) Delay 600000 (reverse
 600000)
     Accumulative biased metrics: TE 100 (reverse 100) IGP 10 (reverse 10) Delay 300000 (reverse
 300000)
      Protected-by PO index: 50
      Path-option attribute: R1-R2-RestoreWorking-pce-1
        PCE enabled
        Biased-to: explicit-path R1-R2-Working-1
*
*
  Current LSP Info:
    Role: Working
    Instance: 16, Signaling Area: PCE computed path: server 100.7.1.1
    In-use path-option: 20
    Uptime: 00:00:41 (since Wed Mar 19 12:12:31 PDT 2025)
    Outgoing Interface: HundredGigE0/8/0/0, Outgoing Label: 24065
    Router-IDs: local      100.1.1.1
                downstream 100.5.1.1
```

```
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 101.1.5.98
          Strict, 101.5.6.2
          Strict, 101.2.6.5
          Strict, 100.2.1.1
*
*
Standby LSP Info:
    Role: Protecting
    Instance: 18, Signaling Area: PCE computed path: server 100.7.1.1
    In-use path-option: 30
    Uptime: 00:00:27 (since Wed Mar 19 12:12:45 PDT 2025), Oper State: Up
    Outgoing Interface: HundredGigE0/8/0/10, Outgoing Label: 24061
    Router-IDs: local      100.1.1.1
                downstream 100.3.1.1
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 101.1.3.18
          Strict, 101.2.3.17
          Strict, 100.2.1.1
```

The above sample displays only a part of the actual output; the actual output displays more details.

c) In this example:

- The explicit secondary path, Path 20, is broken.
- Path 30 becomes the active path, protected by Path 50.
- The PCE computes a restored secondary path, which is Path 50, as a standby path by reusing the links from the original secondary path.

Example:

```
Router# show mpls traffic-eng  tunnels  51 detail
Name: tunnel-te51  Destination: 100.2.1.1  Ifhandle:0x7800a684
  Signalled-Name: R1_t51
  Status:
    Admin:    up Oper:   up (Uptime 00:10:47)

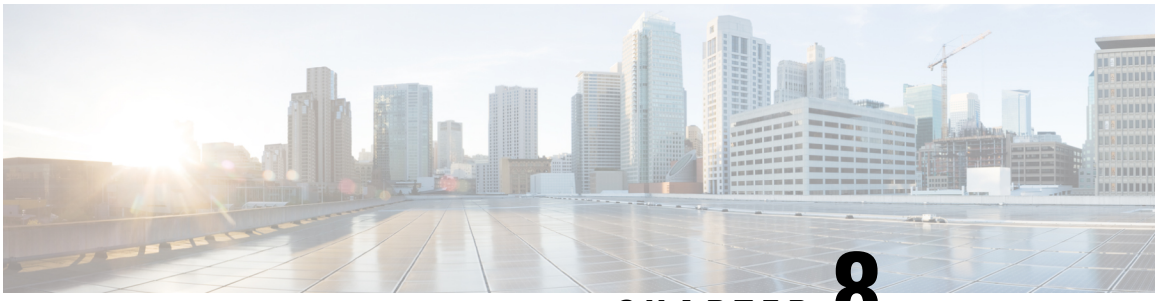
    path option 30,  type dynamic pce (Basis for Setup, path weight 100 (reverse 100))
      Accumulative metrics: TE 10100 (reverse 10100) IGP 20 (reverse 20) Delay 600000 (reverse
 600000)
     Accumulative biased metrics: TE 100 (reverse 100) IGP 10 (reverse 10) Delay 300000 (reverse
 300000)
      Protected-by PO index: 50
      Path-option attribute: R1-R2-RestoreWorking-pce-1
        PCE enabled
        Biased-to: explicit-path R1-R2-Working-1
    path option 50,  type dynamic pce (Basis for Standby, path weight 100 (reverse 100))
      Accumulative metrics: TE 10100 (reverse 10100) IGP 20 (reverse 20) Delay 600000 (reverse
 600000)
     Accumulative biased metrics: TE 100 (reverse 100) IGP 10 (reverse 10) Delay 300000 (reverse
 300000)
      Protected-by PO index: 30
```

```
      Path-option attribute: R1-R2-RestoreProtecting-pce-1
        PCE enabled
        Biased-to: explicit-path R1-R2-Protecting-1
*
*
 Current LSP Info:
    Role: Working
    Instance: 13, Signaling Area: PCE computed path: server 100.7.1.1
    In-use path-option: 30
    Uptime: 00:00:42 (since Wed Mar 19 12:13:24 PDT 2025)
    Outgoing Interface: Bundle-Ether2301, Outgoing Label: 24044
    Router-IDs: local      100.2.1.1
                downstream 100.3.1.1
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 101.2.3.18
          Strict, 101.1.3.17
          Strict, 100.1.1.1

*
*

  Standby LSP Info:
    Role: Protecting
    Instance: 14, Signaling Area: PCE computed path: server 100.7.1.1
    In-use path-option: 50
    Uptime: 00:00:21 (since Wed Mar 19 12:13:45 PDT 2025), Oper State: Up
    Outgoing Interface: Bundle-Ether2501, Outgoing Label: 24045
    Router-IDs: local      100.2.1.1
                downstream 100.5.1.1
    Soft Preemption: None
    SRLGs: not collected
    Path Info:
      Outgoing:
        Explicit Route:
          Strict, 101.2.5.18
          Strict, 101.1.5.97
          Strict, 100.1.1.1
```

The above sample displays only a part of the actual output; the actual output displays more details.



CHAPTER 8

Implementing RSVP for MPLS-TE

Table 51: Feature History Table

Feature Name	Release Information	Feature Description
Implementing RSVP for MPLS-TE	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Implementing RSVP for MPLS-TE	Release 25.4.1	Introduced in this release on: Fixed Systems 8010 [ASIC: A100], (8700 [ASIC: K100])(select variants only*) * This feature is now supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Implementing RSVP for MPLS-TE	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Implementing RSVP for MPLS-TE	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) Enhance network traffic management with RSVP for MPLS Traffic Engineering, enabling precise control over data paths. This feature supports the establishment of Label Switched Paths (LSPs) based on specific bandwidth and latency requirements, optimizing resource utilization. It provides dynamic path selection and rerouting capabilities, ensuring efficient network operations and improved service quality by adapting to changing network conditions and demands. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

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.

MPLS Traffic Engineering (MPLS-TE) learns the topology and resources available in a network and then maps traffic flows to particular paths based on resource requirements and network resources such as bandwidth. MPLS TE builds a unidirectional tunnel from a source to a destination in the form of a label switched path (LSP), which is then used to forward traffic. MPLS-TE uses RSVP to signal LSPs.

• Usage Guidelines and Limitations, on page 290
• Setting up MPLS LSP Using RSVP, on page 291
• Overview of RSVP for MPLS-TE Features, on page 291
• Bandwidth Reservation Percentage, on page 292
• Caveats for Out-of-Sequence , on page 292
• Keychain Configuration For RSVP Authentication, on page 292
• Configuring RSVP for MPLS-TE, on page 292
• RSVP for MPLS-TE Features - Details, on page 311
• Additional References , on page 314

Usage Guidelines and Limitations

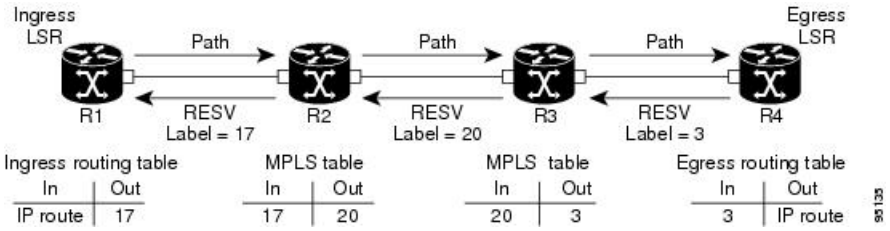
Consider these usage guidelines and limitations before implementing RSVP for MPLS-TE:

- To switch between an RSVP-TE or SR-TE configuration and an MPLS-TE configuration and vice versa, it is necessary to remove the configuration of the first process, say, RSVP-TE or SR-TE, in a separate commit operation and add the configuration of the next process, say, MPLS-TE, in another commit operation. This prevents any commit failures caused due to the delay in unconfiguring the first process before the next process configuration is applied.

Setting up MPLS LSP Using RSVP

The following figure shows how RSVP sets up an LSP from router R1 through router R4 that can be used for TE in an MPLS environment.

Figure 24: MPLS LSP Using RSVP



The LSP setup is initiated when the LSP head node sends path messages to the tail node. The Path messages reserve resources along the path to each node, and creates path states associated with the session 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. The reservation state in each router is considered as a soft state, which means that periodic PATH and RESV messages must be sent at each hop to maintain the state.

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.

Overview of RSVP for MPLS-TE Features

This section provides an overview of the various features of RSVP for MPLS-TE.

RSVP is automatically enabled on interfaces on which MPLS-TE is configured. For MPLS-TE LSPs with 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 Graceful restart ensures high availability and allows RSVP TE enabled routers to recover RSVP state information from neighbors after a failure in the network.

RSVP requires that the path and reservation state that are set up during LSP signaling must be refreshed by periodically sending refresh messages. Refresh messages are used to synchronize the state between RSVP neighbors and to recover from lost RSVP messages. RSVP refresh reduction feature includes support for reliable messages which are transmitted rapidly when the messages are lost. Summary refresh messages contain information to refresh multiple states and reduces the number of messages required to refresh states.

RSVP messages can be authenticated to ensure that only trusted neighbors can set up reservations.

For detailed information about RSVP for MPLS-TE features, see the *RSVP for MPLS-TE Features- Details* topic.

Bandwidth Reservation Percentage

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

For more information on configuring RSVP bandwidth, refer the *Implementing MPLS Traffic Engineering* chapter in the *MPLS Configuration Guide for Cisco 8000 Series Routers*. For more information on commands for configuring RSVP bandwidth, refer the *RSVP Infrastructure Commands* chapter in the *MPLS Command Reference for Cisco 8000 Series Routers*.

Caveats for Out-of-Sequence

These caveats are listed for out-of-sequence:

- When RSVP messages traverse multiple interface types with different maximum transmission unit (MTU) values, some messages can become out-of-sequence if they are fragmented.
- Packets with some IP options may be reordered.
- Change in QoS configurations may lead to a transient reorder of packets.
- QoS policies can cause a reorder of packets in a steady state.

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

Keychain Configuration For 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 *System Security Configuration Guide for Cisco 8000 Series Routers*.



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

Configuring RSVP for MPLS-TE

RSVP requires coordination among several routers, establishing exchange of RSVP messages to set up LSPs. Depending on the requirements, RSVP requires some basic configuration described in the following 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.

Confirming 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.

Configuration Example

In this example, the **bandwidth** command sets the total reservable bandwidth, the maximum RSVP bandwidth available for a flow and the sub-pool bandwidth for the HundredGigE 0/0/0/3 interface.

```
Router# configure
Router(config)# rsvp interface HundredGigE0/0/0/3
Router(config-rsvp-if)# bandwidth 1000 mbps 100 mbps sub-pool 150 mbps
Router(config-rsvp-if)# commit
```

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.

Configure RSVP message authentication globally

Table 52: Feature History Table

Feature Name	Release Information	Feature Description
Configure RSVP message authentication globally	Release 26.1.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8404-SYS-D routers.
Configure RSVP message authentication globally	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) * This feature is now supported on: • 8711-48Z-M • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Configure RSVP message authentication globally	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Configure RSVP message authentication globally	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) Improve network efficiency with RSVP MPLS-TE Automatic Bandwidth Adjustment, which dynamically adjusts the bandwidth of Label Switched Paths (LSPs) based on actual traffic patterns. This feature ensures optimal resource allocation by automatically recalculating and modifying bandwidth reservations, leading to better utilization of network capacity. It enhances service quality by adapting to varying traffic demands without manual intervention, providing a more responsive and flexible network environment. *Previously this feature was supported on Q200 and Q100. This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH

The RSVP authentication feature permits neighbors in an RSVP network to use a secure hash algorithm to authenticate all RSVP signaling messages digitally. The authentication is accomplished on a per-RSVP-hop basis using an RSVP integrity object in the RSVP message. The integrity object includes a key ID, a sequence number for messages, and keyed message digest.

You can globally configure the values of authentication parameters including the key-chain, time interval that RSVP maintains security associations with other trusted RSVP neighbors (life time) and maximum number of RSVP authenticated messages that can be received out of sequence (window size). These defaults are inherited for each neighbor or interface.

Configuration Example

In this example, authentication parameters are configured globally on a router. The authentication parameters including authentication key-chain, lifetime, and window size are configured. A valid key-chain should be configured before performing this task.

```
Router# configure
Router(config)# key chain mpls-keys
Router(config-mpls-keys)# commit
Router(config-mpls-keys)# exit
Router(config)# rsvp authentication
Router(config-rsvp-auth)# key-source key-chain mpls-keys
Router(config-rsvp-auth)# life-time 2000
Router(config-rsvp-auth)# window-size 33
```

Verification

Verify the configuration of authentication parameters using the following command.

```
Router# show rsvp authentication detail

  RSVP Authentication Information:
    Source Address:         3.0.0.1
    Destination Address:    3.0.0.2
    Neighbour Address:      3.0.0.2
    Interface:              HundredGigabitEthernet 0/0/0/3
    Direction:              Send
    LifeTime:               2000 (sec)
    LifeTime left:          1305 (sec)
    KeyType:                Static Global KeyChain
    Key Source:             mpls-keys
    Key Status:             No error
    KeyID:                  1
    Digest:                 HMAC MD5 (16)
    window-size:            33
Challenge:              Not supported
    TX Sequence:            5023969459702858020 (0x45b8b99b00000124)
    Messages successfully authenticated:    245
    Messages failed authentication:         0
```

Configuring RSVP Authentication for an Interface

You can individually configure the values of RSVP authentication parameters including key-chain, life time, and window size on an interface. Interface specific authentication parameters are used to secure specific interfaces between two RSVP neighbors.

Configuration Example

This example configures authentication key-chain, life time for the security association, and window size on an interface. A valid key-chain should be already configured to use it as part of this task.

```
Router# configure
Router(config)# rsvp interface HundredGigE0/0/0/3
Router(config-rsvp-if)# authentication
Router(config-rsvp-if-auth)# key-source key-chain mpls-keys
Router(config-rsvp-if-auth)# life-time 2000
Router(config-rsvp-if-auth)# window-size 33
Router(config-rsvp-if-auth)# commit
```

Cisco IOS XR Release 7.11.1 introduces support to disable RSVP authentication.

Verification

Verify the configuration of authentication parameters using the following command.

```
Router# show rsvp authentication detail

  RSVP Authentication Information:
    Source Address:         3.0.0.1
    Destination Address:    3.0.0.2
    Neighbour Address:      3.0.0.2
    Interface:              HundredGigabitEthernet 0/0/0/3
    Direction:              Send
    LifeTime:               2000 (sec)
    LifeTime left:          1305 (sec)
    KeyType:                Static Global KeyChain
```

```
  Key Source:             mpls-keys
  Key Status:             No error
  KeyID:                  1
  Digest:                 HMAC MD5 (16)
  window-size:            33
  Challenge:              Not supported
  TX Sequence:            5023969459702858020 (0x45b8b99b00000124)
  Messages successfully authenticated:    245
  Messages failed authentication:         0
```

Configuring RSVP Authentication on a Neighbor

You can individually configure the values of RSVP authentication parameters including key-chain, life time, and window size on a neighbor.

Configuration Example

This example configures the authentication key-chain, life time for the security association, and window size on a RSVP neighbor. A valid key-chain should be already configured to use it as part of this task.

```
Router# configure
Router(config)# rsvp neighbor 10.0.0.1 authentication
Router(config-rsvp-nbor-auth)# key-source key-chain mpls-keys
Router(config-rsvp-nbor-auth)# life-time 2000
Router(config-rsvp-nbor-auth)# window-size 33
Router(config-rsvp-nbor-auth)# commit
```

Verification

Verify the configuration of authentication parameters using the following command.

```
Router# show rsvp authentication detail

  RSVP Authentication Information:
    Neighbour Address:      10.0.0.1
    Interface:              HundredGigabitEthernet 0/0/0/3
    Direction:              Send
    LifeTime:               2000 (sec)
    LifeTime left:          1205 (sec)
    KeyType:                Static Global KeyChain
    Key Source:             mpls-keys
    Key Status:             No error
    KeyID:                  1
    Digest:                 HMAC MD5 (16)
    window-size:            33
    Challenge:              Not supported
```

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.

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

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

MPLS-TE PCE authentication

The PCEP session authentication is a security mechanism that

- establishes a secure and trusted communication channel between a PCC and a PCE,
- ensures that only authorized devices can participate in PCEP sessions, and
- protects against unauthorized access and data tampering.

Methods of MPLS-TE PCE authentication

Before Release 26.1.1, you can perform MPLS-TE PCE authentication using TCP Message Digest 5 (MD5) authentication. This method uses a clear text or encrypted password for authentication. The Cisco IOS XR software rejects segments that lack a Message Authentication Code (MAC) matching the configured password.

From Release 26.1.1, you can perform MPLS-TE PCE authentication using TCP Authentication Option (TCP-AO).

TCP-AO uses Message Authentication Codes (MACs), which provides the following benefits:

- It protects against replay attacks for long-lived TCP connections.
- It provides more detailed security association with TCP connections than TCP MD5.
- It supports a larger set of MAC algorithms with minimal system and operational changes.

TCP-AO is compatible with Master Key Tuple (MKT) configuration, protecting connections by deriving traffic keys from the MKT and coordinating changes between endpoints. The Cisco IOS XR software rejects segments that lack a MAC matching the configured key chain.



Note TCP-AO and TCP MD5 are never permitted to be used simultaneously. TCP-AO supports IPv6, and is fully compatible with the proposed requirements for the replacement of TCP MD5.

Configure MPLS-TE PCE authentication

Procedure

Step 1 Before Release 26.1.1, you can configure MPLS-TE PCE authentication using TCP MD5 authentication. Specify if the password is encrypted or clear text.

Example:

```
Router# configure terminal
Router(config)# mpls traffic-eng
Router(config-mpls-te)# pce password [{clear | encrypted} <password>]
```

Step 2 From Release 26.1.1, you can configure MPLS-TE PCE authentication using TCP Authentication Option (TCP-AO). Use the **include-tcp-options** keyword to include other TCP options in the header for MAC calculation.

Example:

```
Router# configure terminal
Router(config)# mpls traffic-eng
Router(config-mpls-te)# pce tcp-ao <password> include-tcp-options
```

Step 3 Run the **show mpls traffic-eng pce peer** command to verify the PCE up status.

Example:

```
Router# show mpls traffic-eng pce peer

Address Precedence State Learned From
--------------- ------------ ------------ --------------------
192.168.1.4 255 Up Static config
```

Step 4 Run the **show tcp detail** command to verify the TCP session details.

Example:

```
Router# show tcp detail pcb all

================================================================
Connection state is ESTAB, I/O status: 0, socket status: 0
Established at Thu Feb 26 05:36:17 2026

PCB 0x00007fc85c0049d0, SO 0x7fc85c0045b0, TCPCB 0x7fc85c004ce0, vrfid 0x60000000,
Pak Prio: Medium, TOS: 192, TTL: 255, Hash index: 1854
Local host: 192.168.0.1, Local port: 52364 (Local App PID: 8630)
Foreign host: 192.168.1.4, Foreign port: 4189
(Local App PID/instance/SPL_APP_ID: 8630/0/0)

Current send queue size in bytes: 0 (max 512000)
Current receive queue size in bytes: 0 (max 512000) mis-ordered: 0 bytes
```

```
Current receive queue size in packets: 0 (max 60)


Num of peers with authentication info: 1
Authentication peer details:
Peer: 192.168.1.4/32, OBJ_ID: 0x40002fc0
Port: 4189, vrf_id: 0x60000000, type: AO, debug_on:0
Keychain_name: test11, options: 0x00000001, linked peer: 0x7fc864004770
Send_SNE: 0, Receive_SNE: 0, Send_SNE_flag: 0
Recv_SNE_flag: 0, Prev_send_seq: 3997493771, Prev_receive_seq: 401915492
ISS: 3997493682, IRS: 401915403
Current key: 1
Traffic keys: send_non_SYN: ffd64ea3, recv_non_SYN: 00000000
RNext key: 1
Traffic keys: send_non_SYN: 00000000, recv_non_SYN: 00000000
Last 1 keys used:
key: 1, time: Feb 26 05:36:17.344.766, reason: No current key set
```

The above sample displays only a part of the actual output; the actual output displays more details.

Configuring Graceful Restart

RSVP graceful restart provides a mechanism to ensure high availability (HA), which allows detection and recovery from failure conditions for systems running Cisco IOS XR software, and ensures non-stop forwarding services. RSVP graceful restart is based on RSVP hello messages and allows RSVP TE enabled routers to recover RSVP state information from neighbors after a failure in the network. RSVP uses a Restart Cap object (RSVP RESTART) in hello messages in which restart and recovery times are specified to advertise the restart capability of a node. The neighboring node helps a restarting node by sending a Recover Label object to recover the forwarding state of the restarting node.

You can configure standard graceful restart which is based on node-id address based hello messages and also interface-based graceful restart which is interface-address based hello messages.

Configuration Example

In this example, RSVP-TE is already enabled on the router nodes on a network and graceful restart needs to be enabled on the router nodes for failure recovery. Graceful restart is configured globally to enabled node-id address based hello messages and also on a router interface to support interface-address based hello messages.

```
Router# configure
Router(config)# rsvp
Router(config-rsvp)# signalling graceful-restart
Router(config-rsvp)# interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# signalling hello graceful-restart interface-based
Router(config-rsvp-if)# commit
```

Verification

Use the following commands to verify that graceful restart is enabled.

```
Router# show rsvp graceful-restart

 Graceful restart: enabled Number of global neighbors: 1
 Local MPLS router id: 192.168.55.55
 Restart time: 60 seconds  Recovery time: 120 seconds
```

```
 Recovery timer: Not running
 Hello interval: 5000 milliseconds Maximum Hello miss-count: 4

Router# show rsvp graceful-restart neighbors detail

 Neighbor: 192.168.77.77 Source: 192.168.55.55 (MPLS)
   Hello instance for application MPLS
     Hello State: UP        (for 00:20:52)
     Number of times communications with neighbor lost: 0
     Reason: N/A
Recovery State: DONE
   Number of Interface neighbors: 1
     address: 192.168.55.0
   Restart time: 120 seconds  Recovery time: 120 seconds
   Restart timer: Not running
   Recovery timer: Not running
   Hello interval: 5000 milliseconds  Maximum allowed missed Hello messages: 4
```

Change the Restart-Time: Example

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

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

Configure refresh reduction

Table 53: Feature History Table

Feature Name	Release Information	Feature Description
Set global RSVP message retransmission interval	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
Set global RSVP message retransmission interval	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.

Feature Name	Release Information	Feature Description
Set global RSVP message retransmission interval	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, P100])(select variants only*) This feature support is now extended to: • 8712-MOD-M • 8212-48FH-M • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E • 88-LC1-36EH
Set global RSVP message retransmission interval	Release 24.1.1	During Fast Reroute (FRR), an RSVP router sends multiple messages to neighbors. If a neighbor fails to acknowledge the messages due to an overload of RSVP message processing or a high frequency of failures, RSVP retransmits the messages, which can result in network congestion. You can now set a longer RSVP message retransmission interval to provide sufficient processing time for neighbors, reduce signaling overhead, and prevent network congestion. You can set this interval for all directly connected neighbors at once or remote neighbors connected through backup tunnels. Previously, you could only enable this option per interface. The feature introduces these changes: **CLI**: **signalling refresh reduction reliable retransmit-time (RSVP configuration)** **YANG Data Model**: `Cisco-IOS-XR-ip-rsvp-cfg.yang` (see GitHub, YANG Data Models Navigator)

RSVP Refresh Reduction improves the reliability of Resource Reservation Protocol (RSVP) signaling to enhance network performance and message delivery and it is enabled by default. Refresh reduction is used with a neighbor only if the neighbor supports it. You can also disable refresh reduction on an interface if you want.

This feature ensures reliable delivery of RSVP messages when network traffic is disrupted. To ensure that its message is delivered to its neighbor, RSVP requests the neighbor to send an acknowledgment message by a given time duration. If it doesn't receive the acknowledgment, it resends the message and doubles its current wait time. After 5 attempts, RSVP stops retransmitting the message to the neighbor.

Configuration Example

The example shows how to configure the various parameters available for the refresh reduction feature.

The following parameters are configured to change their default values:

- refresh interval
- number of refresh messages a node can miss
- retransmit time
- acknowledgment hold time
- acknowledgment message size
- refresh message summary size

```
Router# configure
Router(config)# rsvp
Router(config-rsvp)# interface HundredGigabitEthernet 0/0/0/3
Router(config-rsvp-if)# signalling refresh interval 40
Router(config-rsvp-if)# signalling refresh missed 6
Router(config-rsvp-if)# signalling refresh reduction reliable retransmit-time 2000
Router(config-rsvp-if)# signalling refresh reduction reliable ack-hold-time 1000
Router(config-rsvp-if)# signalling refresh reduction reliable ack-max-size 1000
Router(config-rsvp-if)# signalling refresh reduction summary  max-size 1500
Router(config-rsvp-if)# commit
```

Set Global RSVP Message Retransmission Interval

During FRR, RSVP reroutes tunnels with failed links. With the default RSVP message retransmission interval in use, when a large number of LSPs are rerouted simultaneously, an RSVP router sends a large number of messages to its neighbor – whether connected directly or through a backup tunnel – and this may overwhelm its processing load. While the neighbor is still processing messages, the RSVP router retransmits the earlier messages, further overwhelming it and causing network congestion.

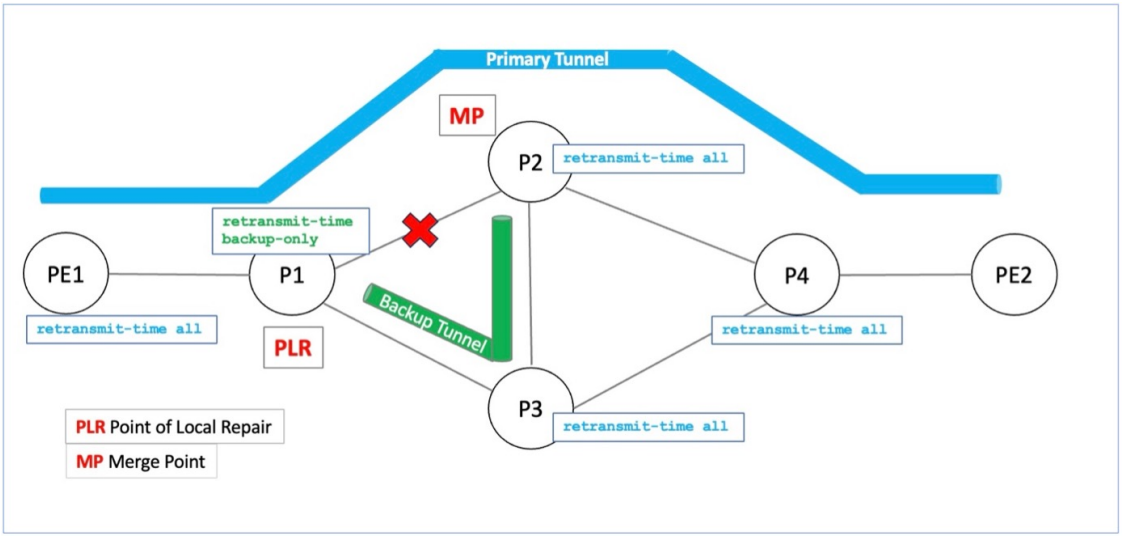
This feature allows you to configure a global timer with a longer retransmission interval so that retransmissions are relatively relaxed and provide more time for the neighbor to acknowledge messages.

To set the retransmission interval for messages sent over the backup tunnel during FRR, use the **backup-only** configuration which applies the RSVP-global configuration on backup tunnels. Another advantage of this configuration is that the interval is automatically applied to auto-backup tunnels, which are created dynamically.

Alternatively, to set the retransmission interval for all RSVP messages sent with reliable messaging (Path, Resv, PathTear, ResvTear, etc), use the **all** configuration which applies the RSVP-global configuration on all interfaces

Configuration Example for Set Global RSVP Message Retransmission Interval

Consider the following network topology.



Pointers:

- MPLS-TE, RSVP and Refresh-Reduction are enabled on all routers.
- The primary tunnel's path is PE1-P1-P2-P4-PE2.

  PE1 is the headend router, P1-to-P4 are Provider routers, and PE2 is the tail-end router.
- P1 is a potential Point of Local Repair (PLR) and headend of the backup tunnel with the path P1-P3-P2.

  ***PLR**: A router that creates a backup tunnel to handle link failures in the primary tunnel.*
- P2 is the Merge Point that merges the backup tunnel with the primary tunnel.

When P1 detects a link failure between P1-P2, it initiates FRR and reroutes the traffic through the backup tunnel.

In the following section, the **all** configuration is enabled on PE1, P2, P3 and P4, and the **backup-only** configuration is enabled on P1.

Configuration Steps:

1. **PE1, P2, P3 and P4 Configuration and Verification**

   A retransmission interval of 4000 milliseconds is set for all RSVP-enabled PE1 interfaces:

```
PE1(config)# rsvp
PE1(config-rsvp)# signalling refresh reduction reliable retransmit-time all 4000
PE1(config-rsvp)# interface hundredGigE 0/0/0/0
PE1(config-rsvp-if)# bandwidth 200000
PE1(config-rsvp-if)# signalling refresh interval 180
PE1(config-rsvp-if)# signalling refresh reduction bundle-max-size 1400
PE1(config-rsvp-if)# commit
```

   Similarly, enable configurations on P2, P3 and P4.

   Verification

```
PE1# show rsvp interface detail

INTERFACE: HundredGigE0/0/0/0
```

```
..
          Retransmit: 4000ms. (Global)
```

RSVP message retransmission count:

```
PE1# show rsvp counters messages
HundredGigE0/0/0/0   Recv      Xmit
                      ..        ..
    Retransmit              8
    ..
```

RSVP message retransmission to its neighbors:

```
PE1# show rsvp neighbors detail

Global Neighbor: 10.1.1.12
  Interface Neighbor: 10.2.2.10
  ..      Retransmitted messages: 34
```

Running Configuration

```
rsvp
  interface HundredGigE0/0/0/0
    bandwidth 200000
    signalling refresh interval 180
    signalling refresh reduction bundle-max-size 1400
    !
  signalling refresh reduction reliable retransmit-time all 4000
```

2. P1 Configuration and Verification

A retransmission interval of 5000 milliseconds is set for P1's backup tunnel interface.

Note This configuration is also applicable for auto-backup tunnels.

```
P1(config)# rsvp
P1(config-rsvp)# signalling refresh reduction reliable retransmit-time backup-only 5000

P1(config-rsvp)# interface hundredGigE 0/0/0/0
P1(config-rsvp-if)# bandwidth 200000
P1(config-rsvp-if)# signalling refresh interval 180
P1(config-rsvp-if)# signalling refresh reduction bundle-max-size 1400
P1(config-rsvp-if)# commit
```

Verification

```
P1# show rsvp interface detail

INTERFACE: HundredGigE0/0/0/0
..
                      Retransmit: 5000ms. (Global)
```

Running Configuration

```
rsvp
  interface HundredGigE0/0/0/0
    bandwidth 200000
    signalling refresh interval 180
    signalling refresh reduction bundle-max-size 1400
    !
```

```
signalling refresh reduction reliable retransmit-time backup-only 5000
```

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
```

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.

```
Router(config)# rsvp
Router(config-rsvp)# interface hundredGigE 0/0/0/0
Router(config-rsvp-if)# signalling refresh reduction disable
```

RSVP Prefix Filtering

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

Configuring ACL Based Prefix Filtering

You can configure extended access lists (ACLs) to forward, drop, or perform normal processing on RSVP router-alert (RA) packets. For each incoming RSVP RA packet, RSVP inspects the IP header and attempts to match the source or destination IP addresses with a prefix configured in an extended ACL. If there is no explicit permit or explicit deny, the ACL infrastructure returns an implicit deny by default. By default, RSVP processes the packet if the ACL match yields an implicit (default) deny.

Configuration Example

This example configures ACL based prefix filtering on RSVP RA packets. When RSVP receives a RA packet from source address 10.0.0.1 it is forwarded and packets destined to the IP address 172.16.0.1 are dropped.

```
Router# configure
Router(config)# ipv4 access-list rsvpac1
Router(config-ipv4-acl)# 10 permit ip host 10.0.0.1 any
Router(config-ipv4-acl)# 20 deny ip any host 172.16.0.1

Router# configure
Router(config)# rsvp
Router(config-rsvp)# signalling prefix-filtering access-list rsvp-acl
Router(config-rsvp)# commit
```

Verification

Verify the configuration of ACL based prefix filtering

```
Router# show rsvp counters prefix-filtering access-list rsvp-ac1

  ACL:rsvp-ac1            Forward          Local          Drop           Total
  Path                      0              0            0              0
```

```
PathTear                  0              0              0              0
ResvConfirm               0              0              0              0
Total                     0              0              0
```

Configuring RSVP Packet Dropping

You can configure extended access lists (ACLs) to forward, drop, or perform normal processing on RSVP router-alert (RA) packets. By default, RSVP processes the RA packets even if the ACL match yields an implicit deny. You can configure RSVP to drop RA packets when the ACL matches results in an implicit deny.

Configuration Example

This example configures ACL based prefix filtering on RSVP RA packets. When RSVP receives a RA packet from source address 10.0.0.1 it is forwarded and packets destined to the IP address 172.16.0.1 are dropped. RA packets are dropped if the ACL matches results in an implicit deny.

```
Router# configure
Router(config)# ipv4 access-list rsvpac1
Router(config-ipv4-acl)# 10 permit ip host 10.0.0.1 any
Router(config-ipv4-acl)# 20 deny ip any host 172.16.0.1
Router(config-ipv4-acl)# exit
Router(config)# rsvp
Router(config-rsvp)# signalling prefix-filtering default-deny-action drop
Router(config-rsvp)# commit
```

Verification

Verify the configuration of RSVP packet drop using the following command.

```
Router# show rsvp counters prefix-filtering access-list rsvpac1

  ACL: rsvpac1            Forward          Local          Drop           Total
  Path                      4              1            0              5
  PathTear                  0              0            0              0
  ResvConfirm               0              0            0              0
  Total                     4              1            0              5
```

Enabling RSVP Traps

By implementing the RSVP MIB, you can use SNMP to access objects belonging to RSVP. You can also specify two traps (NewFlow and LostFlow) which are triggered when a new flow is created or deleted. RSVP MIBs are automatically enabled when you turn on RSVP, but you need to enable RSVP traps.

Configuration Example

This example shows how to enable RSVP MIB traps when a flow is deleted or created and also how to enable both the traps.

```
Router# configure
Router(config)# snmp-server traps rsvp lost-flow
Router(config)# snmp-server traps rsvp new-flow
Router(config)# snmp-server traps rsvp all
Router(config)# commit
```

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.

Extended display and filtering options for interface monitoring

Extended display and filtering options for interface monitoring is an exec mode CLI command enhancement that

- displays full interface names in `brief` and `description` outputs using the `wide` keyword, and
- excludes tunnel interfaces from command output using the `no-tunnels` keyword.

You can use these options with existing show interfaces command variants to improve monitoring clarity in network environments with large numbers of interfaces, especially tunnel interfaces. The default behavior remains unchanged unless the new keywords are explicitly specified.

Table 54: Feature History Table

Feature Name	Release Information	Feature Description
Extended display and filtering options for interface monitoring	Release 26.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC:Q200], 8400 [ASIC: K100]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100] The **show interface brief** CLI command truncates interface names to 12 characters. For tunnel interfaces with similar prefixes, this truncation causes identical names to appear in the output, making it difficult to identify and monitor tunnels correctly. This feature removes the limitation by allowing network operators to view full interface names and exclude tunnel interfaces from the output when required. It introduces optional display and filtering keywords that expand interface name width and filter tunnel interfaces, while keeping the default command behavior unchanged unless the new options are explicitly used. This feature introduces these changes: **CLI:** The **wide** and **no-tunnels** keywords are introduced in the **show interfaces** command.

Extended display and filtering options for the show interfaces command

These **show interfaces** command combinations show how to display full interface names and how to exclude tunnel interfaces across different output formats.

Table 55: Show interface CLI command variants with wide and no-tunnel options

Command	Description
show interface brief wide	Displays the interface brief output with full interface names.

Command	Description
show interface brief no-tunnel	Displays the interface brief output while excluding tunnel interfaces.
show interface brief wide no-tunnel	Displays the interface brief output with full interface names while excluding tunnel interfaces.
show interface brief compact no-tunnel	Displays the interface brief compact output (bandwidth shown in Gbps) while excluding tunnel interfaces.
show interface description wide	Displays the interface description output with full interface names.
show interface description no-tunnel	Displays the interface description output while excluding tunnel interfaces.
show interface description wide no-tunnel	Displays the interface description output with full interface names while excluding tunnel interfaces.
show interface no-tunnel	Displays complete interface details while excluding tunnel interfaces.
show interface non-dynamic no-tunnel	Displays complete interface details while excluding tunnel and subscriber interfaces.

This example shows the output from the **show interfaces brief wide** command.

```
Router# show interfaces brief wide
Mon Jul 21 15:35:54.721 IST

Interface                                     Intf      LineP   Encap   MTU    BW      Reason
                                              State     State    Type  (byte) (Kbps)
-------------------------------------------------------------------------------------------

Null0                                          up        up        Null    1500   0

srte_test_backup_path_idx0002_TE64name
                                              down       down      TUNNEL  1500   0    Phy
member down

srte_test_main_path_idx0003_TE64name
                                              down       down      TUNNEL  1500   0

srte_test_primary_path_idx0001_TE64name
                                              down       down      TUNNEL  1500   0

GigabitEthernet0/0/0/0                        admin-down  admin-down     ARPA    1514 1000000
  Phy down

GigabitEthernet0/1/0/0                        admin-down  admin-down     ARPA    1514 1000000
  Phy down
```

This example shows the output from the **show interfaces description wide** command.

```
Router# show interfaces description wide
Mon Jul 21 15:35:54.721 IST

Interface                      Status      Protocol     Reason             Description
-------------------------------------------------------------------------------------------
srte_test_idx0002_TE64name   down        down         Phy member down

GigabitEthernet0/0/0/0      admin-down  admin-down    Phy down       uplink_to_core_router
```

```
GigabitEthernet0/0/0/0       admin-down  admin-down    Phy Down
Uplink_to_Distribution_Swi
tch_DS2_via_40G_QSFP_Fiber_Location:_Building_R2_Rack_17_|_Role:_Data_Center_Aggregation_Layer_link_with_Priority_QoS_Profile
```

RSVP for MPLS-TE Features - Details

RSVP Graceful Restart Operation

RSVP graceful restart is based on RSVP hello messages. Hello messages are exchanged between the router and its neighbor nodes. Each neighbor node 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. If the sending node supports state recovery, a Restart Cap object that indicates a node's restart capability is also carried in the hello messages. In the Restart Cap object, the restart time and the recovery time is specified. The restart time is the time after a loss in Hello messages within which RSVP hello session can be re-established. The recovery time is the time that the sender waits for the recipient to re-synchronize states after the re-establishment of hello messages.

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. If restart cap objects are sent to an RSVP neighbor and 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 a hello Request message is received from an unknown neighbor, no hello ACK is sent back.

RSVP Authentication

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. The integrity object includes a key ID, a sequence number for messages, and keyed message digest. 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 sending and receiving systems maintain a security association for each authentication key that they share. For detailed information about different security association parameters, see the **Security Association Parameters** table.

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.

Interface and neighbor interface 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.

The following situations 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.

A security association (SA) is a collection of information that is required to maintain secure communications with a peer. The following table lists the main parameters that defines a security association

Table 56: Security Association Parameters

Security Association Parameter	Description
src	IP address of the sender.
dst	IP address of the final destination.
interface	Interface of the security association.
direction	Send or receive type of the security association.
Lifetime	Expiration timer value that is used to collect unused security association data.
Sequence Number	Last sequence number that was either sent or accepted (dependent of the direction type).
key-source	Source of keys for the configurable parameter.
keyID	Key number (returned form the key-source) that was last used.
Window Size	Specifies the maximum number of authenticated messages that can be received out of order.
Window	Specifies the last *window size* value sequence number that is received or accepted.

MPLS-TE LSP OOR

The MPLS-TE LSP OOR function adds capability for the RSVP-TE control plane to track the LSP scale of transit routers, so that it can take a specific set of (pre-configured) actions when threshold limits are crossed, and inform other routers in the network. MPLS-TE keeps track of the number of transit LSPs set up through the router. The limits do not apply to ingress and egress LSP routers since they are driven by explicit configuration. In other words, the configuration determines how many egress or ingress LSPs a router has. For midpoint routers, the number is a function of the topology, the links metrics, and links' bandwidth.

State Transition Triggers - The LSP OOR state transition is triggered by checking the total transit LSP count and the unprotected count. If either count crosses the threshold, the state transition is triggered. If both counts cross the limit, the more critical state is chosen. Each limit will have a value for the *Yellow* threshold and a value for the *Red* threshold. When these thresholds are crossed, the configured MPLS-TE LSP OOR actions take effect. Similarly, the transition to *Green* state occurs when the LSP numbers drop.

LSP OOR State Dampening - The reason for LSP OOR State Dampening is that the number of accepted LSPs would be at the threshold and once an LSP is deleted, the state goes back from Red to Yellow, and a new LSP is setup and the state goes back to Red.

The solution is to introduce dampening when there is a state transition from Red to Yellow or from Yellow to Green. Whenever the transit number of LSPs crosses down a threshold, a timer is started for 10 seconds. After the timer expires, the new state is computed and moved to it. The timer is stopped if the transit number threshold is crossed (up) again. The transition from a state to a more severe state is not dampened.

Low and High Priority LSPs - When the LSP OOR is in yellow or red state, new high priority LSPs will not preempt low priority LSPs. Preemption can still occur but only for bandwidth reasons. In other words, if the router is in Red state where one of the actions is to reject any new LSP, the new high-priority LSPs are rejected even if there is an established low-priority LSP. The low-priority LSP is not removed to make room for the high-priority one.

Configuration Limit - Setting the configured limit to a value that is smaller than the current number of LSPs will trigger state transition but will not cause existing LSPs to be deleted or preempted. Setting the configured limit to a value that is larger than the current number of LSPs takes the node out of LSP OOR state. When an LSP cannot be admitted due to LSP OOR, the LSRs send Path Error messages to the LERs.

Event Logging - This is generated when the system transitions across OOR states, such as a resource change into an *yellow* or *red* state. Reporting level for *Red* is critical (1), and for yellow is warning (4). The following example shows that the count has crossed the threshold of 5000.

```
RP/0/RP1/CPU0:May 15 17:05:48 PDT: te_control[1034]: %ROUTING-MPLS_TE-4-LSP_OOR :

Transit LSP resources changed to Yellow.
Total transit: configured threshold 5000; actual count 5001;
Unprotected transit: configured threshold 4294967295; actual count 0
```

When the resource comes out of OOR, it will report as *green*.

Configuration Example

```
mpls traffic-eng
 lsp-oor
  green
   action accept reopt-lsp
   action flood available-bw 20
   recovery-duration
   action admit lsp-min-bw   X   -- > (in kbps, a lower limit than yellow and red state)

  yellow
   transit-all threshold 75000
   action accept reopt-lsp
   action flood available-bw 0
   action admit lsp-min-bw  Y

  red
   transit-all threshold 90000
   action flood available-bw 0
   action admit lsp-min-bw Z
```

The LSP OOR threshold values are set to yellow as 75000 and red as 90000. When these thresholds are crossed, corresponding actions are applied to all the TE interfaces.

Note The default values of the above thresholds are infinite.

When the LSP OOR *yellow* state is reached, the **accept reopt-lsp** action, **flood available-bw 0** action and **admit lsp-min-bw** actions are activated. This allows headend routers to reoptimize existing LSPs through, but doesn't allow new LSPs to get established. Also, MPLS-TE advertises zero bandwidth out of all interfaces, making this transit router less preferable for new LSPs. To handle a sudden burst of new LSPs that get signaled, the **action admit lsp-min-bw** function ensures only a small number of high bandwidth LSPs get provisioned through the affected router. When the red threshold state is crossed, the **flood available-bw 0** and **admit lsp-min-bw** actions prevent any additional or reoptimized transit LSPs from getting set up through the affected router.

Additional References

For additional information related to RSVP, refer to the following references:

Related Documents

Related Topic	Document Title
RSVP Infrastructure Commands	*RSVP Infrastructure Commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.
MPLS Traffic Engineering Commands	*MPLS Traffic Engineering commands* module in *MPLS Command Reference for Cisco 8000 Series Routers*.

Standards

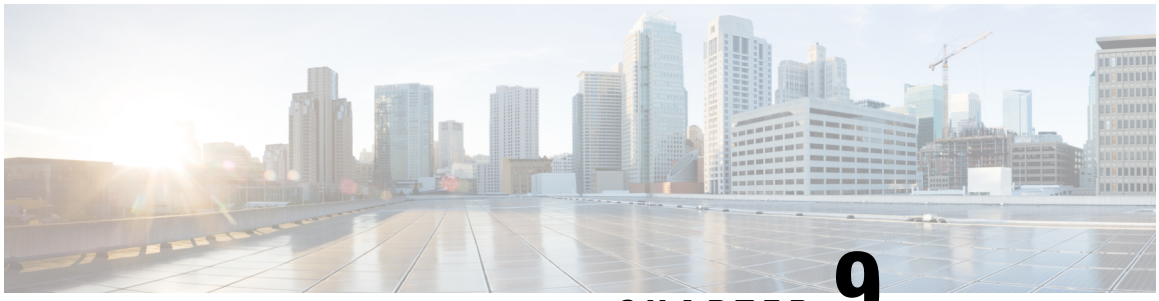
Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	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: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport



CHAPTER 9

MPLS MTU

Maximum transmission unit (MTU) is the maximum size, in bytes, of a packet or frame transmitted over a network. This chapter talks about MPLS MTU configuration for MPLS enabled interfaces.

Table 57: Feature History Table

Feature Name	Release Information	Feature Description
MPLS MTU	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8711-48Z-M routers.
MPLS MTU	Release 7.10.1	You can now enforce the size of MPLS frames in your MPLS network by configuring the MPLS MTU of the MPLS interface. This enables interworking with routers from other vendors that have a strict MPLS MTU enforcement. When a transit packet exceeds the MPLS MTU, the router either fragments the packet or drops it. Cisco 8000 Series Routers and Line Cards with Cisco Silicon One Q200, Q201, or Q202 ASIC support this feature. In earlier releases, Cisco 8000 Series Routers didn't support MPLS MTU feature. While interworking with other routers that had MPLS MTU configured, the Cisco 8000 Series Router sent packets exceeding the MPLS MTU of those router interfaces. This resulted in packet drops at the peer routers. You can configure MPLS MTU with either CLI or YANG data model: **CLI:** • **mpls mtu** **YANG Data Model:** • `Cisco-IOS-XR-um-if-mpls-cfg.yang` (see GitHub, YANG Data Models Navigator)

• Overview of MPLS MTU, on page 318
• Configuration Example for MPLS MTU, on page 320

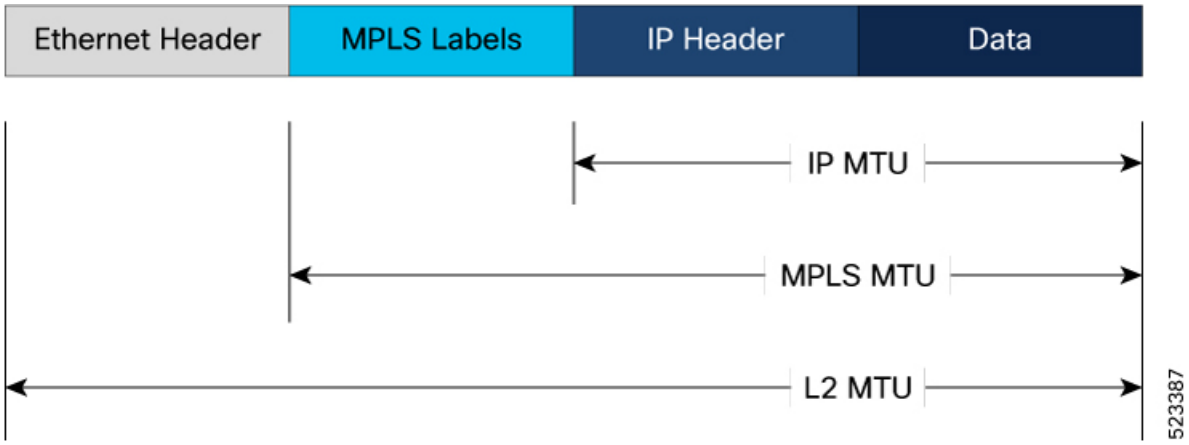
Overview of MPLS MTU

The MTU of an interface determines whether a packet is too large to be transmitted across it. You can configure three types of MTU values on a router interface:

- Layer 2 (L2) or Ethernet MTU which specifies the maximum transmittable size of the packet, measured from the ethernet header until the end of the packet. The default value of L2 MTU for a main interface is 1514 bytes. This value is configurable with the **mtu** command.
- IP MTU which specifies the maximum transmittable size of the packet, measured from the IP header until the end of the packet. This value is applicable only for unlabeled IPv4 or IPv6 packets. The default value of IP MTU is L2 MTU subtracted by 14 bytes which is the size of the ethernet header of the main interface. You can configure the IP MTU with the **ip mtu** command.
- MPLS MTU which specifies the maximum transmittable size of the packet measured from the MPLS labels until the end of the packet. This value is applicable only for labeled packets. The default value of MPLS MTU is L2 MTU subtracted by 14 bytes which is the size of the ethernet header of the main interface. You can configure the MPLS MTU with the **mpls mtu** command.

The following figure depicts the different portions of the egress packet that the router considers for MTU checks.

Figure 25: Different portions of the egress packet that the router considers for MTU checks



In an IPv4 network with MPLS enabled, the Label Switching Routers add one or more MPLS labels to all IPv4 packets. Each label is 4 bytes. So, a packet with *n* number of labels, would have *n* x 4 bytes added to it. Consider that the maximum size of an IPv4 packet on this network is 1592 bytes. If the router imposes a maximum of two labels on all IPv4 packets transmitted, then you can set the following values for L2 MTU and MPLS MTU:

- L2 MTU = 1592 + size of two MPLS labels + size of the ethernet header = 1592 + (2 x 4) + 14 = 1614 bytes
- MPLS MTU = 1592 + size of two MPLS labels = 1592 + (2 x 4) = 1600 bytes

Configuring these values ensures that the router can transmit all IPv4 packets without fragmentation or packet drops.

In cases where the MPLS frame size exceeds the MPLS MTU of the interface, the router sends the packet to the LC CPU for exception processing. Exception processing depends on whether the underlying packet is IPv4 or IPv6.

If an MPLS frame exceeds MPLS MTU and has an underlying IPv4 packet, then the LC CPU fragments or drops the packet based on the Don't Fragment (DF) bit. DF bit is a bit within the IP header that determines whether a router can fragment a packet. The packet originator usually sets the DF bit. But it's possible for any device in the path to set it too.

- If the DF bit isn't set, the LC CPU of the router fragments the packet. After fragmentation, the LC CPU attaches the MPLS header on each of the fragments, and then forwards each fragment to the destination. Since all fragments require their own header, it adds bandwidth overheads and is not desirable.
- If the DF bit is set, the LC CPU of the router drops the packet and generates ICMP unreachable messages which the router sends to the device where the packet originated.

If the MPLS frame that exceeded MPLS MTU has an underlying IPv6 packet, the LC CPU drops the packet and generates ICMPv6 packet-too-big messages which the router sends to the device where the packet originated.

Note There are two scenarios where the router applies a default value of MPLS MTU, that equals the L2 MTU subtracted by the ethernet header size of the main interface, which is 14 bytes:

- When you don’t configure the MPLS MTU for an MPLS enabled main interface or sub-interface.
- When you configure MPLS MTU to a value greater than the L2 MTU.

The **MTU** field in the output of the command **show interface** *interface-name* displays the L2 MTU of the interface.

```
Router#show interfaces fourHundredGigE 0/0/0/0
Thu Jul 27 05:04:54.955 UTC
FourHundredGigE0/0/0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is FourHundredGigE, address is 78dd.c079.8400 (bia 78dd.c079.8400)
  Internet address is Unknown
  MTU 1514 bytes, BW 400000000 Kbit (Max: 400000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 400000Mb/s, link type is force-up
  output flow control is off, input flow control is off
  loopback not set,
  Last link flapped 00:00:20
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
```

For more information on IP MTU and L2 MTU, refer the topics *IP MTU* and *Ethernet MTU* in the chapter *Configuring Ethernet Interfaces* in the *Interface and Hardware Component Configuration Guide for Cisco 8000 Series Routers*.

Restrictions for MPLS MTU

The following restrictions apply for MPLS MTU:

- Routers and line cards with the Cisco Silicon One Q200, Q201, or Q202 ASIC support the MPLS MTU feature. For more information on the routers and line cards with the Q200, Q201, or Q202 ASIC, refer *Cisco 8000 Series Routers Data Sheet*.
- If you configure MPLS MTU to a value greater than the L2 MTU, the router restricts the MPLS MTU to the value that equals the L2 MTU subtracted by the L2 header size of the main interface, which is 14 bytes.
- You can't configure MPLS MTU on Bridged Virtual Interfaces (BVI). Instead, the router applies a default MPLS MTU that equals the L2 MTU subtracted by the L2 header size of the BVI, which is 14 bytes.

Configuration Example for MPLS MTU

Configure **mpls mtu** command under **interface** configuration submode. The following configuration snippet sets the **mpls mtu** for the interface **FourHundredGigE 0/0/0/0** to 1400 bytes:

```
Router#config terminal
Router(config)#interface FourHundredGigE0/0/0/0
Router(config-if)#mpls mtu 1400
Router(config-if)#commit
```

Running Configuration

```
Router#show running-config interface FourHundredGigE0/0/0/0
interface FourHundredGigE0/0/0/0
 mpls
  mtu 1400
 !
!
```

Verification of MPLS MTU

The field starting with **mpls** or **Imposed MTU** in the output of the command **show im database interface** *interface-name* **detail | begin mpls** displays the configured value of MPLS MTU:

```
Router#show im database interface FourHundredGigE0/0/0/0 detail | begin mpls
  mpls             mpls (up, 1400)
    Views:                  LDP|G3P|L3P|OWN
    Owner Private:          144 bytes
    Instance ID:            -
    MPLS Label Stack Depth:
      Primary:              3
      Backup:               8
      SRTE:                 8
    Resource in NetIO:      TRUE
    NPU Mask (0/RP0/CPU0):  0x00000001 (1)
      ONLY IN VIEWS:          LDP
    Control flags:          0x00000030 (RDY|DNLD)
```

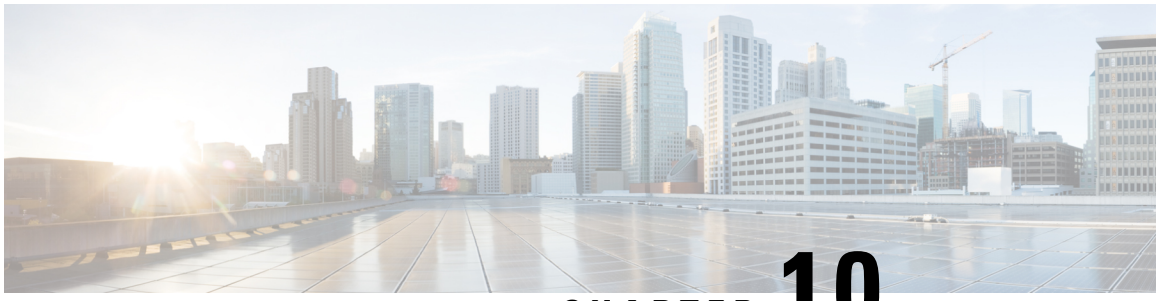
```
    Parent MTU:             1500
    Callback handle:        0x6001ff00
    Imposed MTU:            1400
    State constraint:       up
    Child MTU:              1400
  ether_sock      ether_sock (up, 1500)
    Views:                  LDP|G3P|L3P|OWN
    Owner Private:          144 bytes
    Instance ID:            -
    Resource in NetIO:      TRUE
    NPU Mask (0/RP0/CPU0):  0x00000001 (1)
      ONLY IN VIEWS:          LDP
    Control flags:          0x00000030 (RDY|DNLD)
    Parent MTU:             1500
    Callback handle:        0x6000bf00
    State constraint:       up
    Child MTU:              1500
```

You can also verify the MPLS MTU configuration of an MPLS-enabled interface with the command **show mpls interfaces** *interface-name* **private location** *location*.

, as shown:

```
Router#show mpls interfaces FourHundredGigE 0/0/0/0 private location 0/0/CPU0
Fri Nov  4 07:37:53.560 UTC
Interface      IFH         MTU   Label Stack Depth
                                  Prim Bkup Srte
-------------- ---------- ----- --------------------

FH0/0/0/0      0x01800220  1400     3    3    8
```



CHAPTER 10

Implementing GMPLS UNI

- Implementing GMPLS UNI, on page 323
- Configure GMPLS UNI, on page 331
- Verify GMPLS UNI Configurations , on page 334
- Additional References, on page 343

Implementing GMPLS UNI

Table 58: Feature History Table

Feature Name	Release Information	Feature Description
GMPLS User Network Interface (UNI) to Manage Optical Networks	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D
GMPLS User Network Interface (UNI) to Manage Optical Networks	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
GMPLS User Network Interface (UNI) to Manage Optical Networks	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
GMPLS User Network Interface (UNI) to Manage Optical Networks	Release 7.11.1	We have improved the dynamic optical transport connection provisioning between IP routers and optical network elements to reduce operational time and administrative overhead needed for new connectivity setup by enabling support for Generalized Multiprotocol Label Switching (GMPLS) User Network Interface (UNI), which enables advanced traffic engineering capabilities. This feature introduces the following changes: • **CLI** • New **gmpls uni** commands. • **YANG Data Model** • New XPaths for `Cisco-IOS-XR-um-mpls-te-cfg.yang` (see GitHub, YANG Data Models Navigator).

Restrictions for Implementing GMPLS UNI

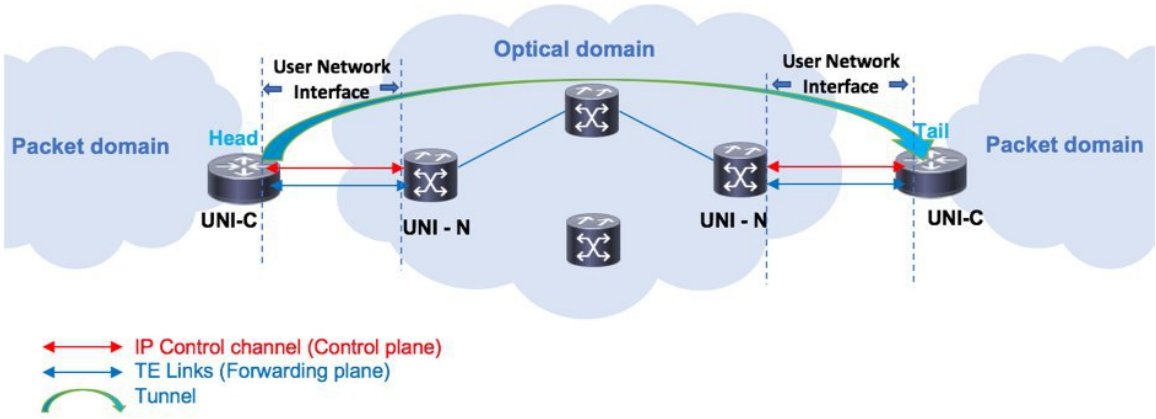
- The total number of configured GMPLS UNI controllers should not exceed the platform scale limit of 500 GMPLS interfaces.
- Each UNI-N (ingress or egress) should be routable from its adjacent UNI-C. The UNI-C nodes need to be routable from the UNI-N nodes too.
- GMPLS UNI is supported only over DWDM controllers and so, over POS and GigabitEthernet interfaces.

GMPLS UNI and GMPLS NNI

The GMPLS NNI optical network topology is known, and path calculations are performed at the NNI head. The GMPLS UNI optical network topology is unknown to the UNI-C nodes, and path calculations are performed by the UNI-N nodes.

GMPLS UNI Use Case

Figure 26: GMPLS UNI



The UNI components are UNI-N and UNI-C. The GMPLS 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. The UNI-C nodes are router nodes and UNI-N nodes are optical nodes.The tunnel originates on the headend UNI, depicted in the left part of the image. The tunnel terminates on the tailend UNI, depicted in the right part of the image. Enable the following configurations on the headend UNI and tailend UNI:

- Control plane - IP control channel between the UNI-C and UNI-N router IDs. This creates LMP adjacency over the control channel.
- Forwarding plane - TE Link between UNI-C and UNI-N optical interfaces.
- Tunnel configuration from Head UNI-C to a Tail UNI-C optical interface, over the optical network.

  For each tunnel, you must enable corresponding tunnel and TE link configurations.

Link Management Protocol (LMP) – LMP manages the control channel across the UNIs, verifies TE link connectivity between the UNI interfaces, and performs fault management.

Dynamic LMP – You can enable the Dynamic LMP function which validates LMP configuration consistency across the headend and tailend UNIs. Consistency check examples:

- You have configured one end of a TE link as an unnumbered interface, and the other end with an IP address.
- You have entered the wrong neighbor interface ID when configuring an unnumbered neighbor interface.

Ensure that you enable the preceding configurations correctly.

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 signaled 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.

Supported LSP encoding types, and corresponding switching types:

Switching Type	LSP Encoding Type
LSC	Lambda
FSC	EthernetType1, EthernetType2, Fiber
DCSC	EthernetType2

A packet network is switched across a fiber, optic, or data channel network. Enable the LSP encoding and switching types under the GMPLS UNI and LMP configuration modes. Also, enable the Generalized PID (G-PID) under the GMPLS UNI configuration mode. G-PID is an identifier of the type of payload that the LSP carries, and the LSP endpoints (the UNI-C devices) use.

The LSP encoding type, switching type, and G-PID are updated to the GMPLS label.

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.

GMPLS RSVP VRF Signaling

The Cisco IOS XR software supports a single non-default VRF for the GMPLS RSVP signaling. This allows GMPLS signaling to work even when the only available communication between the UNI-C and UNI-N nodes is through a VRF. This non-default VRF is supported only for GMPLS signaling; whereas the MPLS-TE signaling continues to support only the default VRF.

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 the following:

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

nLight Enhancements

These topics describe the enhancements made to nLight (also known as GMPLS UNI):

Explicit Route Object

Explicit Route Objects (EROs) limit LSP routing to a specified list of LSRs. Formerly, the UNI Client (UNI-C) node signaled a connection request, without an ERO, to the UNI Network (UNI-N) node. In this IOS XR Software release, the UNI-C node provides support for path message with ERO for GMPLS tunnels. This includes the capability to specify either a strict or a loose ERO to a path option to be included in the path message for processing by the ingress UNI-N.

An ERO in constructed using the strict and loose hops, specified in the explicit path, by the path option.

When a loose hop is configured, it identifies one or more transit LSRs which suggests the preferred path for the LSP. If a suggested path fails, another LSR is tried.

When a strict hop is configured, it identifies an exact path through which the LSP must be routed. Strict hop EROs specify the exact sequence of LSRs in the LSP.

As a result of these operations, a LSP is established from the sender to the destination of the session, following the explicitly routed path specified in the ERO.



Note

- *lockdown* and *verbatim* are mandatory in ERO path option.
- A path option may still be configured to use no ERO.

  In no ERO, *lockdown* is mandatory.

Wavelength Specification

The wavelength (also called label) specification enhancement enables the network planning tool to determine the wavelength, and specify the same at the UNI-C. The UNI-N then accepts the label provided by the UNI-C, or rejects the path entirely. Previously, the wavelength to be used for the GMPLS UNI tunnel was determined by the UNI-N, taking into account the headend UNI-C's capabilities.

The wavelength to be used is added to the path option configuration. This optional configuration allows a fixed wavelength to be specified for the path option.

When signaling using a path option with the specified wavelength takes place, the following changes happen because of the wavelength specification enhancement:

- The configured wavelength is validated against the controller’s capabilities; signaling fails if the wavelength cannot be used by the controller.
- The upstream label is set to the specified wavelength.
- The label-set in the Path message, instead of containing one label for each supported wavelength, contains only the specified wavelength.
- A path-error message with error code 25 and subcode 6 no longer receives special handling. If a suggested label is supplied, it is ignored.



Note A suggested label received in response to signaling with a path option that specifies a different label, is not stored for future use. Other path options, in general, have different constraints and therefore require path calculation to be redone.

Multiple Path Options

Multiple path options are permitted per GMPLS UNI tunnel. The index given to each path option indicates its relative preference level, with lower indices being preferred. This is similar to the existing multiple path option functionality available for packet TE. This allows the provision of multiple path options with, for example, progressively free constraints.

The path-option index is no longer fixed to ten and is now set by the user and distinguishes path options in the same manner as for packet tunnels. In all situations where a tunnel is being brought up or reoptimized, the path-option with the lowest index is tried first; if no LSP can be established with this path option, then subsequent path options are tried in ascending order. This also applies to recovery from failures, unless any recovery path option is specified.

Reoptimization

Reoptimization differs from restoration though the mechanisms involved are similar. Reoptimization occurs without the original connection having failed.

Unlike packet tunnels, reoptimization in GMPLS tunnels is not supposed to be loss free.

Manual Reoptimization

Manual reoptimization of a single GMPLS UNI tunnel can be triggered from the UNI-C node (headend). Use the **mpls traffic-eng optical-uni reoptimize tunnel-id** command to trigger manual reoptimization of a GMPLS UNI tunnel.

The manual trigger for reoptimization causes the currently established LSP to be torn down and signals a new LSP using the normal bring-up process (though the new LSP is same as the current one).

It is not possible to trigger reoptimization for multiple GMPLS UNI tunnels or at the tailend of a tunnel.

SRLG Discovery

Note Shared Risk Link Group (SRLG) discovery, SRLG collection and SRLG recording represent the same function.

The head and tail UNI-C routers have no direct knowledge of the path taken through the optical network by a GMPLS UNI tunnel, or of the properties of that path. All information about the path of a particular GMPLS UNI connection must therefore be explicitly requested and learned during the signaling process.

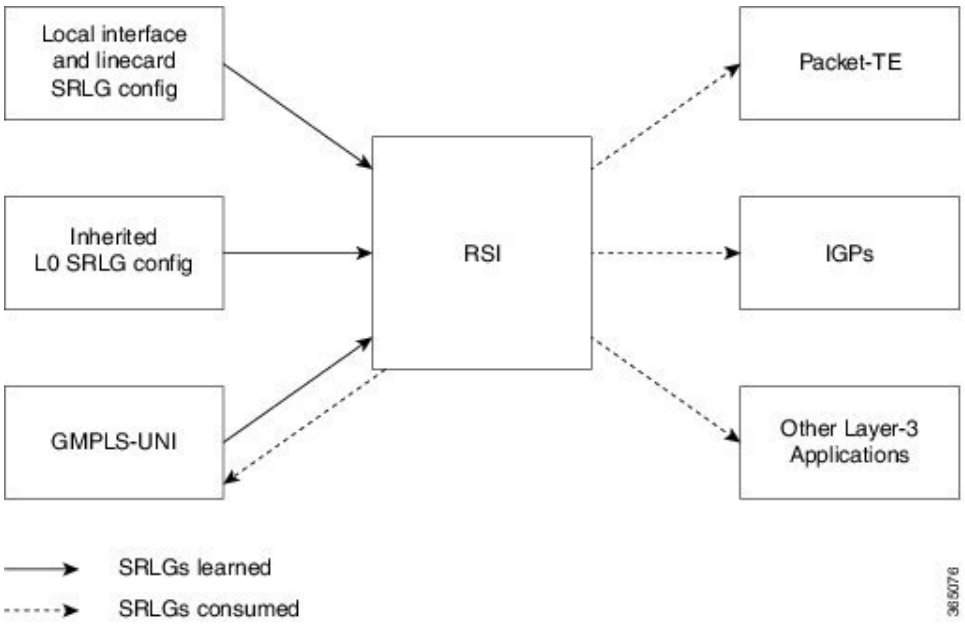
A key property of a GMPLS UNI connection is the set of SRLGs used by the optical links along the connection. It is necessary for the UNI-C routers to learn the set of SRLGs associated with a connection, so that this information can be used, both by GMPLS UNI in the specification of diversity requirements for other connections and by Layer-3 applications for effecting routing and protection decision making.

The learning of SRLGs during GMPLS UNI LSP signaling is done by requesting SRLG collection when LSP signaling is initiated, and by the addition of SRLG RRO sub-objects to the Path and Resv messages during signaling as described in IETF draft *SRLG-collect*. Path message learns egress interfaces from head to tail and Resv message learns egress interfaces from tail to head.

Provision of Discovered SRLGs to RSI

Once the SRLGs used by a GMPLS UNI connection are collected during signaling as in SRLG discovery, they are made available to the Layer-3 processes. This is done through RSI (Router Space Infrastructure), as illustrated in the following diagram:

Figure 27: SRLG Communication



An API is provided by the RSI component to allow SRLGs discovered during GMPLS UNI signaling to be communicated to RSI, as documented in IETF draft *RSI-SRLG*. RSI combines the SRLG sets learned from GMPLS and configuration for an interface and deliver a single set of SRLGs to applications registered as SRLG clients.

The SRLGs discovered during GMPLS UNI signaling are given to RSI for application to the Layer-3 interface of the DWDM controller associated with the GMPLS UNI tunnel. This may be a POS, GigE or an OTN interface.

SRLG Announce

All SRLGs discovered through GMPLS signaling are announced to RSI once the tunnel is up. These SRLGs are withdrawn from RSI when the tunnel goes down.

SRLG Diversity

Note SRLG diversity and SRLG exclusion represent the same function.

Support is added for signaling SRLG based diversity requirements, based on the XRO SRLG sub-object defined in RFC 4874. The use of SRLGs removes the restrictions of LSP based diversity, as SRLGs are flooded throughout the optical network, and by their very nature, reduce the risk of concurrent failure.

SRLG diversity is configured under the XRO attribute-set.

Head UNI-C Behavior

SRLG diversity is configured at the tunnel head. Individual SRLG exclusions are added to an XRO attribute-set; each is specified as either *best-effort* or mandatory (*strict*). Whenever any exclusion is specified, an XRO object is added to the Path message by the head UNI-C. The XRO contains a SRLG sub-object for each specified SRLG. The SRLG exclusions may coexist in the same XRO with LSP exclusions.

The XRO attribute-set is associated with tunnel path options in the same manner as for LSP exclusions.

If a SRLG with a strict exclusion matches an SRLG configured on the local DWDM controller, the bring-up attempt fails.

The SRLG exclusions requested by the head UNI-C are processed by the ingress UNI-N node during path calculation for the tunnel.

Tail UNI-C Behavior

On receiving a Path message containing an XRO, the tail UNI-C inspects each SRLG sub-object. If a SRLG sub-object, with a strict exclusion, matches an SRLG configured on the local DWDM controller, the Path message is rejected and a path-error is generated with error codes. No action is taken if the SRLG sub-object specifies a *best-effort* exclusion.

Multi-Layer Restoration - Optical

Multi-Layer Restoration-Optical (MLR-O) involves restoration from failures in the optical network that can leverage the same router interfaces at both ends.

Optical restoration involves the repair of a failure by the optical network locally. Although the routers may see loss of light until the failure is repaired, there is no signaling involving the routers, and from the routers perspective the GMPLS UNI LSP remains unchanged.

Optical Restoration: Same Wavelength

When a failure occurs on a physical link within the optical network, the routers identify that the link is down and Layer 3 protection mechanisms, such as FRR, are used to minimize the traffic loss. The optical network re-routes the GMPLS connection to an alternative path. This is done without any involvement of the routers.

Limitation - A significant limitation of optical restoration in this case, is that the wavelength in use for the connection cannot be changed. This is because the wavelength must be the same along the entire path and cannot be changed without end-to-end signaling. The constraints imposed on the connection during its initial signaling are also unchanged, which may reduce the chance of finding an alternative path.

Optical Restoration: Wavelength Change

Optical restoration may occur with an associated wavelength change, in the case where the optical network finds an alternative path with the same constraints as were originally signaled, but using a different wavelength. Some signaling is required, since the wavelength (and therefore the labels) used by the GMPLS connection are to change.

Consider a failure within the optical network on the path of a GMPLS UNI LSP. The restoration proceeds as in the previous case (same wavelength), but the new path found, uses a different wavelength. The ingress UNI-N then sends a path-error message indicating the new wavelength to be used; this has error code 24 (routing), sub-error 6 (unacceptable label set) and contains a suggested-label sub-object with the new label to be used. The head UNI-C then signals a new LSP with the new wavelength.

Although the wavelength in use may change in this case, the constraints used in signaling the original LSP remain unchanged.

Configure GMPLS UNI

Configure TE for GMPLS UNI

TE configuration specific to packet tunnels does not affect GMPLS UNI tunnels.

Configure 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.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# gmpls optical-uni
Router(config-te-gmpls)# controller dwdm 0/1/0/1
```

Configure the 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 you configure the tunnel properties, the incoming path messages are rejected and any existing tailend tunnel is torn down.

```
Router(config-te-gmpls-cntl)# mtu 9000
Router(config-te-gmpls-cntl)# tunnel-properties
Router(config-te-gmpls-tun)# g-pid 37
Router(config-te-gmpls-tun)# encoding-type lambda
Router(config-te-gmpls-tun)# tunnel-id 100
Router(config-te-gmpls-tun)# logging events lsp-status state
Router(config-te-gmpls-tun)# destination ipv4 unicast 10.10.3.4
Router(config-te-gmpls-tun)# exit
```

This configuration assigns the Lambda switching type for transporting traffic.

```
Router(config-te-gmpls-cntl)# switching-type lsc
```

(Optional) Tunnel Properties

```
Router(config-te-gmpls-cntl)# tunnel-properties
Router(config-te-gmpls-tun)# priority 3 2
Router(config-te-gmpls-tun)# record-route
Router(config-te-gmpls-tun)# signalled-name sign1
Router(config-te-gmpls-tun)# exit
```



Note If you remove the GMPLS UNI and GMPLS UNI controller submodes, all GMPLS UNI tunnels are removed.

Path Options - These configurations specify a path-option for the GMPLS tunnel headend router.

This configures a path option with no ERO.

```
Router(config-te-gmpls-tun)# path-option 30 no-ero lockdown
```

This configures a path option with an ERO for the GMPLS tunnel. You can specify an explicit path in the range 1 to 1000. **lockdown** and **verbatim** are mandatory in an ERO path option.

```
Router(config-te-gmpls-tun)# path-option 10 explicit name explicit_path_a lockdown verbatim
```

This configures a wavelength for the path option. The DWDM channel number range is 1 to 89. The DWDM channel number configured is formulated as 61-channel number. For example, for channel number 42 (in the supported channel list), the configured "DWDM channel number" is 61 - 42 = 19.

```
Router(config-te-gmpls-tun)# path-option 10 explicit name exp_all_loose_hop signaled-label
 dwdm wavelength 10 lockdown verbatim
```

This configures multiple path options for a single tunnel.

```
Router(config-te-gmpls-tun)# path-option 10 explicit name explicit_path_a lockdown verbatim

Router(config-te-gmpls-tun)# path-option 20 explicit name explicit_path_b lockdown verbatim
Router(config-te-gmpls-tun)# path-option 30 no-ero lockdown
```

This configures the LSP diversity function. You can configure an XRO attribute-set as part of the path-option for MPLS-TE, and specify exclusions for an attribute set for LSP diversity.

```
Router(config-te-gmpls-tun)# path-option 10 no-ero xro-attribute-set A01 lockdown
```

This configures the connection diversity function. You can specify exclusions for an attribute set for LSP diversity.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# attribute-set xro attrset01
Router(config-te-attribute-set)# exclude best-effort lsp source 10.10.1.2 destination
10.20.4.4 tunnel-id 17 extended-tunnel-id 10.20.3.3 lsp-id 17
Router(config-te-attribute-set)# commit
```

Configure LMP for GMPLS UNI

To implement LMP configuration for GMPLS UNI, follow these steps:

LMP Configuration - Configure the optical router ID and LMP neighbor:

```
Router# configure
Router(config)# lmp
Router(config-lmp)# gmpls optical-uni
```

```
Router(config-lmp-gmpls-uni)# router-id ipv4 unicast 10.10.4.4
Router(config-lmp-gmpls-uni)# neighbor nbr1
Router(config-lmp-gmpls-uni-nbr1)# dynamic
Router(config-lmp-gmpls-uni-nbr1)# hello 1000 10000
Router(config-lmp-gmpls-uni-nbr-nbr1)# ipcc routed
Router(config-lmp-gmpls-uni-nbr-nbr1)# router-id ipv4 unicast 10.10.4.5
```

LMP Controller Configuration - Configure the LMP link for the GMPLS UNI controller:

```
Router(config-lmp-gmpls-uni)# controller dwdm 0/4/0/0
Router(config-lmp-gmpls-uni-cntl)# neighbor nbr1
Router(config-lmp-gmpls-uni-cntl)# link-id ipv4 unicast 10.2.2.4
Router(config-lmp-gmpls-uni-cntl)# switching-type lsc
Router(config-lmp-gmpls-uni-cntl)# encoding-type Lambda
Router(config-lmp-gmpls-uni-cntl)# neighbor link-id ipv4 unicast 10.2.2.5
Router(config-lmp-gmpls-uni-cntl)# neighbor interface-id unnumbered 17
Router(config-lmp-gmpls-uni-cntl)# commit
```

Configure 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.

```
Router# configure
Router(config)# rsvp
Router(config-rsvp)# controller dwdm 0/1/0/1
Router(config-rsvp-cntl)# signalling refresh out-of-band interval 200
Router(config-rsvp-cntl)# signalling refresh out-of-band missed 30
Router(config-rsvp-cntl)# commit
```

Configure SRLG

SRLG discovery on the head router of a nLight tunnel.

- SRLG discovery/recording is enabled only on the headend for each tunnel.
- SRLG discovery/recording allows a maximum of 62 SRLGs in RSVP, which is different from the maximum count of 64 in RSI.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# gmpls optical-uni
Router(config-te-gmpls)# controller dwdm 0/1/0/1
Router(config-te-gmpls-cntl)# tunnel-properties
Router(config-te-gmpls-tun)# logging events lsp-status state
Router(config-te-gmpls-tun)# tunnel-id 100
Router(config-te-gmpls-tun)# record srlg
Router(config-te-gmpls-tun)# destination ipv4 unicast 10.10.3.4
Router(config-te-gmpls-tun)# path-option 10 no-ero lockdown
Router(config-te-gmpls-tun)# exit
```

To announce the discovered SRLGs to the system, use the **announce srlgs** command.

```
Router(config-te-gmpls-cntl)# announce srlgs
Router(config-te-gmpls-cntl)# commit
```

You can configure the attribute-set and SRLG path diversity with the best-effort or strict exclusion function.

```
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# attribute-set xro exclude_srlgs
Router(config-te-attribute-set)# exclude best-effort srlg value 21
Router(config-te-attribute-set)# exit
```

Associate the XRO attribute set to the GMPLS UNI tunnel through the path option. The path-option range is 1 to 1000.

```
Router(config-mpls-te)# gmpls optical-uni
Router(config-te-gmpls)# controller dwdm 0/1/0/1
router(config-te-gmpls-cntl)# tunnel-properties
Router(config-te-gmpls-tun)# path-option 10 no-ero xro-attribute-set exclude_srlgs lockdown
```

Verify GMPLS UNI Configurations

Various GMPLS UNI function configurations are displayed in this section:

Explicit Route Object (ERO)

```
Router# show mpls traffic-eng tunnels 1001 detail

Name: GMPLS-UNI-dwdm0_3_0_0 Destination: 172.16.0.1 Signalled-Name: head_ot1001_172.16.0.1

GMPLS UNI tunnel controlling link dwdm0/3/0/0, tunnel-id: 1001
Status:
Admin:    up Oper:   up   Path:  valid   Signalling: connected
path option 10, (LOCKDOWN verbatim) type explicit explicit_path_a (Basis for Setup, path
weight 0)
G-PID: 0x0800 (derived from egress interface properties) Creation Time: Fri Jul 17 08:41:21
 ---- (3d07h ago)
..
Current LSP Info:
  Instance: 20
  Uptime: 00:00:33 (since Mon Jul 20 ---- 15:45:22)

Upstream label:

  Optical label:
  Grid              : DWDM
  Channel spacing   : 50GHz
  Identifier        : 0
  Channel Number    : 60

Downstream label:

  Optical label:
  Grid              : DWDM
  Channel spacing   : 50GHz
  Identifier        : 0
  Channel Number    : 60

Router-IDs: local   10.0.0.1
downstream          172.16.0.1
Soft Preemption: None
SRLGs: not collected
Path Info:
  Outgoing:
    Explicit Route:
        Strict, 10.10.10.2
        Strict, 10.11.11.3
        Strict, 10.12.12.3
```

Wavelength

The following sequence of examples show how to add a wavelength to a path option for a GMPLS tunnel and verify the outgoing label is set accordingly.

This example shows how to configure a GMPLS tunnel with no ERO path option.

```
gmpls optical-uni
  controller dwdm0/3/0/0
   tunnel-properties
    tunnel-id 1001
    destination ipv4 unicast 172.16.0.1
    path-option 10 no-ero lockdown
   !
```

This example shows how to verify the default values for the outgoing label (UNI-N source, channel number same as Default Channel) and the list of valid wavelengths.

```
Router# show mpls traffic-eng link-management optical-uni controller dwdm 0/3/0/0

Optical interface: dwdm0/3/0/0
  Overview:
    IM state: Up
    Child interface: POS0_3_0_0: IM state Up
    OLM/LMP state: Up
    Optical tunnel state: up
  Connection:
    Tunnel role: Head
    Tunnel-id: 1001, LSP-id 21, Extended tunnel-id 10.0.0.1
    Tunnel source: 10.0.0.1, destination: 172.16.0.1
    Optical router-ids: Local: 10.0.0.1, Remote: 172.16.0.1
    Label source: UNI-N
    Upstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 60
    Downstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 60
    SRLG discovery: Disabled
    SRLG announcement: None
...
  Optical capabilities:
    Controller type: DWDM
    Channel spacing: 50 GHz
    Default channel: 60
    89 supported channels:
      -28, -27, -26, -25, -24, -23, -22, -21
      -20, -19, -18, -17, -16, -15, -14, -13
      -12, -11, -10, -9, -8, -7, -6, -5
      -4, -3, -2, -1, 0, 1, 2, 3
      4, 5, 6, 7, 8, 9, 10, 11
      12, 13, 14, 15, 16, 17, 18, 19
      20, 21, 22, 23, 24, 25, 26, 27
      28, 29, 30, 31, 32, 33, 34, 35
      36, 37, 38, 39, 40, 41, 42, 43
      44, 45, 46, 47, 48, 49, 50, 51
      52, 53, 54, 55, 56, 57, 58, 59
      60
```

```
    Controller SRLGs
      None
```

This example shows how to set valid wavelength 10 (61 - 51) for the current path.

```
gmpls optical-uni
  controller dwdm0/3/0/0
   tunnel-properties
    tunnel-id 1001
    destination ipv4 unicast 172.16.0.1
   path-option 10 explicit name explicit_all_loose_multi_hop signaled-label dwdm wavelength
 10 lockdown verbatim
   !
```

This example shows how to verify that the tunnel is up and the specified wavelength is used (label source is UNI-C and outgoing label is 51).

```
Router# show mpls traffic-eng link-management optical-uni controller dwdm 0/3/0/0

Optical interface: dwdm0/3/0/0
  Overview:
    IM state: Up
    Child interface: POS0_3_0_0: IM state Up
    OLM/LMP state: Up
    Optical tunnel state: up
  Connection:
    Tunnel role: Head
    Tunnel-id: 1001, LSP-id 23, Extended tunnel-id 10.0.0.1
    Tunnel source: 10.0.0.1, destination: 172.16.0.1
    Optical router-ids: Local: 10.0.0.1, Remote: 172.16.0.1
    Label source: UNI-C
    Upstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 51
    Downstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 51
    SRLG discovery: Disabled
    SRLG announcement: None
...
  Optical capabilities:
    Controller type: DWDM
    Channel spacing: 50 GHz
    Default channel: 60
    89 supported channels:
      -28, -27, -26, -25, -24, -23, -22, -21
      -20, -19, -18, -17, -16, -15, -14, -13
      -12, -11, -10, -9, -8, -7, -6, -5
      -4, -3, -2, -1, 0, 1, 2, 3
      4, 5, 6, 7, 8, 9, 10, 11
      12, 13, 14, 15, 16, 17, 18, 19
      20, 21, 22, 23, 24, 25, 26, 27
      28, 29, 30, 31, 32, 33, 34, 35
      36, 37, 38, 39, 40, 41, 42, 43
      44, 45, 46, 47, 48, 49, 50, 51
      52, 53, 54, 55, 56, 57, 58, 59
      60
```

```
    Controller SRLGs
      None
```

This example shows how to verify the upstream label on the tunnel tail.

```
Router# show mpls traffic-eng link-management optical-uni controller dwdm 0/3/0/0

Optical interface: dwdm0/3/0/0
  Overview:
    IM state: Up
    Child interface: POS0_3_0_0: IM state Up
    OLM/LMP state: Up
    Optical tunnel state: up
  Connection:
    Tunnel role: Tail
    Tunnel-id: 1001, LSP-id 23, Extended tunnel-id 10.0.0.1
    Tunnel source: 10.0.0.1, destination: 172.16.0.1
    Optical router-ids: Local: 172.16.0.1, Remote: 10.0.0.1
    Label source: UNI-N
    Upstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 51
    Downstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 51
    SRLG discovery: Disabled
    SRLG announcement: None
…..
  Optical capabilities:
    Controller type: DWDM
    Channel spacing: 50 GHz
    Default channel: 60
    89 supported channels:
      -28, -27, -26, -25, -24, -23, -22, -21
      -20, -19, -18, -17, -16, -15, -14, -13
      -12, -11, -10, -9, -8, -7, -6, -5
      -4, -3, -2, -1, 0, 1, 2, 3
      4, 5, 6, 7, 8, 9, 10, 11
      12, 13, 14, 15, 16, 17, 18, 19
      20, 21, 22, 23, 24, 25, 26, 27
      28, 29, 30, 31, 32, 33, 34, 35
      36, 37, 38, 39, 40, 41, 42, 43
      44, 45, 46, 47, 48, 49, 50, 51
      52, 53, 54, 55, 56, 57, 58, 59
      60
    Controller SRLGs
      None
```

Multiple Path Options

This example shows how to configure multiple path options.

```
mpls traffic-eng
 gmpls optical-uni
   controller dwdm0/2/0/2
    tunnel-properties
     path-option 10 explicit name explicit_path_a lockdown verbatim
     path-option 20 explicit name explicit_path_b lockdown verbatim
```

```
     path-option 30 no-ero lockdown
    !
```

The following sequence of examples show how to configure a GMPLS tunnel, add a new path option with a lower index than the path option in use, flap the tunnel and verify that the new path option (with a lower index) is used.

This example shows how to configure a GMPLS tunnel with one path option.

```
gmpls optical-uni
  controller dwdm0/3/0/0
   tunnel-properties
    tunnel-id 1001
    destination ipv4 unicast 172.16.0.1
    path-option 10 explicit name explicit_path_a lockdown verbatim
   !
```

This example shows how to verify the tunnel path and status with a show command.

```
Router# show mpls traffic-eng tunnels 1001 detail

Name: GMPLS-UNI-dwdm0_3_0_0  Destination: 172.16.0.1
  Signalled-Name: head_ot1001_172.16.0.1
GMPLS UNI tunnel controlling link dwdm0/3/0/0, tunnel-id: 1001
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10, (LOCKDOWN verbatim) type explicit explicit_path_a (Basis for Setup,
path weight 0)
    G-PID: 0x0800 (derived from egress interface properties)
    Creation Time: Fri Jul 17 08:41:21 ---- (3d06h ago)
..
```

This example shows how to add another path option with a lower index.

```
gmpls optical-uni
  controller dwdm0/3/0/0
   tunnel-properties
    tunnel-id 1001
    destination ipv4 unicast 172.16.0.1
    path-option 1 no-ero lockdown
    path-option 10 explicit name explicit_path_a lockdown verbatim
   !
```

Flag the tunnel (or trigger reoptimization) and verify that the tunnel comes up on the path with a lower index.

```
Router# show mpls traffic-eng tunnels 1001 detail

Name: GMPLS-UNI-dwdm0_3_0_0  Destination: 172.16.0.1
  Signalled-Name: head_ot1001_172.16.0.1
GMPLS UNI tunnel controlling link dwdm0/3/0/0, tunnel-id: 1001
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (LOCKDOWN) type no-ero  (Basis for Setup, path weight 0)
    Last Signalled Error : Mon Jul 20 17:03:00 2015
      Info: [24] PathErr(2,2)-(Admin, reason unknown) at 50.0.0.2
    path option 10, (LOCKDOWN verbatim) type explicit explicit_all_loose_multi_hop
    Last Signalled Error : Mon Jul 20 17:03:00 ----
      Info: [25] PathErr(2,2)-(Admin, reason unknown) at 50.0.0.2
```

SRLG

SRLG Discovery

```
Router# show mpls traffic-eng tunnels 100 detail

Name: GMPLS-UNI-dwdm0_2_0_0  Destination: 192.168.1.2
  Signalled-Name: rtrA_ot100_192.168.1.2
GMPLS UNI tunnel controlling link dwdm0/2/0/0, tunnel-id: 100
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10, (LOCKDOWN) type no-ero  (Basis for Setup, path weight 0)
    G-PID: 0x0800 (derived from egress interface properties)
    Creation Time: Mon Jul 20 19:32:03 ---- (00:48:02 ago)
  Config Parameters:
    Priority:  7  7 Affinity: 0x0/0xffff
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Enabled
….
 Soft Preemption: None
    SRLGs: mandatory collection
    Path Info:
  ….
    Resv Info:
      Record Route:
        IPv4 10.10.10.2, flags 0x0
        SRLGs: 21, 22, 23, 24
      Fspec: avg rate=10000 kbits, burst=1000 bytes, peak rate=10000 kbits
Displayed 1 (of 3) heads, 0 (of 0) midpoints, 0 (of 2) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

This example shows how to verify SRLG discovery configuration at the headend and the tailend. The output shows the list of SRLGs.

```
Router# show srlg

  System Information::
  Interface Count          : 4 (Maximum Interfaces Supported 512)
  Group Count              : 0 (Maximum Groups Supported 50)
  Inherit Location Count   : 0 (Maximum Inherit Locations Supported 10)
  Optical Interfaces Count : 4 (Maximum Optical Interfaces Supported 500)

  Interface    : GigabitEthernet0/2/0/0, Value Count : 10, Registrations : 2
  SRLG Values  : 11, 12, 13, 14, 15
                 21, 22, 23, 24, 25                                          -> Note:
These are announced srlgs.

  Interface    : GigabitEthernet0/2/0/2, Value Count : 0, Registrations : 1
  SRLG Values  :

  Interface    : GigabitEthernet0/2/0/4, Value Count : 0, Registrations : 2
  SRLG Values  :

  Interface    : GigabitEthernet0/2/0/5, Value Count : 0, Registrations : 2
  SRLG Values  :

  Optical Interface: dwdm0/2/0/0, Value Count : 5, References: 2
  SRLG Values      : 11, 12, 13, 14, 15                                      -> Note:
These are locally configured srlgs for controller (dwdm)

  Optical Interface: dwdm0/2/0/1, Value Count : 0, References: 1
  SRLG Values      :

  Optical Interface: dwdm0/2/0/2, Value Count : 0, References: 1
  SRLG Values      :
```

```
  Optical Interface: dwdm0/2/0/3, Value Count : 0, References: 1
  SRLG Values      :
```

SRLG Announce

```
Router# show srlg

  System Information::
  Interface Count          : 2 (Maximum Interfaces Supported 512)
  Group Count              : 0 (Maximum Groups Supported 50)
  Inherit Location Count   : 0 (Maximum Inherit Locations Supported 10)
  Optical Interfaces Count : 5 (Maximum Optical Interfaces Supported 500)

  Interface    : GigabitEthernet0/2/0/4, Value Count : 0, Registrations : 2
  SRLG Values  :

  Interface    : GigabitEthernet0/2/0/5, Value Count : 0, Registrations : 2
  SRLG Values  :

  Interface: GigabitEthernet0/2/0/0, Value Count : 4, References: 1
  SRLG Values      : 21, 22, 23, 24

  Optical Interface: dwdm0/2/0/0, Value Count : 3, References: 2
  SRLG Values      : 11, 12, 13

  Optical Interface: dwdm0/2/0/1, Value Count : 0, References: 1
  SRLG Values      :

  Optical Interface: dwdm0/2/0/2, Value Count : 0, References: 1
  SRLG Values      :

  Optical Interface: dwdm0/2/0/3, Value Count : 0, References: 1
  SRLG Values      :
```

SRLG Diversity

```
Router# show mpls traffic-eng tunnels 100 detail

Name: GMPLS-UNI-dwdm0_2_0_0  Destination: 192.168.1.2
  Signalled-Name: rtrA_ot100_192.168.1.2
GMPLS UNI tunnel controlling link dwdm0/2/0/0, tunnel-id: 100
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10, (LOCKDOWN) type no-ero  (Basis for Setup, path weight 0)
      XRO attribute-set: exclude_srlgs
        Best-effort, SRLG id 21
    Last Signalled Error : Mon Jul 20 20:55:33 ----
      Info: [5] PathErr(24,67)-(routing, route blocked by exclude route) at 10.10.10.2
    G-PID: 0x0800 (derived from egress interface properties)
    Creation Time: Mon Jul 20 19:32:03 2015 (01:25:19 ago)
  Config Parameters:
    Priority:  7  7 Affinity: 0x0/0xffff
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Enabled
  Binding Label: 0
  History:
    Tunnel has been up for: 00:00:23 (since Mon Jul 20 20:56:59 EDT 2015)
    Current LSP:
      Uptime: 00:00:23 (since Mon Jul 20 20:56:59 EDT ----)
  Current LSP Info:
    Instance: 6
```

```
    Uptime: 00:00:23 (since Mon Jul 20 20:56:59 EDT ----)
    Upstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 16
    Downstream label:
      Optical label:
      Grid                : DWDM
      Channel spacing     : 50 GHz
      Identifier          : 0
      Channel Number      : 16
    Router-IDs: local      192.168.1.1
                downstream 192.168.1.2
    Soft Preemption: None
    SRLGs: mandatory collection
    Path Info:
      Outgoing:
        No ERO
      Route Exclusions:
        Best-effort, SRLG id 21
      Record Route: Disabled
      Tspec: avg rate=10000 kbits, burst=1000 bytes, peak rate=10000 kbits
      Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
    Resv Info:
      Record Route:
        IPv4 10.10.10.2, flags 0x0
        SRLGs: 21, 22, 23, 24
      Fspec: avg rate=10000 kbits, burst=1000 bytes, peak rate=10000 kbits
Displayed 1 (of 3) heads, 0 (of 0) midpoints, 0 (of 2) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

GMPLS UNI Running Configuration

Head UNI-C for 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
          destination 100.20.20.20
          path-option 10 no-ero
        !
      !
    !
  !
  lmp
    gmpls optical-uni
      router-id 100.11.11.11
      neighbor nbr_A
        ipcc routed
        neighbor router-id ipv4 unicast 100.12.12.12
      !
      controller dwdm 0/1/0/1
        neighbor nbr_A
        link-id ipv4 unicast 192.168.100.1
```

```
        neighbor link-id ipv4 unicast 192.168.100.2
        neighbor interface-id unnumbered 13
      !
```

Tail UNI-C for a GMPLS Tunnel

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.

```
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
      !
    !
  !
  lmp
    gmpls optical-uni
      router-id 100.20.20.20
      neighbor nbr_B
        ipcc routed
        neighbor router-id ipv4 unicast 100.19.19.19
      !
      controller dwdm 0/1/0/1
        neighbor nbr_B
        link-id ipv4 unicast 192.168.103.2
        neighbor link-id ipv4 unicast 192.168.103.1
        neighbor interface-id unnumbered 22
      !
```

LSP Diversity

This example shows the configuration for two diverse LSPs:

```
mpls traffic-eng
   attribute-set xro exclude-tun1
    exclude best-effort lsp source 88.0.0.8 destination 10.0.0.2 tunnel-id 1
extended-tunnel-id 88.0.0.8
    !
   attribute-set xro exclude-tun2
    exclude strict lsp source 88.0.0.8 destination 10.0.1.2 tunnel-id 2 extended-tunnel-id
 88.0.0.8 lsp-id 2
    !
   gmpls optical-uni
    controller dwdm 0/1/0/0
     tunnel-properties
      logging events lsp-status state
      tunnel-id 1
      destination ipv4 unicast 10.0.0.2
      path-option 10 no-ero xro-attribute-set exclude-tun2
     !
    !
    controller dwdm 0/1/0/1
     tunnel-properties
      logging events lsp-status state
      tunnel-id 2
```

```
 destination ipv4 unicast 10.0.1.2
 path-option 10 no-ero xro-attribute-set exclude-tun1
!
!
```

Additional References

Standards

Standard	Title
OIF UNI 1.0	*User Network Interface (UNI) 1.0 Signaling Specification*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	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: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 3471	*Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description*
RFC 3473	*Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*
RFC 4208	*Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model*
RFC 4872	*RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery*

RFCs	Title
RFC 4874	*Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)*
RFC 6205	*Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers*

Technical Assistance

Description	Link
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.	http://www.cisco.com/techsupport